Design of 2D Lunar Lander using MATLAB

Lung-Hui Wu

***Abstract*—In this project, we derived the dynamic model of landing process of a 2D lunar lander. Then we designed a control law based on lander angle to stabilize the lander after impact, and it can handle nearly most of the input case.**

Keywords—lunar lander, AMEID, angle control, MATLAB

I. Introduction

Based on previous work on 1D lander, we can easily extend the model to 2D. To precisely describe a lander's physical states with rotation, we only need to add two more atttributes, l_2 and θ. And the remaining calculation and simulation is similar. The full mathematic and control theory background is referred from the given guideline [1].

II. Explanation on Files

A. *"Main_symb.m"*

This file is used to design the dynamic model, simulate the result, and calculate the direct formulas of variables for later use in "main.m". In this file we can plot the simulation result to see whether the lander without extra damping control works physically right.

B. *"Main.m"*

In this file, we directly calculate the motion by the formulas derived from "main_symb.m". And then, we create a for loop to recursively derive next physical state. We design an visualized animation and simulate the landing behavior for five seconds. In this way, we can easily see if the lander successfully lands.

III. Control Rules

Since we already obtain the information of lander angle, and in "main_symb.m" we check that for small landing angle, the lander can land safely without control. Therefore, we choose to apply PD control while the lander is in mid-air to adjust its angle to a smaller landing value. And when the lander falls near ground, we turn off the control to make sure the lander can land. Below is the PD control design:

$$\begin{cases} V_l(t) = K_p e_{rl}(t) - K_d\left(e_{rl}(t+\Delta t) - e_{rl}(t)\right) \\ V_r(t) = -K_p e_{rr}(t) + K_d\left(e_{rr}(t+\Delta t) - e_{rr}(t)\right) \end{cases}$$

, where

$$e_l(t+2\Delta t) = e_r(t+2\Delta t) = \theta(t)$$

, and $V_l(t)$, $V_r(t)$ stands for the operating voltage on the two legs. Also, we add some restriction on PD control factor as follows so the angular velocity won't get too high if the angle is relatively big:

$$K_p = \begin{cases} 0\,, & if\ z_b < 2.5; \\ 3, & if\ |\theta| > 0.65 * \pi\ and\ z_b \geq 2.5; \\ 10, & otherwise; \end{cases}$$

, and

$$K_d = \begin{cases} 0\,, & if\ z_b < 2.5; \\ 800, & otherwise. \end{cases}$$

IV. Simulation Results

In this section, we give different input states and examine the landing result. Finally, we review on the lander's robustness and stability.

A. *Reference input*

Consider the input

$$q(t) = [0(m)\ 11(m)\ 0.8(m)\ 0.8(m)\ 1.05(rad)]^T$$

$$u(t) = [0(m/s)\ 0(m/s)\ 0(m/s)\ 0(m/s)\ 0(rad/s)]^T.$$

We draw some plots to verify if it can firmly land on ground.

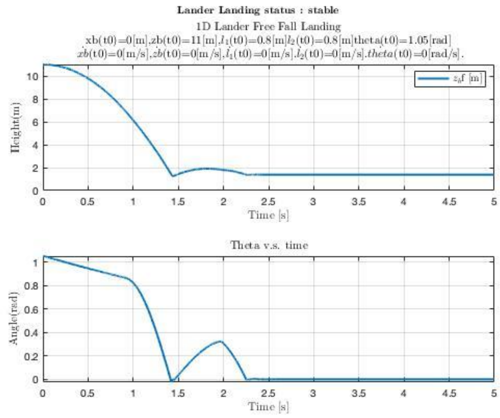

Fig. 1. The height-time plat (above) and the angle-time plot (below). We can observe that when the lander is almost landing with an approximate height of 1.5 m, the lander almost corrects its angle exactly back to zero. And after it touched the ground for the first time, it bounced and slowly landed within 1 second. The total time from falling to landed is about 2.25 seconds.

B. *Analysis on the performance*

We give different angle inputs to the system, and analyze whether it lands. If it lands, we show the total landing time.

$\theta(t_0)$ (rad)	Landed or not (T: true, F: false)	Landing Time (sec)
0.1π	T	1.35
0.2π	T	1.45
0.3π	T	1.43
0.4π	T	1.75
0.5π	T	4.20
0.6π	F	X
0.7π	T	1.49
0.8π	F	X
0.9π	T	2.35
π	T	2.81

Fig. 2. The performance on different angle inputs. We can observe that in most of the cases the lander can land within 5 seconds. However, in few cases, the angle control cannot reduce the angle to a small value due to overshoot, so the falling angle might not be stable.

V. DISCUSSION

The overall performance on this design is still not optimal, since the overshoot issue is not properly solved. The best method to solve this problem might be a I controller, because we want to rotate the lander by an angle eventually. However, designing a great I controller that rotates a fixed angle in a short time is very costly in time or power. Our design is still feasible and efficient if we neglect the special cases of angle, as it can quickly land the lander within 5 seconds.

VI. REFERENCES

[1] Cheng-Wei Chen, Yi-Lun Hsu, et al. Control Systems: Final Project.

[2] HARA, Susumu, et al. Momentum-exchange-impact-damper-based shock response control for planetary exploration spacecraft. Journal of Guidance, Control, and Dynamics, 2011, 34.6: 1828-1838.

[3] KUSHIDA, Yohei, et al. Robust landing gear system based on a hybrid momentum exchange impact damper. Journal of Guidance, Control, and Dynamics, 2013, 36.3: 776-789.

[4] Cloutier, James R. State-dependent Riccati equation techniques: an overview. Proceedings of the 1997 American control conference (Cat. No. 97CH36041). Vol. 2. IEEE, 1997.