Control on a Lunar Lander

Weiche Tseng
College of Electrical Engineering
National Taiwan University
Taipei, Taiwan
e-mail: b07901165@ntu.edu.tw

***Abstract*—Spacecraft landing is a critical issue when it comes to space exploring. This paper introduces a PD-control landing system, and the control timing for it.**

Keywords—PD control, lunar lander, AMEID, actuate timing

I. Introduction

This paper will cover how I use a PD controller to stabilize the landing. Also, several AMEID actuating conditions would be covered, from the simplest to a more complex one. And more and more critical initial states would have a successful landing with these actuating conditions. In addition, we will talk about our ideas when tuning the PD constants and their suitable value for each AMEID actuating condition. Finally, we will conclude how to trade-off between power, landing convergence speed, and other specifications in concern.

II. PD-Control Model

In our PD-Control, we use theta, the angle of our lander w.r.t. a horizontal line, as our reference. The P-Controller here is to make our response faster, and the D-Controller is to shorten our convergence time. Through adequate tuning, these two together will make our landing faster, and with less oscillation.

Note that we don't use an I-Controller in this project. Due to physical principles, landing horizontally on the ground is the only steady state. Otherwise, the lander would be upside-down with its footpads oscillating, or even exceed the observed space. Both cases are unstable. Once our lander is stabilized, there is no need to fix a steady-state error. Thus, we don't have to apply I-Controller to our design.

Also note that our PD-Controller and the subsequent AMEID system will work only when our launcher is actuated. Otherwise the AMEID output force vector (*hr* or *hl*) will be 0, and there will be no control. The actuate condition is described in part A below. And we will show our control model in part B.

A. AMEID Actuate Condition

Modifying TA's algorithm, we also actuate our launcher when the lander touches the ground. But we determine in a slightly different way. TA determines whether the lander touches the ground by checking if *zb_dot* changes its direction. However, there exist a case where *zb_dot* changes its direction without touching the ground. For example, the lander bumps from the ground, rise to the air, and then fall down. This algorithm may cause mis-launch in our 2D model.

Thus, in our algorithm, we simply check if *z1* is no more than zero. If it does, it means that Footpad1 (the **left** one) has touched the ground, so we actuate our **right** AMEID system with its PD-Controller to launch the **right** plate. As for *z2*, just perform similar process, with "left" and "right" exchanged in the above statement.

We also consider the case when the two Footpad touches the ground simultaneously. If the right plate isn't launched (i.e. xmr(2) = 0), then we don't launch the plate. Same rules are applied to the left plate. The main consideration is to save the chance for the later bumpy duration. Another consideration is that if the lander is already landing horizontally, further launching would cause unwanted force to our footpad, and thus more bumping.

For harsh initial condition, we also develop a unique actuate strategy. If *theta* is too large during touchdown, chances are that our AMEID force cannot rotate the body back in time. even if we do succeed, the AMEID voltage input would be too large to implement in reality. Thus, we need our launcher actuated earlier so that *theta* would be more moderate right before touchdown.

The specific algorithm goes like this:

If *z1 > |zb_dot|*buffer_time + 0.5*g*buffer_time^2* and *xpr < L_stroke*, actuate the right AMEID.

The same algorithm is also applied on Footpad2 and the left AMEID.

The idea is quite intuitive. Since we want the lander to rotate in the air, we need to set a height buffer for it. Through simulation, we also found that it takes less than *buffer_time* in average for *theta* to decline to acceptable range. So, we calculate how far the lander would drop in *buffer_time*; such distance is what we need for our height buffer.

Cases with initial angular velocity (a.k.a. *theta != 0*) can also occur. So we also actuate the right AMEID when *theta > 0 and theta_dot > 0.* And actuate the left one with symmetric condition.

In conclusion, the right AMEID actuating timing is described as the following pseudo code:

Actuate Condition for the Right AMEID
1. if (z1 <= 0 ||
2. (z1 > |zb_dot|*buffer_time + 0.5*g*buffer_time^2 && theta > theta_threshold) ||

```
(theta > 0 && theta_dot > 0) ){
  if ( xpr < L_stroke ){
      Launchr = TRUE; }
  else{
      Launchr = FALSE;}
{
```

Similar (symmetric) algorithm is applied to the left AMEID.

In the rest of this paper, this actuate strategy would be addressed as the AAITA (AMEID Actuated In The Air) method.

B. *Ideal Continuous Control Model*

Here is our control block graph.

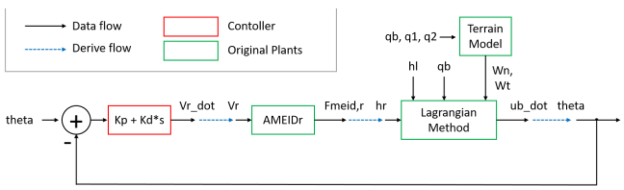

Fig. 1. Control flow of the right AMEID

As shown in Fig. 1, we compose a PD-Controller and the original plants together to form a feedback control system. And as mentioned before, we take *qb(5) = theta* as the reference.

The control flow of the left AMEID is similar to its right counterpart. The only difference is that we take – *theta* instead of *theta* as the reference. This is quite intuitive since the left and the right structure are symmetric.

Note that what Fig. 1 shows is a continuous time control system in the ideal world. In simulation, every argument should be discretized.

III. Simulation and Results

To decide on the value of our control constants, we first need a moderate initial condition, one with *zb* not too fast and *theta* not too large. Actually, we choose the initial condition

$$qi = [0\ 6\ 0.8\ 0.8\ pi/6] \quad \text{,and}$$

$$ui = [0\ {-5}\ 0\ 0\ 0]$$

for our first tuning. The result is described in part A below.

After our first tuning, we test our control system further with various initial conditions. Then we adjust our control constants (especially *Kp*) to accommodate harsh conditions. The result is described in part B below.

A. *First Tuning*

When tuning, we found that a D-Controller would result in high AMEID voltage input, which is hard to implement in reality. So we will discuss two cases below, one without considering reality limitations, and another with such consideration.

1) For unlimited voltage supply: As shown in Fig. 2, with rise of *Kp*, we have a faster *theta* response. But we also get a fluctuated *zb* during touchdown. This is because our AMEID force is too much for the footpads and causes bouncing. Eventually, we choose *Kp* to be 500.

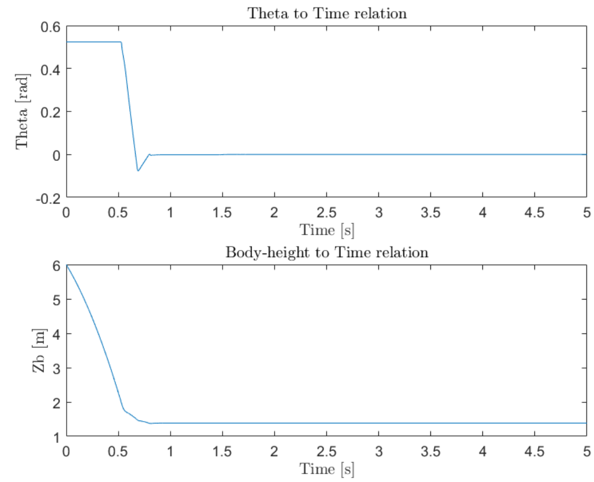

Fig. 2. (a) theta and body-height for *Kp=100, Kd=12*

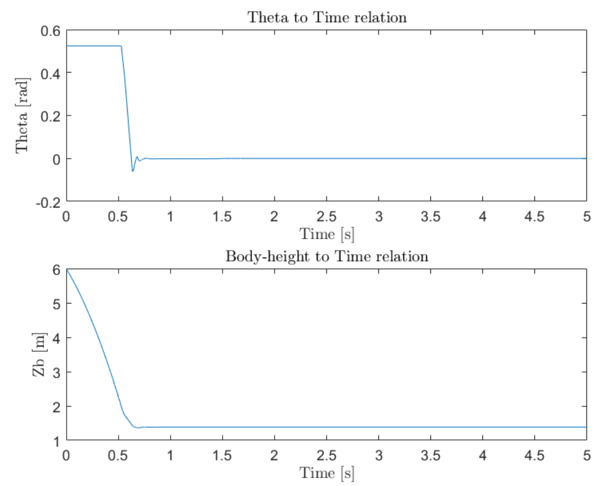

Fig. 2. (b) theta and body-height for *Kp=500, Kd=12*

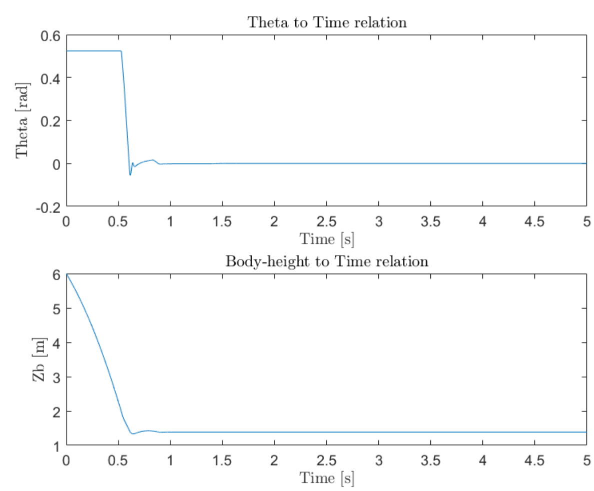

Fig. 2. (c) theta and body-height for *Kp=1000, Kd=12*

We also have some trade-off in tuning *Kd*. Fig. 3. display the result of *Kp=100, Kd=5* and *Kp=100, Kd=50.* We found that our convergence time of *theta* decreases when *Kd* increases. But there is a bouncing problem similar to the case with *Kp*. And there is also a significant undershooting when *Kd* is too small or too large. The large case does not quite agree with what we've learnt in class, probably for some reason similar to the bouncing problem.

Eventually, we choose *Kd = 12.*

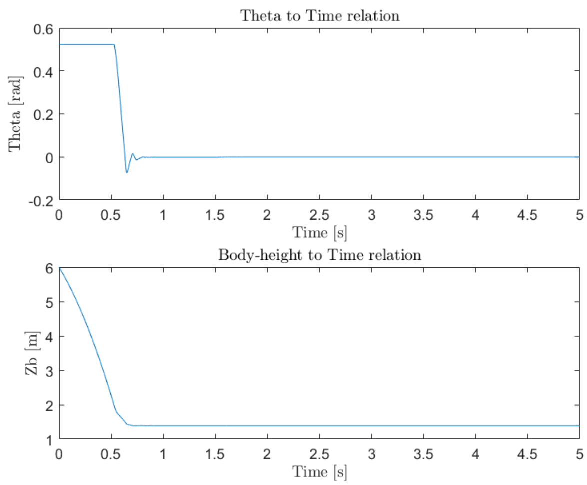

Fig. 3. (a) theta and body-height for *Kp=500, Kd=5*

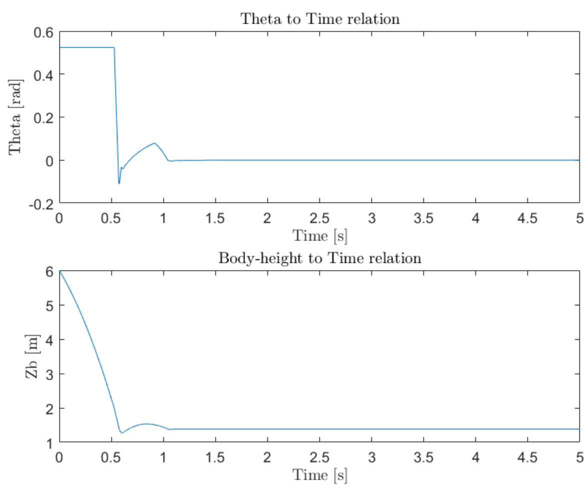

Fig. 3. (b) theta and body-height for *Kp=500, Kd=50*

2) For limited power supply: TABLE I shows the peak input votage of the right AMEID for different *Kp* and *Kd*. Apparently, large *Kd* will lead to extremely large AMEID input voltage. Considering power consumption, *Kd* should not exceed *1* in our first tuning. As for *Kp*, we observe that it has little impact on the input voltage. So we maintain the result of the ideal cases in part(1).

TABLE I. Vr FOR DIFFERENT CONTROL CONSTANT

Vr (V)		**Kd**			
		1	***5***	***12***	***50***
Kp	***100***	1100	5288	12620	52410
	500	1309	5498	12830	52620
	1000	1571	5760	13090	52880

Note that in TABLE I, we only show values of the right AMEID input voltage. The left one in this case is much smaller. Eventually, we take *Kp = 100, Kd = 1* for reality concerns.

B. Further Tuning

After first tuning, let us consider more general cases. We conduct simulations of our control with different initial *theta* and *zb_dot*, then adjust *Kp* and *Kd* in these cases. The results are shown below.

1) For different initial theta: Without AMEID, if the initial *theta* exceeds 52 degree, the lander would not land succesfully. Thus, in this section we will discuss the case with large initial *theta*. For *theta* that is less than 52 degree, we try to improve its landing; for those more than 52 degree, we try to achive a landing success. The initial values of arguments other than *theta* is the same as in part A.

After a procedure similar to our first tuning, we decided that *Kp = 1000.* As for *Kd*, if there is unlimited voltage supply, we choose *Kd = 25*. However, if there is limited voltage supply, we should not use a D-Controller, and there would be a more bumpy (some swing in *theta* and slow convergence) touchdown. The results are shown in Fig. 4. And the corresponding peak input voltage of the right AMEID is shown in TABLE II.

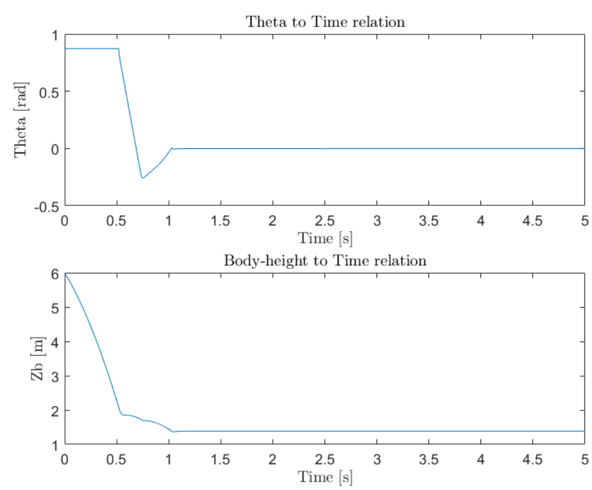

Fig. 4. (a) theta and body-height for *Kp=1000, Kd=12, theta=50 degree*

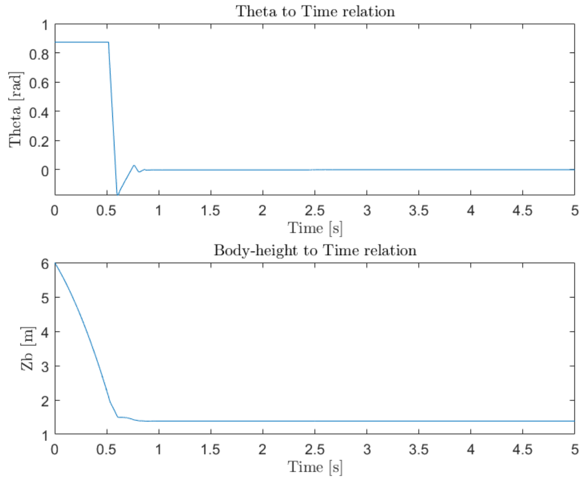

Fig. 4. (b) theta and body-height for *Kp=1000, Kd=25, theta=50 degree*

TABLE II. Vr FOR DIFFERENT Kd, WITH Kp = 1000

	Kd			
	1	***12***	***25***	***50***
Vr(V)	2618	21820	44510	88140

With *Kp = 1000, Kd =25*, the maximum initial *theta* value leading to a successful touchdown is still 61 degree.

2) For different initial zb_dot: In this case we test our control with initial *zb_dot* = *-25*. Other parameters are the same as in part A. As shown in Fig. 5(a), the control constants in part A still works. However, there is an obvious bouncing in *theta*. To flatten our *theta*, we raise *Kp* to 2000. And as we can see in Fig. 5(b), the change of *theta* is lowered down. Also, we found that *Kd* has little impact on the landing performance here. So we choose not to use a D-controller in this case. In Fig.5, note that with such high speed, a significant bump up is inevitable. And our job here is to make *theta* close to zero after the lander bumps up.

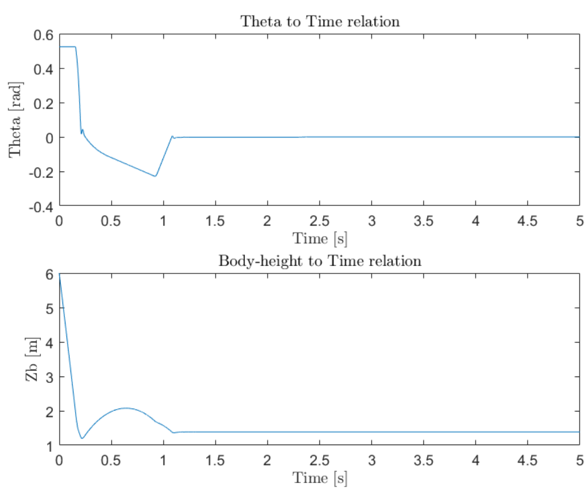

Fig. 5. (a) theta and body-height for *Kp=500, Kd=12*

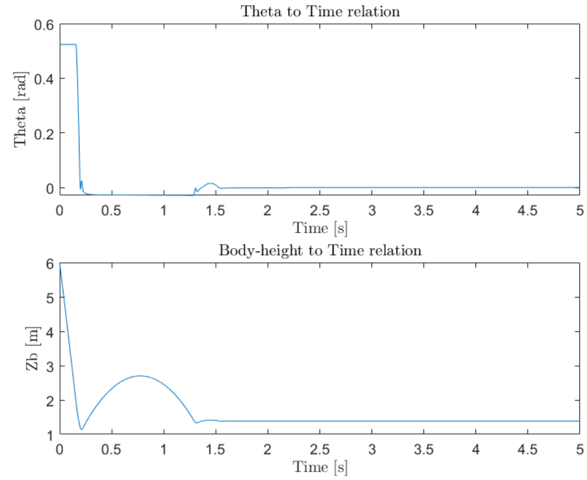

Fig. 5. (b) theta and body-height for *Kp=2000, Kd=12*

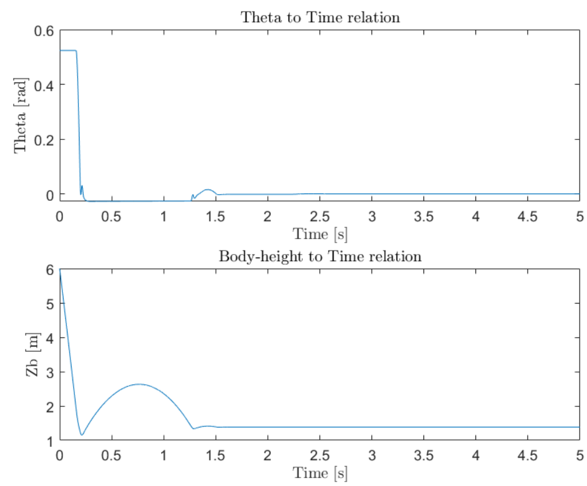

Fig. 5. (c) theta and body-height for *Kp=2000, Kd=0*

C. Landing Performance with the AAITA Method

As stated before, the AAITA (AMEID Actuated In The Air) Method launch the AMEID plate in advance for large *theta* or *theta_dot*.

1) Tuning with large theta: As stated before, the maximum theta for safe landing without AMEID is 52 degree. So we set *theta_threshold = 52*pi/180.*

After tuning, we found that we don't need high value of *Kp* and *Kd* to mild the bumping or accelerate theta convergence as we did in part B. In fact, large *Kp* might lead to an "overturn" problem: The AMEID force would be so large that theta goes from positive to negative or the other way around. Thus, we finally decided to maintain the value of *Kp* and *Kd* in part A, which is *Kp = 200* and *Kd = 5*. We also decided that *buffer_time = 0.7 sec*.

With AAITA, even cases with incredibly large *theta* can have a comfortable (quick convergence and less swing) landing. The results of *theta = 60* and *theta = 90* are shown in Fig. 6.

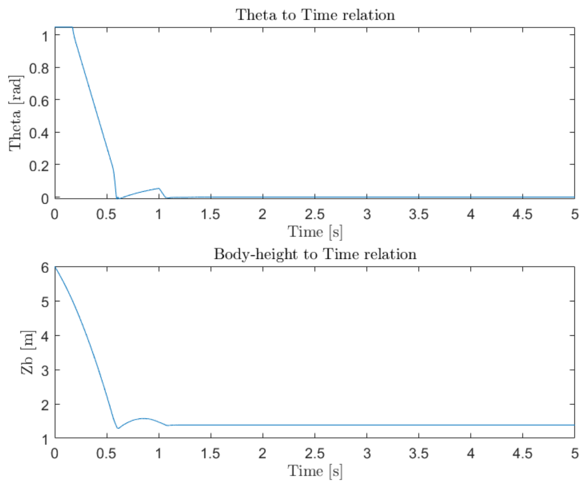

Fig. 6. (a) theta and body-height for *theta = 60 degree*

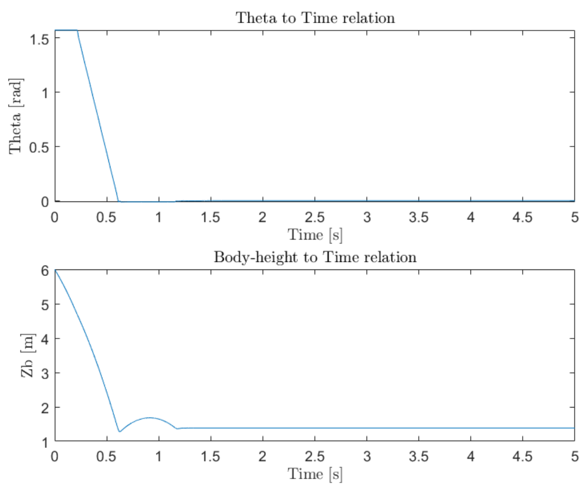

Fig. 6. (a) theta and body-height for *theta = 90 degree*

2) Tuning with theta_dot: In our simulation, we found that a small *theta_dot* may require a large torque to hold it back. Thus, we decided to raise *Kp* up. Some initial *theta_dot* and the required *Kp* are displayed in TABLE III. (Here "required" means "< minimum-success value + 100".) And an example of *theta-time relation* and *zb-time relation* is shown in Fig. 7. (with initial *zb_dot=0, theta=0, theta_dot=5, Kp=3200, Kd=5*) As we can see in Fig. 7, such control would decrease and fix the slope significantly when our lander is still in the air. Thus, *theta* and *theta_dot* would be less critical during touchdown.

However, as described in the last part, such high value of *Kp* might cause an "overturn" problem. So we lower *buffer_time* to *0.1 sec*. Note that this adjust would only impact on the actuate timing with large *theta*, since we don't consider height buffer for the case with *theta_dot*. After this adjust, our landing would be more violent than in the last part. But the simulation result shows that our control still works.

TABLE III. REQUIRED Kp FOR DIFFERENT THETA_DOT (Kd=5)

	THETA_DOT			
	1	***3***	***5***	***10***
Kp	100	1200	2800	6600

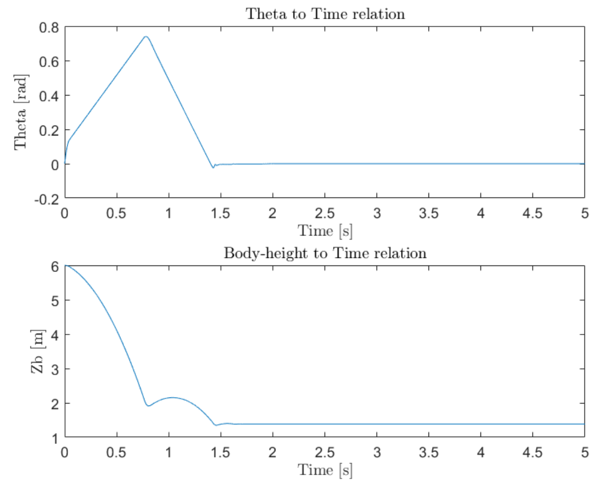

Fig. 7. theta and body-height for *theta_dot = 5 rad/sec*

Surprisingly, the input voltage of AMEID decreases significantly compared to the result in part A. For instance, with *Kd = 5*, the peak value of *Vr* is more than 5000 volts in our first tuning. However, it only takes 221.9 and 23.27 volts for *Vr* and *Vl* respectively when using AAITA with same value of *Kd* and even larger *Kp*. This implies that AAITA with *theta_dot* is more energy economic.

IV. CONCLUSION

Comparing the result of our first and further tuning, we found that the larger the initial *theta*, *zb_dot* is, the larger *Kp*, *Kd* we need. Large *Kp* can decrease *theta* faster (a.k.a. reduce rise time), while *Kd* can cause less oscillation (a.k.a. reduce convergence time). However, such large *Kp* and *Kd* might lead to unnecessary bouncing when *theta* is not **that large**. Even worse, large *Kd* would give rise to extremely large AMEID input voltage which is hard to implement in reality. Thus, if we can make sure that *zb_dot* and *theta* are in specific small range (add some landing repulsor to our lander, for example) , then we can use rather small values of *Kp* and *Kd*, as we did in the first tuning.

On the other hand, part C of the simulation section shows us the significance of the AMEID actuating timing. Should we consider critical initial conditions, such as large *theta* and *theta_dot*, AAITA would be a good choice. With accurate timing, AAITA can succeed in cases that touchdown-actuate system can't. Furthermore, dealing with same cases, AAITA allows smaller value of *Kp* and *Kd* than touchdown-actuate system does, and therefore require lower voltage supply.

REFERENCES

[1] 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.

[2] 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.

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