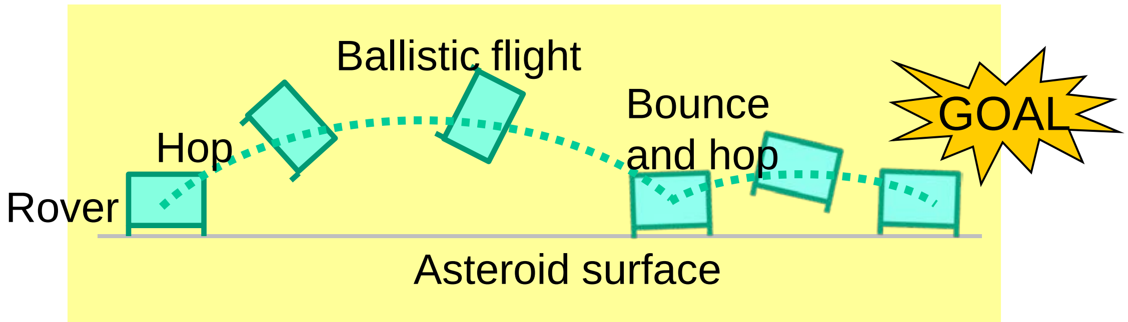
Welcome to
8th Spacecraft Control System Design Contest

July 10, Friday
14:00-16:00
Room 501
Kobe International Conference Center 5F



The Theme of the 8th Spacecraft Control System Design Contest is
"**Hopping Rover**"

Outline of the problem

On the asteroid surface, the gravity is very week. Therefore, the rover easily hops while it moves. Here we consider the hopping mechanism instead of the traditional wheeled mechanisms for the rover to explore the asteroid surface.

In the beginning, the rover is stationary on the asteroid surface. The rover includes a torquer inside. By rotating the torquer, the contact point of the bottom face of the rover is pushed against the surface until the rover hops. Due to the friction force, the rover gets the horizontal velocity.

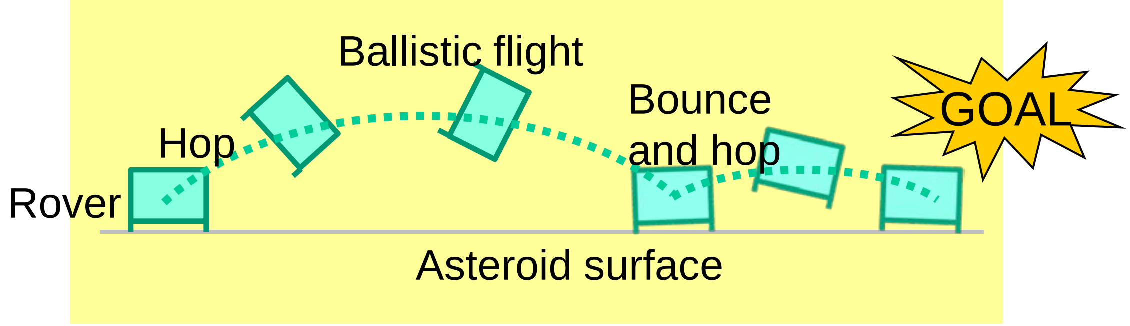
After the ballistic flight under the micro-gravity of the asteroid, the rover re-lands on the surface. It will have several bounces, get contact with the surface and then hop again. By repeating hops and ballistic flights, the rover will reach the target point.

The time to reach the target point is defined as the minimum simulation time when the rover keeps its contact point within +/- 0.05 meters from the target point for longer than or equal to 5 seconds.

Evaluation is made how quickly the rover reaches the target point with the limited mobility using hopping.

The difficulties lie in the indirect control of the position through the contact force and high nonlinearity in the slide and stop dynamics.

NOTE: Values may be changed in order to make the contest more exciting.



Ref. T. Yoshimitsu, et al: Analysis on Hopping Mechanism of Planetary Robotic Vehicle by Microgravity Experiments, pp.3-13, Microgravity sci. technol. XIII/4 (2002)

- Coordinates (Fig.1)

The asteroid surface frame is defined as follows .
The surface is assumed to be flat and horizontal.

Origin (O)	: a fixed point on the asteroid surface
x-axis (X)	: horizontal direction
y-axis (Y)	: vertical direction (opposite to the gravity)

The rover position and attitude is defined as follows.

(x, y)	: position of center of mass (c.m.)
θ	: attitude

The forces and torques acting on the rover are defined as follows. Figure 1 shows the case when the rover contacts with the asteroid surface at contact point A.

f_x^A	: friction force at A
f_y^A	: contact force at A
mg	: gravity(g=9.8e-5 m/s^2)
T	: control torque

The rover may contact with the asteroid surface at contact point B. For such case, friction and contact forces at B are defined as follows.

f_x^B	: friction force at B
f_y^B	: contact force at B

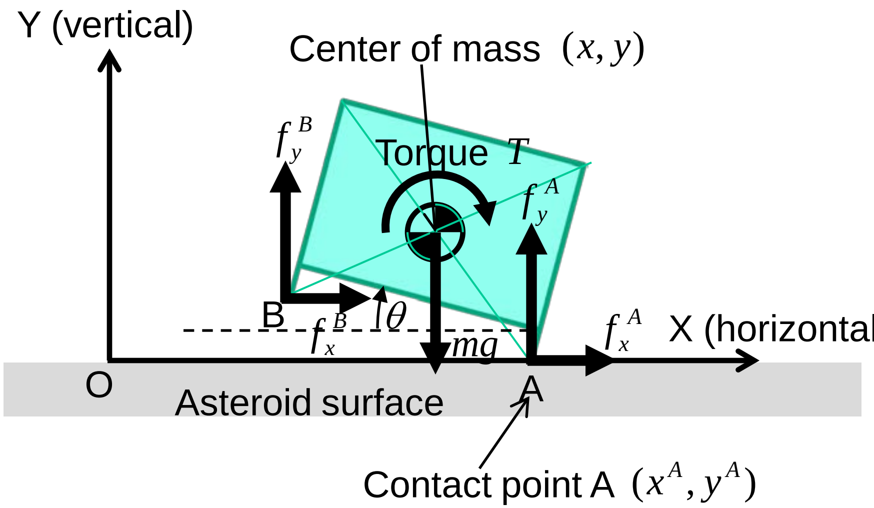


Fig 1. Definition of the asteroid surface frame

- Rover configuration (Fig.2)

The rover configuration is as follows.

m	: mass of the rover(1.0kg)
I	: rover inertia around c.m.(0.0017kgm^2)
$2r, 2h$	: width and height (0.1m, 0.1m)
$2l$	: diagonal length(0.14m)
α	: $\tan^{-1}(h/r)$=45deg

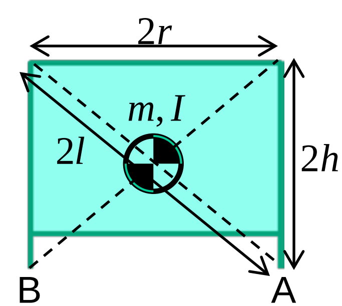


Fig 2. Rover configuration

- Rover state

Table 1 summarizes the rover states.

Table 1 Summary of rover states

State	Number of contact points	Motion
A	0	Hops
B	1 (A)	Pivots
C	1 (A)	Slides
D	1 (B)	Pivots
E	1 (B)	Slides
F	2 (A,B)	Stops
G	2 (A,B)	Slides

Table 2 gives kinematical expressions for Table 1.

Table 2 Kinematical expressions for rover states

State	Position	Velocity
A	$y_A \geq 0,\ y_B \geq 0$	-
B	$y_A = 0,\ y_B > 0$	$\dot{x}_A = 0,\ \dot{y}_A = 0$
C	$y_A = 0,\ y_B > 0$	$\dot{x}_A \neq 0,\ \dot{y}_A = 0$
D	$y_A > 0,\ y_B = 0$	$\dot{x}_B = 0,\ \dot{y}_B = 0$
E	$y_A > 0,\ y_B = 0$	$\dot{x}_B \neq 0,\ \dot{y}_B = 0$
F	$y_A = 0,\ y_B = 0$	$\dot{x}_A = \dot{x}_B = 0,\ \dot{y}_A = \dot{y}_B = 0$
G	$y_A = 0,\ y_B = 0$	$\dot{x}_A \neq 0, \dot{x}_B \neq 0, \dot{y}_A = \dot{y}_B = 0$

- Equations of motion

The translational and rotational motion equations are given by

$$
\begin{aligned}
m\ddot{x} &= f_x^A + f_x^B \\
m\ddot{y} &= f_y^A + f_y^B - mg \\
I\ddot{\theta} &= T - lf_x^A \sin(\theta+\alpha) - lf_y^A \cos(\theta+\alpha) \\
&\quad + lf_x^B \sin(\theta-\alpha) + lf_y^B \cos(\theta-\alpha)
\end{aligned}
$$

where $\ddot{x}$ denotes d^2x/dt^2 .

- Friction model

When the rover stops on the surface, the static friction is applied. For example, in the state B, it holds

$$\left|f_x^A\right| \leq \mu_s f_y^A,\ f_y^A \geq 0,\ \dot{x}_A = 0$$

where μ_s (=0.6) is coefficient of static friction. When the rover slides on the surface, the dynamic friction is applied. For example, in the state C, it holds

$$f_x^A = -\mu_d f_y^A,\ f_y^A \geq 0,\ \dot{x}_A > 0$$

where μ_d (=0.5) is coefficient of dynamic friction.

- Actuator model

The torquer can produce a control torque between $T_{\min}$ (=-5e-4 Nm) to $T_{\max}$(=+5e-4 Nm) continuously.

- Contact model

When the rover falls on to the surface or tumbles down on to the surface, an impulsive force will be applied to the rover at the contact point.
Figure 3 describes the contact model when the rover contacts with the surface at point A, where $e_x(=0.8), e_y(=0.2)$ denotes coefficients of restitution for horizontal and vertical direction and upper case symbol '-' and '+' denotes 'before' and 'after' the contact.

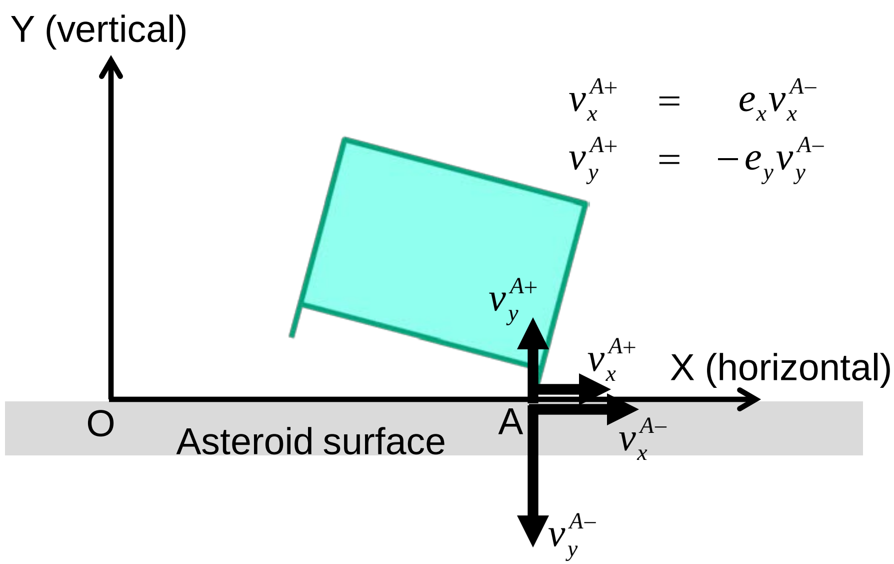


Fig 3. Contact model

In the simulation, the velocity change due to the contact is done instantaneously. If the size of the resultant velocity is below the threshold value $v_{x\min}(=1e-6\mathrm{m/s}), v_{y\min}(=1e-6\mathrm{m/s})$, then the resultant velocity is regarded as zero.

NOTE: The contact points are only on the bottom face of the rover. If the top face contacts with the asteroid surface, the rover will be broken (the simulation is terminated.)

- Initial condition

In the beginning, the rover stands upright on the asteroid surface.

$$x=0, y=h, \theta=0, \dot{x}=0, \dot{y}=0, \dot{\theta}=0 \quad (\text{at } t=0)$$

- Goal of the control

The rover needs to reach the target point.
The time to reach the target point is defined as the minimum simulation time when the rover keeps its contact point (either A or B) within +/- 0.05 meters from the target point for longer than or equal to 5 seconds.

- State transition

The rover state changes with time depending on the applied torque and contact dynamics. Figure 4 shows the state transition diagram.

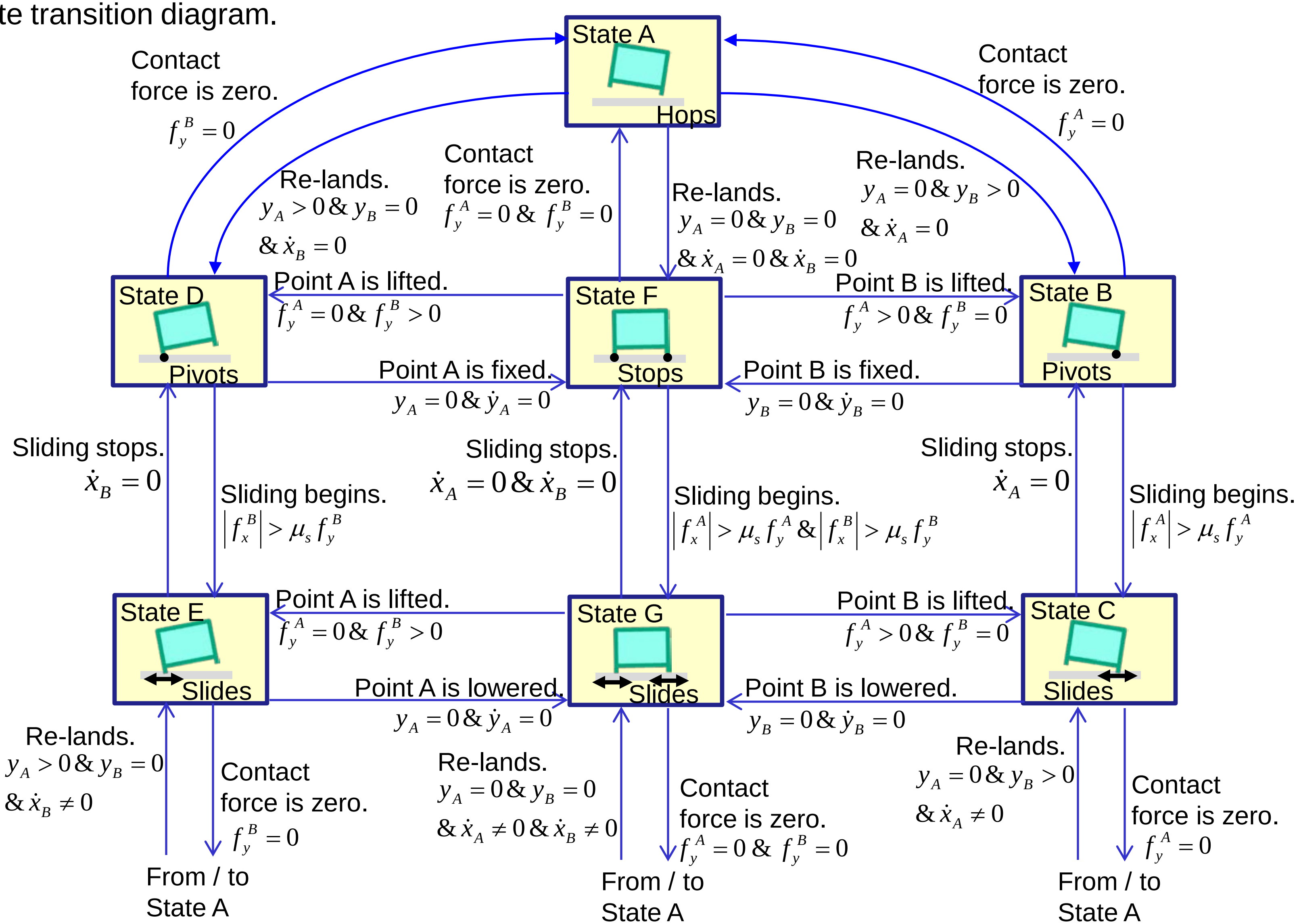


Fig 4. State transition diagram

- Detailed specifications of software to be prepared by participants

Matlab® program (“m file”) should be prepared in the form of the following function. Please refer to the sample program to be distributed for the detailed information such as the size of the vectors.

function t_cmd = Control(pos,vel,the,omg,posT,t,smt,state)

INPUTS

“pos”: the position of the c.m. of the rover expressed in the asteroid surface frame [m]
“vel”: the velocity of the c.m. of the rover expressed in the asteroid surface frame [m/s]
“the”: the attitude of the rover [rad]
“omg”: the attitude rate of the rover [rad/s]
“posT”: the position of the target point in the asteroid surface frame [m]
“t”: current simulation time [s]
“smt”: total simulation time [s]
“state”: the rover state (A,B,…, or G) []

OUTPUTS

“t_cmd“: torquer input command [Nm]

Regarding calling/calculation sequence, please see the timing chart below (Fig. 5). There is no time delay in measurement or actuation.

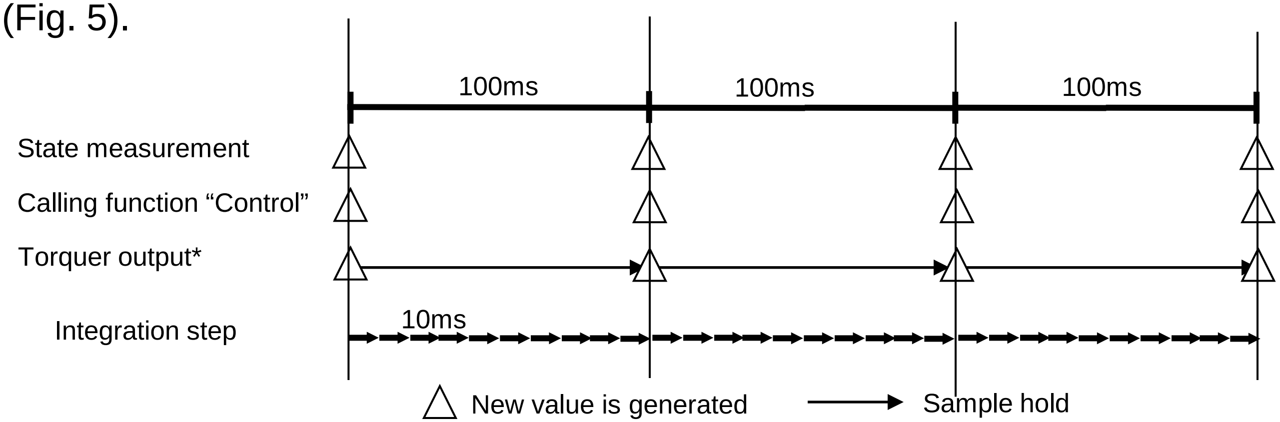


Fig. 5 Calling/calculation sequence

Matlab is a trademark of MathWorks Inc. (http://www.mathworks.com/index.html)

- Q&As

(Q1) When I ran the sample program (Rev. A) with Matlab R2015a, it stopped with the error message about the matlab.ui.Figure in the “HRdisp.m”.
(A1) The file “HRdisp.m” was modified so that you can run the sample program with Matlab R2015a (Rev.B).
(Q2) When I ran the sample program (Rev. A) with Matlab R2015a, the animation did not appear.
(A2) The file “HRdisp.m” was modified so that you can see animation with Matlab R2015a (Rev.B).
(Q3) When I ran the sample program (Rev. A), the rover was judged to have reached the target point if its opposite side’s point was in contact with the asteroid surface. This is a bug, isn’t it?
(A3) Yes, it is. We have fixed it in the file “HRjudge.m” (Rev.B).
(Q4) Considering the size of the rover, the rover should be judged to have reached the target point if the goal point is the origin. But when I ran the sample program (Rev. A), it was not judged to have reached the target point. This is a bug, isn’t it?
(A4) Yes, it is. We have fixed it in the file “HRjudge.m” (Rev.B).
(Q5) In the contest guide (Rev. B), it reads “Values may be changed in order to make the contest more exciting.” in page 2. Which parameter may be changed?
(A5) The horizontal target position will be changed at the contest site. Other parameters will not be changed after Rev. B unless some errors are found and need to be fixed.
(Q6) The function “state = HRjudge(posT);” should be processed before the dynamics is computed, because the contact force is computed based on the current state and current state is judged by function “state = HRjudge(posT);”.
(A6) A function to check the state based on the kinematics was added before the dynamics is computed (Rev. B).
(Q7) I found a case when the state should be the STATE E (contact point is B) with the position y almost 0.5 and the attitude “the” a little negative but its state is judged as STATE C (contact point is A) (Rev. B).
(A7) It is a bug. We have fixed it in the file “set_state.m”. (Rev. C)
(Q8) I have tuned my algorithm with sample program Rev.A. Its response changes greatly with Rev.B.
(A8) The contest organizer sincerely apologizes for these inconveniences due to fixing the bugs

- Revision history

15.05.20 Contest guide and sample program files were created (Rev. A).
15.06.19 In page 8, the matlab version used at the contest was upgraded to 8.5.0(R2015a).
15.06.19 In page 9, final preparation room is fixed as Room 405.
15.06.19 In page 10, Q&As were added.
15.06.19 The sample program files were modified to fix several bugs.
The file “set_state.m” was added to check the state kinematically.
In the file “HRdisp.m”, graph size and axis limits were modified.
In the file “HRdisp.m”, figure setting was changed and pause function was called for R2015a.
In the file “HRdynmc.m”, state check (kinematics and sign of contact forces) was added.
In the file “HRdynmc.m”, size check of the friction forces was added.
In the file “HRjudge.m”, condition of reaching the target was corrected.
In the files “HR judge.m” and “HRpreset.m”, state check (kinematics) was added.
In the file “HRpreset.m”, TTRTP check was added.
In the file “HR main.m”, integration method was changed from Runge-Kutta-Gill to Euler.
In the file “calc_frc_domg.m”, the derivative of “omg” was set such that in states F and G, the “omg” was kept zero. Also the contact forces were adjusted such that in states F and G, the residual vertical velocity were removed
15.06.22 The updated contest guide and sample program files were uploaded (Rev. B).
15.06.30 In page 8, final preparation room is fixed as Room 405.
15.06.30 The sample program files were modified to fix several bugs.
In the file “set_state.m”, the check of contact point A versus B was added using attitde "the".
In the file “HRmain.m”, the initial torque command was reflected in the graph plots.
In the file “HRsplot.m”, units in the graph labels were corrected.
In the file “calc_frc_domg.m”, the contact forces (fAy,fBy) reflected the derivative of “omg” in states F and G.
15.06.30 The updated contest guide and sample program files were uploaded (Rev. C).