Mission-Integrated Path Planning for Planetary Rover Exploration

WU Peng
Beijing University of Technology, College of Electronic Information and Control Engineering, Beijing, China
Email: wupeng_1987@emails.bjut.edu.cn

JU Hehua
Beijing University of Technology, College of Electronic Information and Control Engineering, Beijing, China
Email: Juhehua@bjut.edu.cn

***Abstract*—This paper presents a mission planning method for planetary rover. As a decision support system, mission planning is important for rover navigation and control. Generating travel path and task sequence synchronously is the key point. In this paper, the environment model of planetary surface is summed up as two parts which are obstacle map and timeline. According to this, the mission planner utilizes state space search method for path planning on obstacle map. When expanding state, it executes task sequence planning and scheduling on timeline. Experimental results showed that this method could conduct the path planning and task sequence planning synchronously, which proved the viability of the method.**

***Index Terms*—Mission planning, Task scheduling, Timeline, Predicate calculus.**

I. INTRODUCTION

With the rapid development of the deep space exploration, planetary rover has become the most important mean to explore the planet’s surface. Previous successful experience demonstrates that it is necessary to develop decision-making system with mission planning functions [1–3].

The fundamental function of mission planning is to customize a set of task sequences for rover according to the planet's surface environment, ephemeris and the rover’s state. It is the basis of Guidance Navigation and Control (GNC). Unlike satellite and spacecraft, there is no specific orbit for planetary rover to trail, therefore it needs to plan path and task synchronously.

To see from the successful projects so far, there are two methods for rover mission planning:

One is based on the planning and scheduling theory, such as MER’s mission planning system MAPGEN [4], which is developed by NASA. This system relies on the operator to select the task sites and tasks manually. The progress between two sites is a task to be scheduled, and the cost of the path is estimated manually by the operator according to the terrain features [5].

Another method represented by TEMPEST system is based on path planning [6–8]. This system utilizes an Incremental Search Engine to search in the multidimensional state-space of the location, time, and energy. According to starting point and target point which are set manually, TEMPEST will plan path with task sequence on it.

Consider that the quick and safe progress is the basis of exploration, and tasks must be performed as progress, so the second method is better. Furthermore, in order to implement higher degree of automatic planning, science mission such as exploration and environment sensing need to be integrated into path, and it is necessary to take more environment and state constraints into consideration. So this paper presents a mission-integrated path planning method. It searches in a two-dimensional position state-space. When expanding state, it generates task sequence by predicate calculus and task scheduling, and adds the cost of the task sequence to the objective function, so as to achieve the purpose of planning path and task synchronously.

II. OUTLINE

According to rover’s current position in the elevation map of the planet's surface, the goal position, and the current time, Mission-Integrated Path Planner generates optimal traveling path, the mission sites as well as the sites’ task sequence, which satisfy the communication constraints, lighting constraints, temperature constraints, mobility constraints, resource constraints and kinematic constraints. Its outline is showed in Fig. 1. Mission planning relies on the pre-acquired planet's terrain information and ephemeris data, as well as the predefined rover parameters and tasks to be performed as progress.

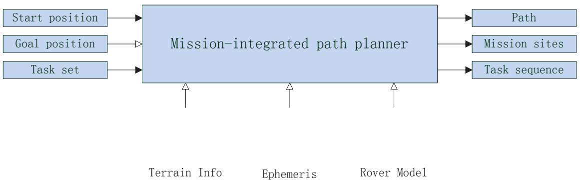


Figure 1. Outline of Mission-Integrated Path Planner.

Terrain information is usually described as digital elevation model (DEM). It records the height values for each grid point in the map. The raw data is a low-resolution model snapped by satellite or lander. As the rover progresses consecutively, DEM will be updated to a

doi:10.4304/jsw.8.10.2620-2627

high-resolution model via the vision camera and laser radar mounted on the rover.

Ephemeris data gives the spatial relationship at any time among the sun, the earth and extraterrestrial planet. The elevation angle and direction angle of the sun, the ground control station or the relay satellite relative to the rover are calculated according to ephemeris data. They will be used to judge the visibility of the sun and the ground station, and to compute the optimal working attitude.

Rover model includes performance parameters and resource parameters. The performance parameters describe the ability of rover to perform tasks, such as climbing ability, maximum charging power, structure geometry parameters, etc. Resource parameters record the status of the remaining continuous resources such as battery energy and free storage.

In this paper, a mission example with a set of seven tasks is taken into consideration. Table 1 summarizes the tasks and their state transition functions. The concept of state transition will be further defined in part IV.

TABLE I.
TASKS AND STATE TRANSITION FUNCTIONS

Task	State transition			
	$(\Delta X, \Delta Y)$ /grid	ΔT /hours	ΔE /Ah	ΔC /KB
Move	$(0/\pm1, 0/\pm1)$	$=f(\Delta X, \Delta Y)$	$=g(\Delta X, \Delta Y)$	0
Charge	(0,0)	$=f(\Delta E)$	$=E_{max}-E$	0
Sense	(0,0)	const.	const.	const.
Explore	(0,0)	const.	const.	const.
Transmit	(0,0)	const.	const.	const.
Dormancy	(0,0)	ΔT_d	0	0
Wait	(0,0)	ΔT_w	$=g(\Delta T_w)$	0

Among these tasks, only the ‘move’ task has effect on rover position. The ‘charge’ task supplements energy by adjusting the solar panel to track the sun. The ‘sense’, ‘explore’ and ‘transmit’ task are science related tasks. Their time expense, energy expense and storage capacity expense are defined by users according to mission requirements. The ‘dormancy’ task is mandatory when the rover lives long term without sunshine, and its time expense depends on the ephemeris data of specific planet.

III. ENVIRONMENT MODELING

A. Constraint Analysis and Modeling Method

Whether scientific tasks mentioned in Table 1 can be performed depends on many complex constraints. These constraints are listed in table 2.

Affected by the relative movement of celestial bodies and the unstructured terrain on planet surface, the environment conditions change over time and location. Here gives analysis of each constraint.

Slope: Whether terrain is traversable is affected by rover’s mobile performance and terrain slope. It is the chief constraint that all tasks should comply with. This constraint could be transformed into position constraint through setting slope threshold and computing traversable grids in DEM.

Direct sun: To ensure the energy supply, all tasks except ‘dormancy’ can only be performed when the sun is above the horizon and the rover is exposed to the sun.

Direct earth: The ground station on the earth should be in line of rover’s sight, when it needs data transmission and teleoperation.

Thermal control: In order to ensure the normal work of the scientific instruments and electrical equipment, the temperature should be maintained within safe scope. As the planet's surface temperature is under the influence of solar radiation, the rover must prevent direct solar radiation to its inside by body shading. Whether its inside is exposed depends on the sun elevation, the rover’s heading and the shading component’s attitude. In the mission level, the rover’s heading and the shading component’s attitude are specified, so this constraint is also time constraint.

Resource: There are two kinds of resources that affect the tasks performance, which are battery energy and free storage capacity.

Time bound: In order to meet scientists’ requirements, it should be allowed for tasks to set the permitted time bounds and prohibited time bounds, or to specify a corresponding moment of specific environment state, such as the sun elevation at max value.

TABLE II.
CONSTRAINTS LIMITED TASK PERFORMANCE

Constraint	Affected task
Slope	Move, Charge, Sense, Explore, Transmit, Dormancy, Wait
Direct sun	Move, Charge, Sense, Explore, Transmit, Wait
Direct earth	Move, Charge, Sense, Explore, Transmit
Thermal control	Move, Charge, Sense, Explore, Transmit, Wait
Resource	Move, Sense, Explore, Transmit, Wait
Time bound	Charge, Sense, Explore, Transmit, Dormancy, Wait

Both the ‘direct sun’ and ‘direct earth’ constraint can be classified as ‘target visibility’ constraint. It is mainly affected by two factors. Firstly, the elevation angle of the target relative to the rover should be in a certain range. As celestial bodies move, the elevation angle varies over time. So this constraint can be transformed into time constraint by ephemeris calculation. Secondly, when the elevation angle is valid, the line of sight must not be blocked by terrain. By computing with DEM and ephemeris data, there are a lighting shadow area and a communication shadow area. So target visibility constraint is also position constraint.

According to analysis above, some constraints can be simplified through environment processing. The procedure is depicted in Fig. 2.

Firstly, terrain and ephemeris data can be used to compute lighting shadow area, communication shadow area, and obstacle area. So the target visibility constraint and the slope constraint will be transformed into position constraint in obstacle map.

Secondly, ephemeris calculation generates the lighting blind interval, the communication blind interval, the thermal control interval, as well as the interval of time-limited task. Then the target visibility constraint, temperature constraint, and time bounds can be transform into time constraint.

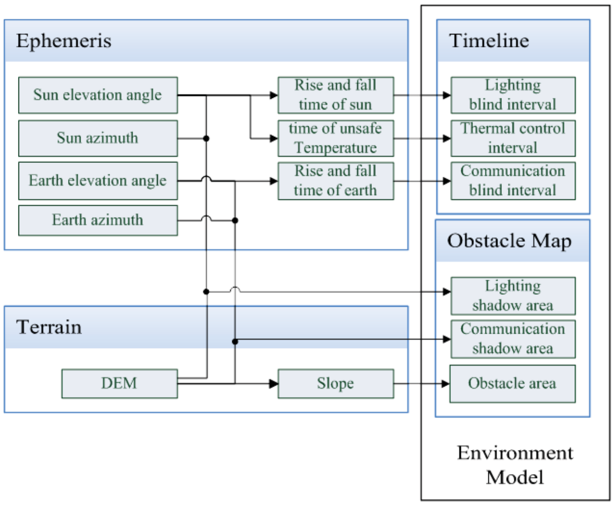


Figure 2. The procedure of environment processing.

B. Obstacle Map Extraction

Terrain obstacle extraction may employ the slope calculation method. That is, for every grid in DEM, compute the slope and set the grid of which the slope is beyond the threshold value. The result is just as Fig. 3 shows.

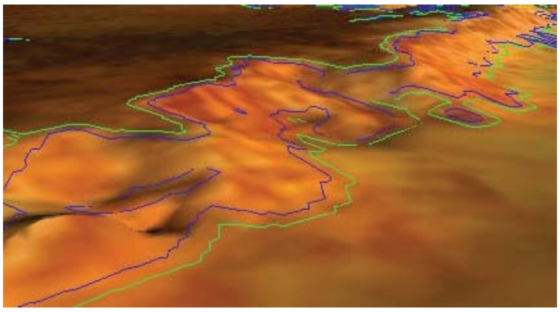


Figure 3. Terrain obstacle extraction result.

For the computation of lighting shadow area and communication shadow area, this paper employs the interpolation test method. (x_0, y_0) represents the horizontal coordinates of a location point. θ and σ represent the elevation angle and azimuth angle of the target relative to this location point. They are showed in Fig. 4 and Fig.5.

Then the direction vector of the target is:

$$p = (\cos\theta\sin\sigma, \cos\theta\cos\sigma, \sin\theta) \tag{1}$$

The target sight equation is:

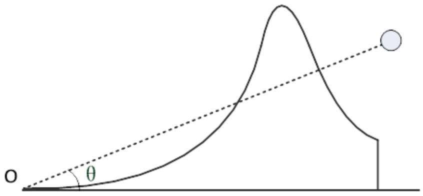


Figure 4. Lateral view of line-of-sight.

$$\begin{cases} x = x_0 + t\cdot\cos\theta\sin\sigma \\ y = y_0 + t\cdot\cos\theta\cos\sigma \\ z = z_0 + t\cdot\sin\theta \end{cases} \tag{2}$$

Perform interpolation test along the line-of-sight With a resolution of ε. The n-th point's coordinate is:

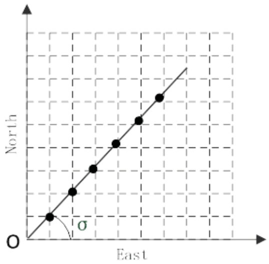


Figure 5. The horizontal projection of the line-of-sight.

$$(x_n, y_n) = (x_0 + n\cdot\varepsilon\cdot\cos\theta\sin\sigma, y_0 + n\cdot\varepsilon\cdot\cos\theta\cos\sigma) \tag{3}$$

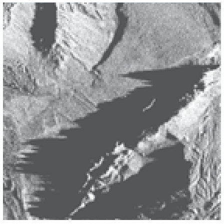
Its height value in the line-of-sight is:

$$z_n = z_0 + n\cdot\varepsilon\cdot\sin\theta \tag{4}$$

Let h_n represents the geographical elevation of this point, $h_{\max}$ represents the maximal elevation in DEM. Make n=0, 1, 2… and test, and end the test when meets the condition $z_n > h_{\max}$. If there appears:

$$\Delta = z_n - h_n < 0 \tag{5}$$

Then mark the point as a shadow point. Fig. 6 is the effect figure of the lighting shadow.



Solar elevation angle: 5.0503°
Solar azimuth angle: 93.1995°

Figure 6. The effect figure of lighting shadow.

C. Timeline Computation

As the surface temperature influences task performance, when the rover takes specific heading angle and right wing angle, whether a task is performable depends on the solar elevation. It is showed with (6). As depicted in Fig.7, α,β,θ and represent the heading angle, right wing angle, solar elevation respectively.

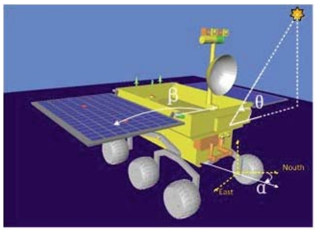
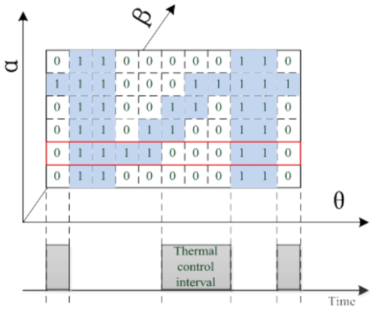


Figure 7. Rover's specific angles.

$$T(\alpha,\beta,\theta) = \begin{cases} 0 & \text{task is unperformable} \\ 1 & \text{task is performable} \end{cases} \tag{6}$$

The value of $T(\alpha,\beta,\theta)$ needs to be computed under thermal control demand, and it is not the main job of this paper. According to (6), Fig. 8 depicts the performability in a three-dimensional space. As the solar elevation $\theta=\theta(t)$ is time-varying, it will be mapped to thermal control interval I_T .



0	1	1	0	0	0	0	0	1	1	0
1	1	1	0	0	0	1	1	1	1	1
0	1	1	0	0	1	1	0	1	1	0
0	1	1	0	1	1	0	0	1	1	0
0	1	1	1	1	0	0	0	1	1	0
0	1	1	0	0	0	0	0	1	1	0

Figure 8. Temperature effect maps to thermal control interval

I_T is defined by (7) and (8), which is composed with several subintervals.

$$I_{Ti}=[t_i^-,t_i^+],(\forall t\in I_{Ti},T(\alpha,\beta,\theta(t))=0) \qquad (7)$$

$$I_T=\bigcup_{i=1} I_{Ti},(\forall i\neq j,I_{Ti}\cap I_{Tj}=\varnothing) \qquad (8)$$

Similarly, lighting blind interval I_{LB} is defined by (9) and (10). $\theta_{threshold}$ is the threshold value of solar elevation angle, under which the rover is not allowed to work and 'Dormancy' task should be performed.

$$I_{LBi}=[t_{LBi}^-,t_{LBi}^+],(\forall t\in I_{LBi},\theta(t)<\theta_{threshold}) \qquad (9)$$

$$I_{LB}=\bigcup_{i=1} I_{LBi},(\forall i\neq j,I_{LBi}\cap I_{LBj}=\varnothing) \qquad (10)$$

Communication blind interval I_{CB} is defined by (11) and (12). $\varphi(t)$ is the elevation angle of ground station relative to the rover. $\varphi_{threshold}$ is the threshold angle, which should not be surpassed if the rover need to communicate with ground control station.

$$I_{CBi}=[t_{CBi}^-,t_{CBi}^+],(\forall t\in I_{CBi},\varphi(t)<\varphi_{threshold}) \qquad (11)$$

$$I_{CB}=\bigcup_{i=1} I_{CBi},(\forall i\neq j,I_{CBi}\cap I_{CBj}=\varnothing) \qquad (12)$$

Fig. 9 shows the solar elevation angle and azimuth angle varies with time, and how it maps to lighting blind interval.

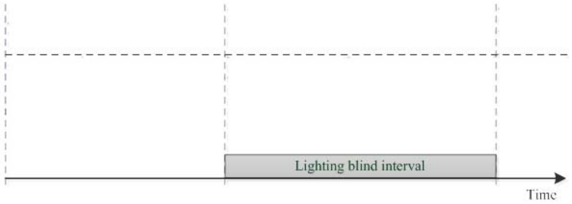


Figure 9. Solar elevation maps to lighting blind interval.

Considering the intervals defined above, as well as the manually set time bounds, there should be a set of limited intervals which showed by (13). I_{LT} is the interval of time-limited task. It emerges with the manually set time bounds.

$$I=I_T\cup I_{LB}\cup I_{CB}\cup I_{LT} \qquad (13)$$

A characteristic timeline integrated with these intervals is also generated. It can be used for task sequence planning.

IV. MISSION-INTEGRATED PATH PLANNING

For the environmental model established above, this paper presents a planning engine based on hierarchical structure, which is showed in Fig. 10. There are path planner and task sequence planner from top to down. Path planner searches in position state space to seek for optimal path. When expand a node, task sequence planner generates task sequence according to rover's status, environmental constraints and scientific tasks demand. The transfer cost to the next node is also calculated by task sequence planner.

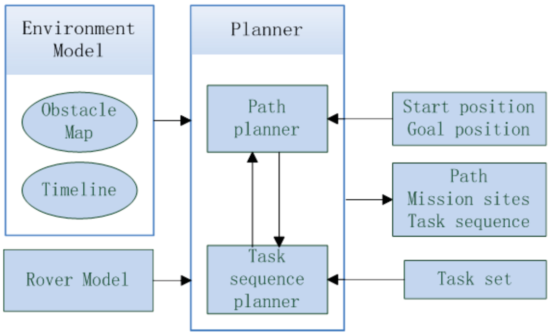


Figure 10. Planner's structure.

A. Path Planning

Traditional path planning algorithm such as A* is based on the state-space search [9]. The core of A* algorithm is to sort and expand the nodes in the state space by the objective function as follows:

$$f(x)=g(x)+h(x) \qquad (14)$$

$h(x)$ is the heuristic function used to reduce the search space. For all nodes x and its successor node x' , there is:

$$g(x')=g(x)+C(x,x') \qquad (15)$$

In mission-integrated path planning, $C(x,x')$ is computed as (16).

$$C(x,x')=\Phi(trav(x,x'))+\sum_{i=1}^{k}\Psi(a_i(x,x')) \qquad (16)$$

$\Phi(trav(x,x'))$ is the function to compute travelling cost. $\Psi(a_i(x,x'))$ is the cost due to perform task a_i .

In order to ensure the admissibility of the algorithm, the heuristic must meet consistency. That is:

$$\begin{cases} h(x_{goal},x_{goal})=0 \\ h(x,x_{goal})\le C(x,x')+h(x',x_{goal}) \end{cases} \qquad (17)$$

So it should be satisfied that $C(x,x')\ge 0$. In this paper, In this paper, $\Phi(trav(x,x'))$ and $\Psi(a_i(x,x'))$ are computed as time cost of task sequence. That means:

$$\Phi(trav(x,x'))=t_{trav(x,x')} \qquad (18)$$

$$\Psi(a_i(x,x'))=t_{a_i(x,x')} \qquad (19)$$

B. Task sequence planning

Task planning and scheduling has been warmly talked for a long time in the field of artificial intelligence. The STRIPS[10] planner is famous for planning task sequence. However, it is unable to handle temporal constraints between tasks. The HSTS[11] planner and the subsequent EUROPA[12] planner enable the temporal constraint reasoning function using the famous theory 'Temporal Constraint Networks'[13][14]. Under this framework, temporal planning is treated as a Constraint Satisfaction Problems (CSP). According to Dechter's[15] opinion, the basis constraint processing algorithms are inference and search. Among inference algorithms, the most common ones are local consistency algorithms, such as arc-consistency, path-consistency, i-consistency etc. This theory is widely used, but it is not much useful for our mission. Because most tasks to be performed have an effect on resource states such as ΔE and ΔC listed in table 1, and every variable assignment in CSP may cause the domain variation of other variables. However, inspired by this theory, we mentioned a planning and scheduling algorithm for the task sequence planning.

In order to explain the planning algorithm, we make some definition.

Definition 1: State transition system is a triple $\Sigma = (\mathrm{S}, \mathrm{A}, \gamma)$.

- $\mathrm{S} = \{\mathrm{s}_1, \mathrm{s}_2 \cdots\}$ is a limited set of states
- $\mathrm{A} = \{\mathrm{a}_1, \mathrm{a}_2, \cdots\}$ is a limited set of actions
- $\gamma : \mathrm{S} \times \mathrm{A} \rightarrow 2^{\mathrm{S}}$ is a state transition function

For task sequence planning, *S* contains the rover's resource status and other state variables. *A* is task set. Task performance is described as a state transition, that is:

$$S \xrightarrow{a} S' \qquad (S' = \gamma(S, a), a \in A)$$

For example:

$$(x, y, t, e) \xrightarrow{Move} (x + \Delta x, y + \Delta y, t + \Delta t, e + \Delta e)$$

$$(e, c, t) \xrightarrow{Explore} (e + \Delta e, c + \Delta c, t + \Delta t)$$

$$(\mathrm{e}, \mathrm{t}) \xrightarrow{\mathrm{Charge}} (e + \Delta e, t + \Delta t)$$

Definition 2: A task is a quintuple

$$\mathrm{m} = (\mathrm{name}(\mathrm{m}), \mathrm{trigger}(\mathrm{m}), \mathrm{precond}(\mathrm{m}), \mathrm{effect}(\mathrm{m}), \mathrm{interval}(\mathrm{m}))$$

For a task 'm', its five elements' meanings are:

- name(m): The name of the task 'm'.
- trigger(m): The trigger condition of the task 'm'. It is a conjunction of atomic predicates on the state variables. If it is true, this task must be performed.
- precond(m): The precondition of the task 'm'. It is a set of atomic predicates on the state variables. When 'm' must be performed, this condition should be check if it meets current state S.
- effect(m): The effect of the task 'm'. It is also a set of atomic predicates on the state variables. When 'm' has been performed, it will update the current state S. New state $S' = \gamma(S, m)$.
- interval(m): The period of time occupied by the task 'm'. It is expressed as $[\mathrm{I}^-, \mathrm{I}^+)$.

Analytically, the task can be divided into the following categories:

- State Constraint Mission (SCM): The task must be performed when the external state reached its trigger condition.
- Resource Constraint Mission (RCM): The task must be performed when the internal resource is lack.
- Time Constraint Mission (TCM): Task that must be performed within a certain time interval.

Based on the Predicate calculus and the temporal collision detection, the planning procedure is showed as Fig. 11.

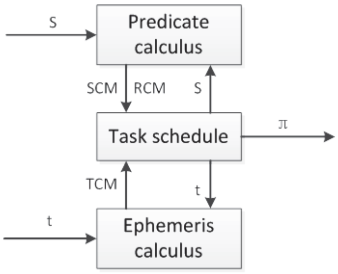


Figure 11. The procedure of task sequence planning.

For given state S, predicate calculus module generates the SCM or RCM to be executed. Ephemeris calculus provides the next time blind interval (TBI) which is adjacent to the current time 't'. As a result, it also generates the corresponding TCM. The core task scheduling module set the start time point and the end time point for SCM or RCM, and then generates the task plan 'π' finally.

The pseudo-code of task scheduling algorithm is showed as Fig. 12.

Procedure Schedule(π)

```
SCM ← PredicateCal(s,trigger(scm));
if(SCM=Null)    return(π);
else    L.Push(SCM);
TBI ← EphemerisCal(t);
while(L ≠ ∅)
    m ← L.Top();
    while(Consistent(m,TBI)=false)
        $\pi \leftarrow \pi \cup \{Idle, TCM \mid I_{Idle}=[t, I_{TBI}^-], I_{TCM}=[I_{TBI}^-, I_{TBI}^+]\}$;
        $t \leftarrow I_{TBI}^+$;
        TBI ← EphemerisCal(t);
    s ← UpdateState(t);
    RCM ← PredicateCal(res,precond(m));
    if(RCM=Null)
        m ← L.Pop();
        $\pi \leftarrow \pi \cup \{m \mid Im=[t, t+\Delta t]\}$;;
        $t \leftarrow I_m^+$;
        s ← UpdateState(t);
    else
        L.Push(RCM);
Schedule(π);
```

Figure 12. The task scheduling algorithm

In the pseudo-code, 't' is a time variable represents current time. The variable 'res' represents resource status. It is a part of rover state 'S'. According to the current state 'S', the predicate calculus function '*PredicateCal(S,atom)*' scans each SCM's toggle condition if (atom=trigger(scm)), and generates a SCM.

Or it will scans SCM's precondition if (atom=precond(m)), and generates corresponding RCM to supply resources. According to the current time 't', the ephemeris calculus function '*EphemerisCal(t)*' generates next TBI, and '*UpdateState(t)*' update 'S'. '*Consistent(m,TBI)*' use (20) to check if the task 'm' is consistent with TBI.

$$Consistent(\mathrm{m},\mathrm{TBI})=\begin{cases} true & (interval(\mathrm{m})\cap\mathrm{TBI}=\varnothing) \\ false & (interval(\mathrm{m})\cap\mathrm{TBI}\neq\varnothing) \end{cases} \tag{20}$$

Here we employ an example of the scheduling for a single task 'explore' to illustrate the task schedule algorithm. Its procedure is depicted in Fig. 13.

This algorithm maintains a queue of tasks to be scheduled. It is a stack and named '*L*' in the pseudo-code. At the beginning, $L=\varnothing$ and mission plan $\pi=\varnothing$. The predicate calculus module generates a SCM{01}, and pushs it into stack *L*{03}. Ephemeris calculus module generates next TBI according to current time variable 't'{04}. Then, the algorithm performs time constraint handling{07-10} and resource constraint handling{11-19} for every task at the top of *L*{06}. In the time constraint handling part, if interval(m) is not consistent with given TBI, an 'Idle' task and a corresponding TCM will be added to mission plan at current time 't'{08}. This step is showed as Fig. 13(a) and (b). After 't' is updated, ephemeris calculus module will be called again to generate next TBI{10}, and the algorithm will check consistency circularly until all intervals are consistent{07}. In the resource constraint handling part, predicate calculus module is called to check if precond(m) is satisfied{12}. If it is satisfied, *PredicateCal(res,precond(m))* returns Null{13}. The task 'm' will be added to mission plan at current time 't', and then resource states will be updated{14-17}. If it is not satisfied, *PredicateCal(res,precond(m))* returns corresponding RCM to supplement resource{19}. In the example, the RCM is the 'Charge' task as showed in Fig. 13(c).

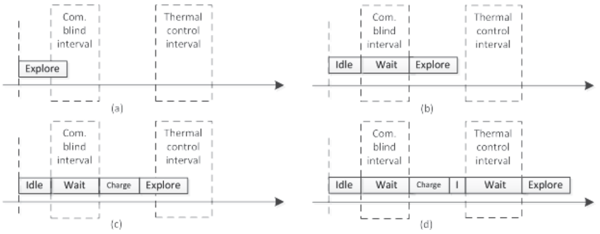


Figure 13. An example of scheduling procedure for 'explore' task.

So as result, the planner generates segmental plan after scheduling for the task 'explore', which is showed in Fig. 13 (d).The result plan is π={Idle, Wait, Charge, Idle, Wait, Explore}. Here, the 'Idle' task is generated by the planner automatically to fill the idle period on the timeline. Its meaning should be predefined by user.

V. EXPERIMENT AND ANALYSIS

According to the environment modeling method and mission-integrated path planning method described above, we developed a principle verification system.

Although several mission planning methods were mentioned before, there is still no comparability among them, because the functions achieved by different methods are not quite similar. So here we only verify mission-integrated path planning method mentioned above, without comparing with others.

In the experiment, task set includes seven tasks which are 'move', 'explore', 'sense', 'dormancy', 'transmit', 'wait' and 'charge'. Path planning used the time cost to calculate the target function. We conducted two experiments, in which the start position and goal position are same but the start time differs. In experiment 1, we set the starting time at 17:00 on Feb. 21, 2013, and the result plan showed the end time is at 00:22 on 2013 Feb. 23. In experiment 2, we set it at 22:00 on Feb. 28, 2013, and the result plan showed the end time is at 07:47 on 2013 Mar. 03. Fig. 14 showed the paths and the mission sites in both experiment.

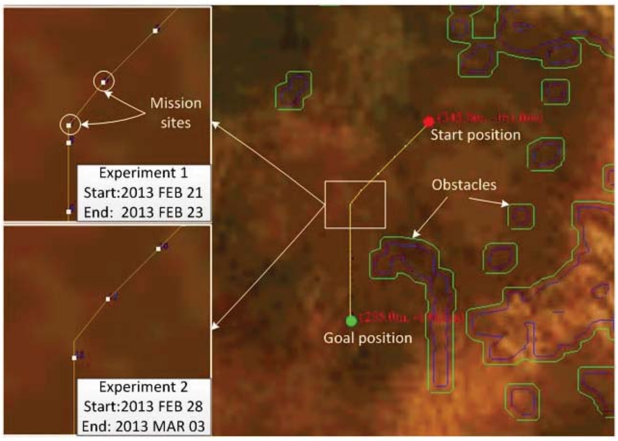


Figure 14. Path and mission sites in two experiments.

The timeline of each task in experiment 1 is showed in Fig. 15. Progress distance and battery energy vary as in Fig. 16.

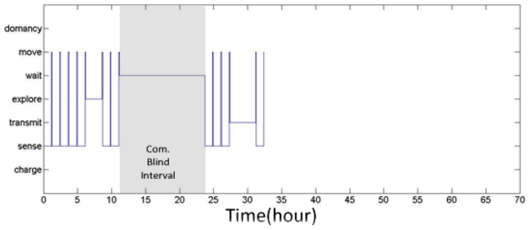


Figure 15. Timeline of each task in experiment 1.

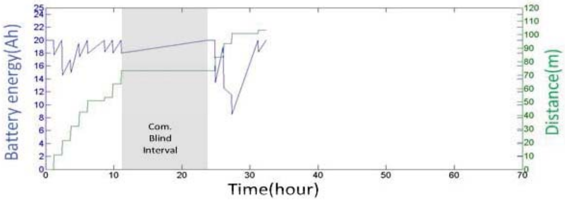


Figure 16. Progress distance and battery energy vary with time in experiment 1.

Affected by the starting time, the result plan contains only one communication-blind-interval. From the two figures we could see that during the communication-blind-interval, the rover must perform 'wait' task, and the progress distance does not increase. For the planner allows charging when performing stationary task in the sun, battery energy only decreases when moving, and increases in other time.

The statistics of task in experiment 1 is listed in table 3. As the battery energy is always maintained in the permitted range, there's no 'charge' task in this plan.

TABLE III.
STATISTICS OF ALL TASKS IN EXPERIMENT I

Task	number	time cost (min)	percentage
move	23	62.8	3.20%
charge	0	0	0%
sense	11	748	38.50%
explore	1	148	7.60%
transmit	1	230	11.80%
wait	1	753.8	38.80%
dormancy	0	0	0%

As a comparison, the timeline of each task in experiment 2 is showed in Fig. 17. Progress distance and battery energy vary as in Fig. 18. Also under the influence of the starting time which is different from the one in experiment 1, there are four lighting-blind-intervals and three communication-blind-intervals in this result plan. As a consequence, more 'wait' and 'charge' tasks are added, and the span of this plan is extended to about 64 hours.

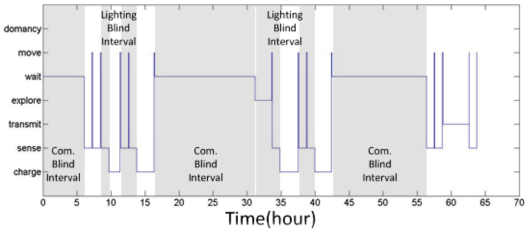


Figure 17. Timeline of each task in experiment 2.

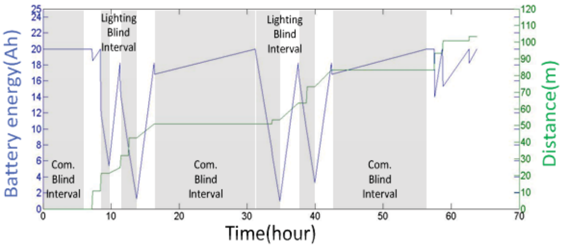


Figure 18. Progress distance and battery energy vary with time in experiment 2.

The statistics of task in experiment 2 is listed in table 4. From table 3 and table 4 we see that the 'wait' and 'sense' task are the most time-consuming tasks, and the 'move' task is the least time-consuming task.

TABLE IV.
STATISTICS OF ALL TASKS IN EXPERIMENT II

Task	number	time cost (min)	percentage
move	25	62.8	1.6%
charge	4	558.2	14.6%
sense	11	748	19.5%
explore	1	148	3.9%
transmit	1	230	6.0%
wait	3	2080.5	54.4%
dormancy	0	0	0.0%

After comparison, the statistics of two result plans are listed in table 5.

TABLE V.
RESULT COMPARISON BETWEEN TWO EXPERIMENTS

	Experiment 1	Experiment 2
travelled distance(m)	104.65	104.65
number mission sites	23	25
total number of tasks	37	45
total time cost(min)	1942.6	3827.5
max depth of discharge	57.40%	95.15%

VI. CONCLUSION

Considering the requirements of mission planning system for planetary rover, and multiple factors that affect the task performance, this paper presents a mission-integrated path planning method. This method simplifies complex constraints through environment processing, and deal with them hierarchically in planning engine. The planner can generate the optimal patrol path, the mission sites and the task sequence synchronously.

Current planning algorithm is only for transient environment model. Future research will focus on improving the re-planning performance, so as to satisfy the needs of mission planning in time-varying environment.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation of China under grants No. 60975065 and No.61004061.

REFERENCES

[1] N. Muscettola, P. Nayak, B. Pell, and B. Williams, "Remote agent: To boldly go where no AI system has gone before," Artificial Intelligence, vol. 103, no. 1, pp. 5-47, 1998.
[2] S. Chien, G. Rabideau, and R. Knight, "Aspen–automated planning and scheduling for space mission operations" Space Ops, pp. 1-10, 2000.
[3] J. L. Bresina, A. K. Jónsson, P. H. Morris, and K. Rajan, "Activity Planning for the Mars Exploration Rovers," International Conference on Automated Planning and Scheduling. 2005.
[4] M. Ai-Chang and J. Bresina, "Mapgen: mixed-initiative planning and scheduling for the mars exploration rover mission," Intelligent Systems, vol. 19, no. 1, pp. 8-12, 2004.
[5] J. Bresina, A. Jónsson, P. Morris, and K. Rajan, "Mixed-initiative planning in MAPGEN: Capabilities and shortcomings," Proceedings of the ICAPS-05 Workshop on Mixed-initiative Planning and Scheduling, Monterey, CA. 2005.
[6] P. Tompkins, "Mission-directed path planning for planetary rover exploration," Diss. Jet Propulsion Laboratory, 2005.
[7] P. Tompkins, A. Stentz, and D. Wettergreen, "Mission-level path planning and re-planning for rover exploration," Robotics and Autonomous Systems, vol. 54, no. 2, pp. 174-183, Feb. 2006.

[8] P. Tompkins and A. Stentz, “Mission-level path planning for rover exploration,” Proceedings of the 8th Conference on Intelligent Autonomous Systems (IAS-8). 2004.

[9] N. J. Nilsson, Principles of artificial intelligence. Symbolic Computation, Berlin: Springer, 1982.

[10] N. J. Fikes, Richard E and Nilsson, “STRIPS: A new approach to the application of theorem proving to problem solving,” Artificial intelligence, vol. 2, no. 3, pp. 189--208, 1972.

[11] a. Garrido and F. Barber, “Integrating planning and scheduling,” Applied Artificial Intelligence, vol. 15, no. 5, pp. 471-491, May 2001.

[12] Jonsson , Ari, et al. "Planning in interplanetary space: Theory and practice." Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling. 2000.

[13] R. Dechter, “Temporal constraint networks,” Artificial Intelligence, vol. 49, no. 1, pp. 61-95, May 1991.

[14] M. Ghallab, D. Nau, and P. Traverso, Automated planning: theory & practice. Morgan Kaufmann, 2004, pp. chapter 13.

[15] R. Dechter, Constraint processing. Morgan Kaufmann, 2003.

Peng Wu was born in 1987. He received his B.S in Measuring and Control Technology and Instrumentations from Harbin Institute of Technology, China, in 2010, and received the M.S. degree in Control Science and Engineering from Beijing University of Technology, China, in 2013. His research focuses on planetary rover mission planning.

Hehua Ju was born in1969. He received his Ph.D degree in Flight Vehicle Design from Harbin Institute of Technology, China in 2003.

He is currently a professor in College of Electronic Information and Control Engineering, and the director of Deep Space Exploration Robotics Research Center, Beijing University of Technology. His interests include robot motion planning and control, deep space probes navigation and control, and behavior control theory.