Pattern Recognition Control Design

Elisabeth A. Gambone*

NASA Johnson Space Center, Houston, Texas

Georgia Institute of Technology, Atlanta, Georgia

Spacecraft control algorithms must know the expected vehicle response to any command to the available control effectors, such as reaction thrusters or torque devices. Spacecraft control system design approaches have traditionally relied on the estimated vehicle mass properties to determine the desired force and moment, as well as knowledge of the effector performance to efficiently control the spacecraft. A pattern recognition approach was used to investigate the relationship between the control effector commands and spacecraft responses. Instead of supplying the approximated vehicle properties and the thruster performance characteristics, a database of information relating the thruster firing commands and the desired vehicle response was used for closed-loop control. A Monte Carlo simulation data set of the spacecraft dynamic response to effector commands was analyzed to establish the influence a command has on the behavior of the spacecraft. A tool developed at NASA Johnson Space Center to analyze flight dynamics Monte Carlo data sets through pattern recognition methods was used to perform this analysis. Once a comprehensive data set relating spacecraft responses with commands was established, it was used in place of traditional control methods and gains set. This pattern recognition approach was compared with traditional control algorithms to determine the potential benefits and uses.

I. Introduction

Spacecraft control is typically achieved using phase plane and proportional-integral-derivate (PID) controllers that require knowledge of the vehicle mass properties and effector performance in order to calculate state errors and effector commands. Acquisition of this knowledge can be difficult during spacecraft operation. As a result, a spacecraft control system has been designed that does not require knowledge of vehicle mass properties or thruster information, and remains robust to uncertainties. The resulting pattern recognition control logic uses a database of thruster firing times and corresponding rate changes to command thruster firings for rotational control. This method of control allows for a simplistic implementation and more flexibility when performing maneuvers. A Monte Carlo analysis was performed to verify the robustness of the database to varying mass properties. The Tool for Rapid Analysis of Monte Carlo Simulations (TRAM) was utilized to identify the inputs and thruster commands that have the largest impact on failures. The control method was tested by simulating multi-axis maneuvers with varying maneuver rates, as well as attitude holds with large external disturbances. An adaptive controller can be designed that updates the database in response to thruster failures or performance changes. Similarly, stochastic programming can be used to deduce efficient firing patterns for performing maneuvers.

II. Methodology

A generic spacecraft model was built using Simulink, a graphical programming environment for system modeling and simulation. The guidance logic analyzes inertial attitude and attitude rate commands and passes them to the control system. The original controller uses a phase plane that compares the spacecraft attitude error and rate error to predetermined limits to calculate thruster firing times. The phase plane is adapted from the Morpheus Lander roll control system.[1] The firing commands are sent to the propulsion model, which generates the thruster forces and torques. The spacecraft has 16 thrusters that can be used for

[*] Aerospace Engineer, Integrated GN&C Analysis

rotational control. The resulting force and torque are passed to six degree of freedom equations of motion in order to produce changes in the vehicle state. The spacecraft is assumed to be in Lunar orbit for this analysis. The vehicle state is read by the navigation system in order to calculate the attitude and rate error. Table 1 shows the assumed mass properties of the spacecraft, where CG is center of gravity and I is moment of inertia. Table 2 shows the designed firing direction of each thruster. The model was used to simulate rotational maneuvers for this analysis.

The spacecraft model was verified by commanding a five degree maneuver in each axis. The phase plane maintains control throughout the maneuver. The thrusters fire in response to the initial rate error, and the resulting vehicle rate maneuvers the spacecraft to the desired attitude. The thrusters fire to arrest the vehicle rate once the attitude error has converged. The maneuver is shown in Figures 1 and 2.

Parameter	Unit	Value
Mass	kg	3505.00
$\mathrm{CG_x}$	m	1.41
$\mathrm{I_{xx}}$	kg-m^2	3583.09
$\mathrm{I_{yy}}$	kg-m^2	3160.22
$\mathrm{I_{zz}}$	kg-m^2	2457.06
Thrust	N	22.24

Table 1. Spacecraft mass properties.

Jet	Roll	Pitch	Yaw
1	-1	0	0
2	1	0	0
3	-1	0	0
4	1	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	-1	0
10	0	1	0
11	0	-1	0
12	0	1	0
13	0	0	-1
14	0	0	1
15	0	0	-1
16	0	0	1

Table 2. Spacecraft jet mapping.

The pattern recognition control logic was designed by running simulations for 26 unique attitude rate commands. The simulations include combinations of positive, negative and zero rate commands in roll, pitch, and yaw, for a 0.3 degrees per second total rate command for each run. The simulations were run until thrusters were fired to reach the desired rate. The spacecraft altitude was selected to minimize the effects of gravity on thruster firings. A database was then created comparing the rate commands to the firing times. A script was developed that saves the firing time per thruster for each run, as well as the rate settling time after firing is complete. The 26 rate commands are shown in Table 3, and the corresponding thruster firing times are shown in Appendix A. Thrusters 5 to 8 are not used and are therefore not included in the table.

Simulations were then performed with the 26 rate combinations as the initial rates and a zero rate command. These runs confirmed that damping the 26 rate commands requires equal firing times with the opposing thrusters. The database can therefore be used for both maneuvers and rate damping. A comparison of using the 26 rate commands as the initial and final rates is shown in Figures 3 and 4.

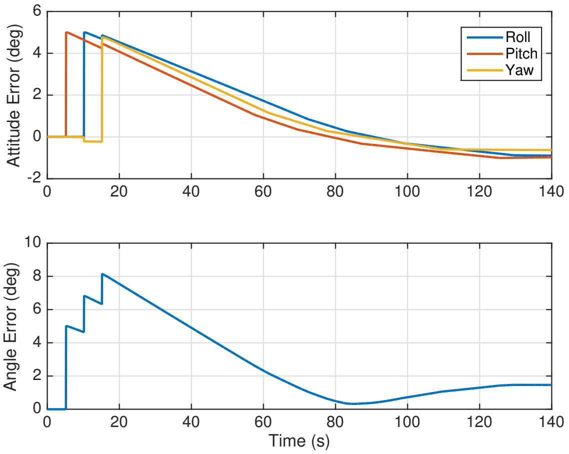

Figure 1. Rotational maneuver attitude error.

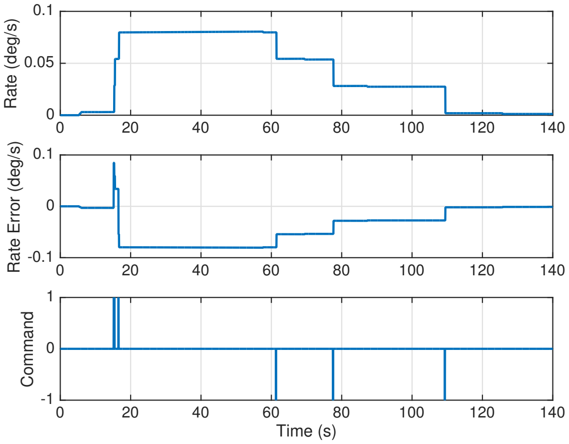

Figure 2. Yaw rate and thruster commands.

Case	Roll	Pitch	Yaw
1	-0.1732	-0.1732	-0.1732
2	-0.2121	-0.2121	0
3	-0.1732	-0.1732	0.1732
4	-0.2121	0	-0.2121
5	-0.3	0	0
6	-0.2121	0	0.2121
7	-0.1732	0.1732	-0.1732
8	0.2121	0.2121	0
9	-0.1732	0.1732	0.1732
10	0	-0.2121	0.2121
11	0	-0.3	0
12	0	-0.2121	0.2121
13	0	0	-0.3
14	0	0	0.3
15	0	0.2121	-0.2121
16	0	0.3	0
17	0	0.2121	0.2121
18	0.1732	-0.1732	-0.1732
19	0.2121	-0.2121	0
20	0.1732	-0.1732	0.1732
21	0.2121	0	-0.2121
22	0.3	0	0
23	0.2121	0	0.2121
24	0.1732	0.1732	-0.1732
25	0.2121	0.2121	0
26	0.1732	0.1732	0.1732

Table 3. Rate commands in degrees per second.

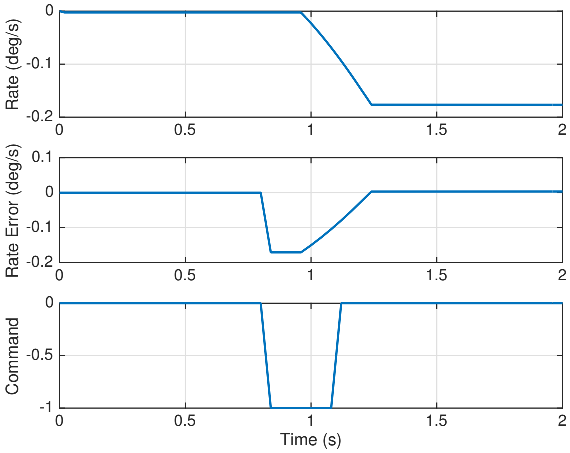

Figure 3. Thruster firing for rate command.

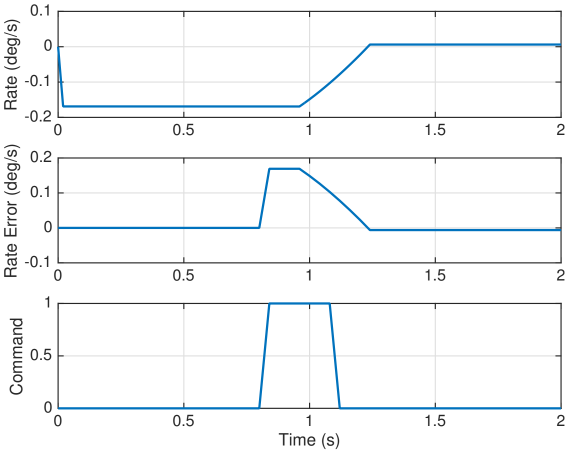

Figure 4. Thruster firing for rate damping.

The pattern recognition control logic first reads the lookup table to determine the thrusters firing in each axis, as well as the indices of opposing thrusters. The current attitude error and rate error are calculated, and the rate error is compared to the lookup table to determine the firing time scale factor in each axis. If the total angle error is above a set limit, the vehicle rate is compared to a maneuver rate to determine the rate error. The total maneuver rate is separated into each axis based on the current attitude error. If the angle error is below the limit, the vehicle rate is compared to the commanded rate. If the calculated rate error is below a set limit, the rate error is set to zero for comparison with the lookup table. This type of dead band control prevents thrusters from firing for small errors.[2] The maneuver rate should therefore be at least three times the rate error limit. The rate error limit should be large enough to account for the rate change due to a minimum impulse thruster firing. The logic loops through the available thrusters, and if the remaining firing time of the current thruster and its opposing thruster are below a set limit, a new firing time is calculated by multiplying the firing time in the lookup table with the scale factor for the current axis. A hold time was added to the logic to allow the rate to settle before another firing is commanded. The on time is then decremented by the control rate before the next control cycle begins. The control rate for the spacecraft model is set to 40 milliseconds. The on time limit was set to slightly less than the control rate. The inputs used for the pattern recognition logic are shown in Table 4. The new logic requires fewer inputs than the phase plane logic, and does not require knowledge of the jet mapping shown in Table 2.

Parameter	Unit	Value
Maneuver Limit	deg	0.6
Maneuver Rate	deg/sec	0.2
Rate Error Limit	deg/sec	0.06
On Time Limit	sec	0.03
Hold Time	sec	0.16

Table 4. Pattern recognition control logic inputs.

III. Results

The pattern recognition control logic was added to the spacecraft model in place of the phase plane logic. The new control logic was verified by commanding single axis maneuvers. The simulations require two equal and opposite thruster firings to complete the maneuver. An initial firing is commanded to reach the maneuver rate, and a final firing is commanded to zero the rate once the angle error falls below the maneuver limit. This behavior is shown in Figures 5 and 6 for a yaw maneuver. Multi-axis maneuvers were also simulated to compare the controller performance to the phase plane logic. The new control method has similar performance and thruster firings to the phase plane, but has more robustness and flexibility. A five degree maneuver in each axis is shown in Figures 7 and 8. Additional simulations are shown in Appendix C for large maneuvers and varying maneuver rates.

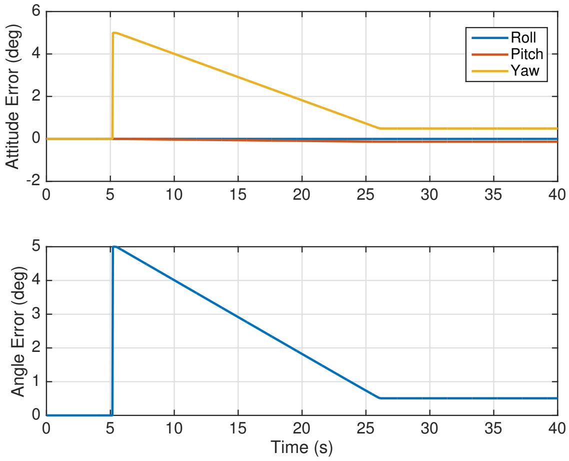

Figure 5. Single axis maneuver attitude error.

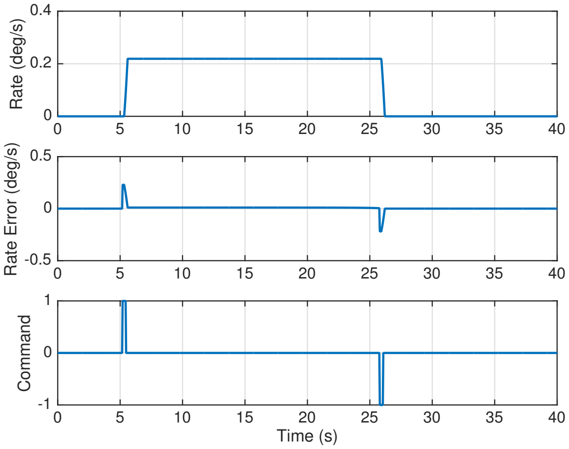

Figure 6. Pattern recognition thruster commands.

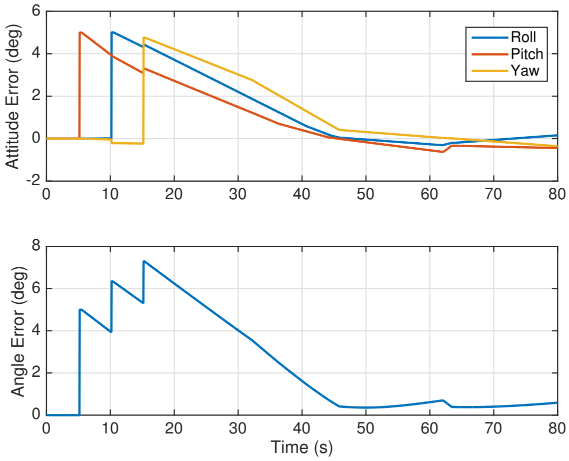

Figure 7. New rotational maneuver attitude error.

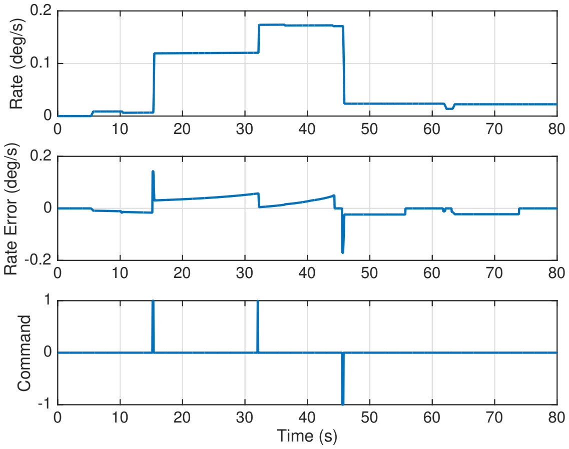

Figure 8. New yaw rate and thruster commands.

A Monte Carlo analysis was then performed by varying the mass properties and thruster orientations. The three-sigma values used to generate normal Monte Carlo inputs are shown in Table 5. The Monte Carlo dispersions are adapted from those used to verify the Morpheus Lander flight control system.[3] Other potential uncertainties in the model include the delay time between the sensors and actuators, the firing delay, and the propellant flow. The Monte Carlo analysis assessed the robustness of the firing times in the lookup table, which led to some modifications to the database.

TRAM is a software developed at NASA Johnson Space Center for identifying patterns in large data sets.[4] The tool uses kernel density estimation to identify single variables and combinations of variables that influence failures.[5] TRAM was used to assess the relationship between Monte Carlo dispersions and thruster firing times. Failed cases were those that did not reach a near zero rate error at the end of the run.

Parameter	Unit	3σ
Mass	kg	73.60
$\mathrm{CG_x}$	m	0.01
$\mathrm{I_{xx}}$	kg-m^2	322.48
$\mathrm{I_{yy}}$	kg-m^2	284.42
$\mathrm{I_{zz}}$	kg-m^2	221.13
Thrust	N	2.00
Thruster angle	deg	5

Table 5. Monte Carlo dispersions.

A Monte Carlo analysis was performed for the 26 rate commands with both the original and pattern recognition control logic. Running the phase plane logic with dispersed inputs showed how some yaw rate commands have varying firing times, with one firing time having fewer failed runs than others. The lookup table was therefore updated for two yaw cases. Similarly, running a Monte Carlo with the pattern recognition logic identified runs that require multiple firings to reach the desired rate. If the rate error for these runs did not converge, the firing time was updated in the lookup table. This method resulted in updated firing times for 12 additional yaw cases. Monte Carlos with the updated lookup table require one thruster firing to reach the desired rate, and have a near zero rate error at the end of the run. Updating the lookup table based on Monte Carlo results can be automated by measuring the number of firings and the final rate error to determine if the firing time in the lookup table is robust to dispersions. An example of the Monte Carlo analysis before and after updating the lookup table is shown in Figures 9 and 10. The updated thruster firing times are shown in Appendix B.

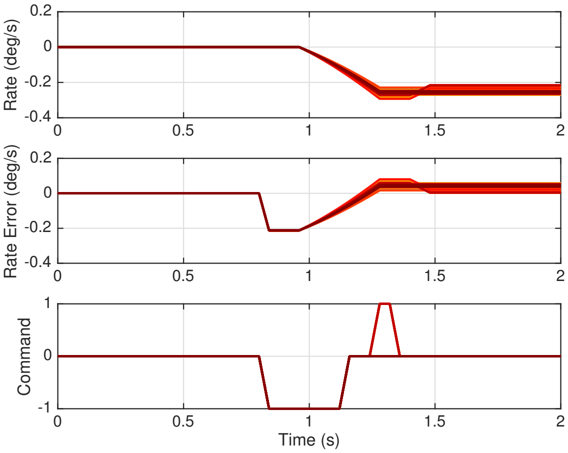

Figure 9. Monte Carlo thruster firing.

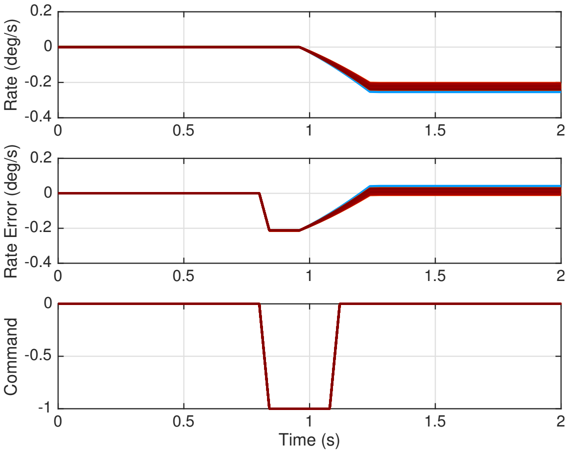

Figure 10. Updated Monte Carlo thruster firing.

An analysis was also performed for the single axis maneuvers with the updated pattern recognition logic. Running a Monte Carlo for the single axis maneuvers showed how the control logic adapts to dispersions despite having no knowledge of vehicle mass properties or effector performance. The yaw maneuver Monte Carlo is shown in Figures 11 and 12. The Monte Carlo dispersions were then increased to up to three times the original inputs. The pattern recognition logic maintains control throughout the maneuver by commanding additional thruster firings, as shown in Figures 13 and 14. A multi-axis maneuver was also run with off-nominal inputs, with similar performance to Figures 7 and 8.

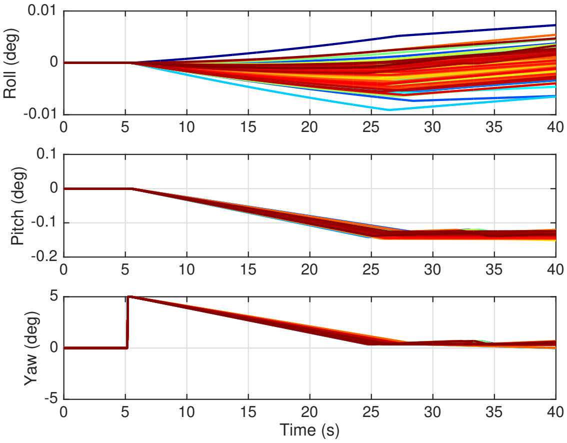

Figure 11. Monte Carlo maneuver attitude error.

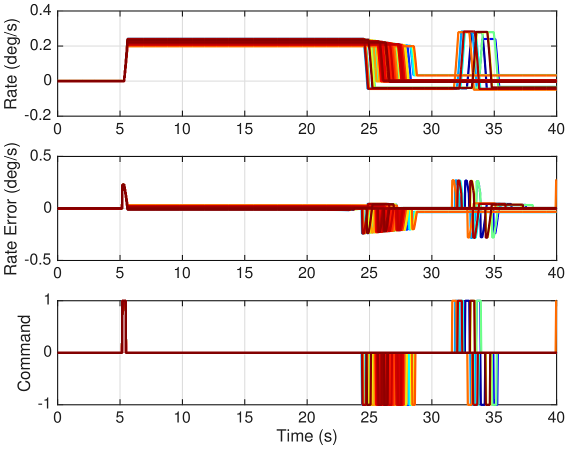

Figure 12. Monte Carlo yaw rate and thruster commands.

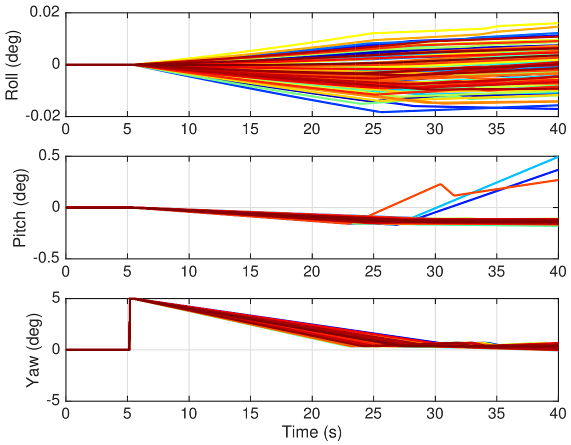

Figure 13. Stress maneuver attitude error.

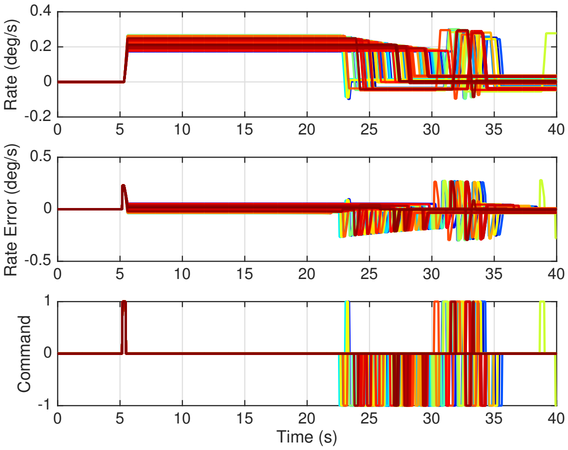

Figure 14. Stress yaw rate and thruster commands.

The pattern recognition software TRAM was used to identify the inputs that have the largest impact on failure probability. The software used the k-Nearest Neighbors algorithm to generate maps of the variables of interest based on failures.[6] The analysis showed that thrust and moment of inertia in the firing direction were closely related to failures, and total thrust has more of an impact than thruster orientation. Available thrust therefore influences rotational control accuracy more than thruster positioning. Moment of inertia versus thrust for the yaw rate case from Figure 9 is shown in Figure 15. Red dots are runs that did not reach a near zero rate error, and blue dots are runs that converged. The corresponding histogram of rate error bins is shown in Figure 16.[7] The larger rate errors correspond to failures in Figure 15. For this case, runs with large dispersions require multiple thruster firings, and are therefore more likely to converge. Reducing the rate error limit from 0.06 to 0.04 degrees per second eliminates the failures, but increases the number of runs with multiple thruster firings. Updating the thruster firing times in the lookup table eliminates the failures with one firing per run, as shown in Figure 10.

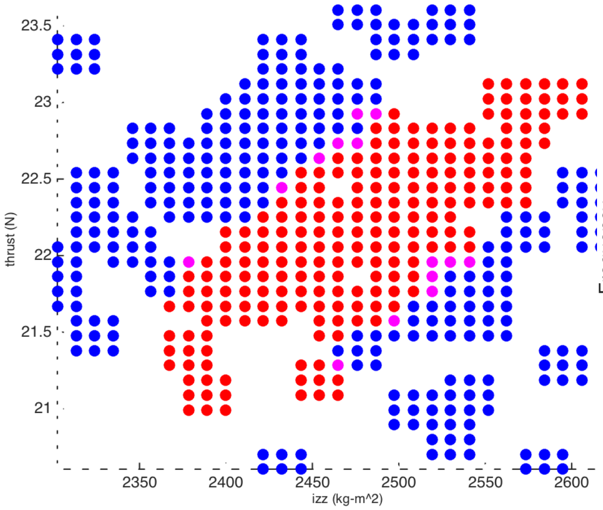

Figure 15. Monte Carlo input failure probability.

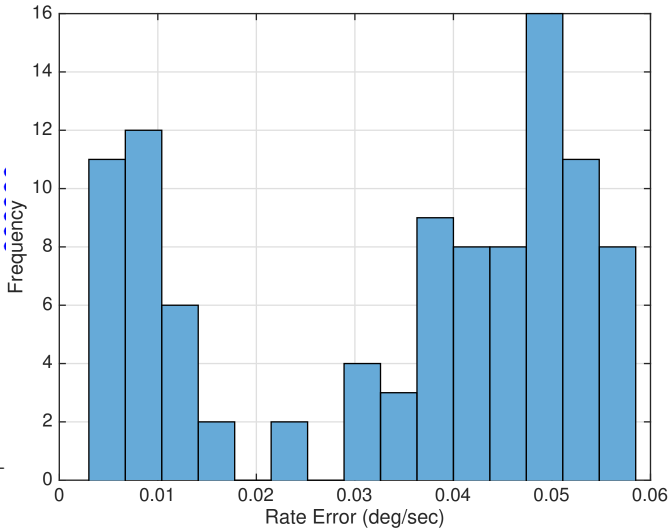

Figure 16. Rate error probability.

The model was run with a large external disturbance in order to verify the attitude hold capabilities of the pattern recognition logic. The disturbance causes the angle error to reach the maneuver limit, and a thruster firing is commanded to build maneuver rate. However, the angle error immediately falls below the limit, and an opposite firing is commanded to null the rate. This behavior is shown in Figure 17. The control logic was therefore modified to allow a no firing region between the maneuver limit and the attitude hold limit. The maneuver rate then brings the spacecraft further inside the attitude hold region before the rate is arrested. The no firing region was set to 0.1 degrees above and below the maneuver limit. This updated behavior is shown in Figure 18. A multi-axis maneuver with an external disturbance was also simulated to verify the control capabilities of the pattern recognition logic. This maneuver is shown in Figures 19 and 20.

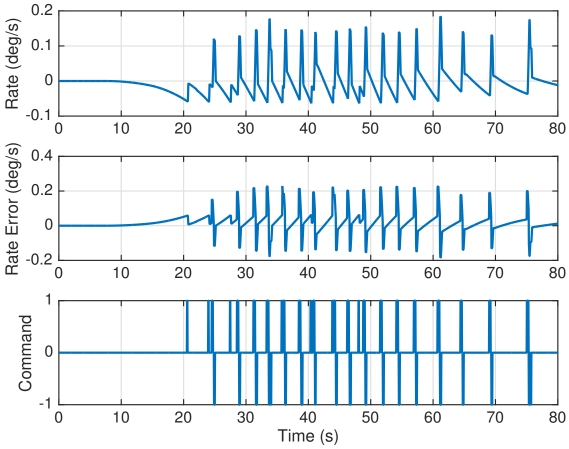

Figure 17. Attitude hold thruster commands.

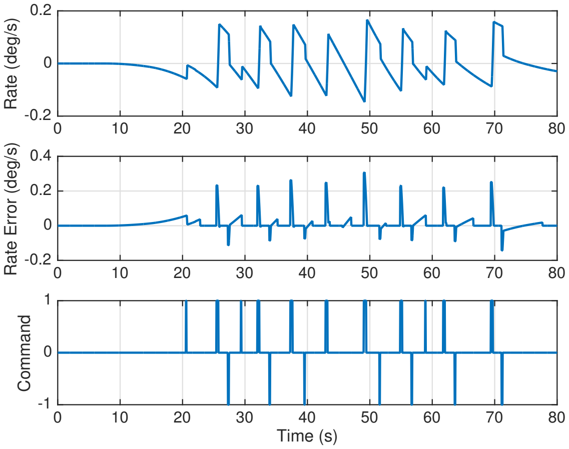

Figure 18. Modified attitude hold thruster commands.

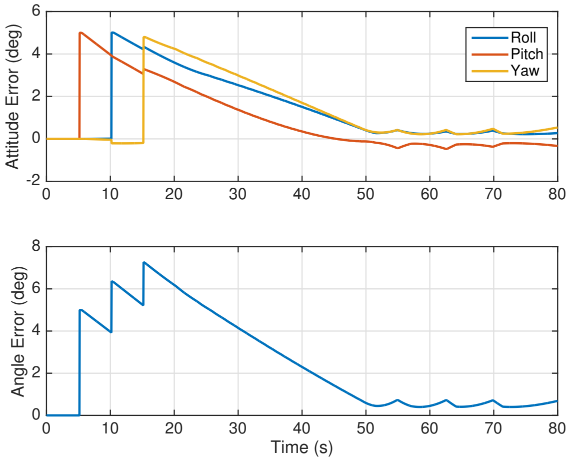

Figure 19. Disturbed maneuver attitude error.

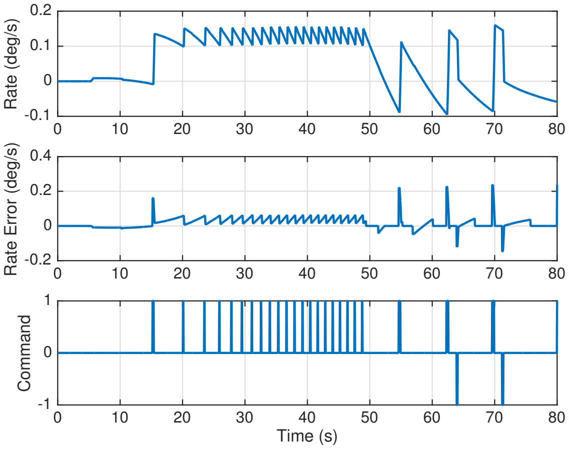

Figure 20. Disturbed yaw rate and thruster commands.

IV. Discussion

The pattern recognition control logic is robust to mass property and thruster orientation changes by using a lookup table to command thruster firings. The logic can be improved by considering potential changes to the spacecraft model. The analysis can be completed using a PID controller for the initial lookup table generation in addition to the phase plane. The simulations could also be run in Earth orbit in order to generate more accurate external disturbances. Additionally, the propulsion model could include fuel mass tracking to better measure the potential benefits of the control logic.

The new controller would require more sophisticated logic to handle complex thruster configurations and propulsion models. The current logic does not allow for a single thruster to be used for multiple axes. For example, if a pitch thruster generates roll torque, a roll thruster must fire to cancel that torque. A more complex jet mapping would be needed to account for torque coupling. Similarly, the logic assumed a near linear relationship between rate change and thruster firing time. If the propulsion system has a non-linear thrust profile, multiple lookup tables would be needed to compare different rates to the corresponding firing times. The logic also does not consider how thruster phasing can result in a loss of control, and how firing the opposing thruster combination can potentially recover control.

Further analysis can be done to develop a control system that calculates the rate combinations necessary to complete a maneuver. The uncertainty of the current command can be calculated by considering the potential future vehicle response.[8] Stochastic programming and receding horizon control would be useful for this type of maneuver commanding.[9] Furthermore, an adaptive controller can be designed that senses when multiple thruster firings are necessary to reach the commanded rate. If the rate error consistently does not converge with multiple firings, the controller can update the lookup table accordingly. An adaptive controller could also assess the available thrusters and modify the lookup table to account for thruster failures.

A more flexible database can be generated by instead firing each thruster for a set time and recording the rate change. The resulting lookup table would allow for multiple thruster combinations for a given rate command. The most efficient thruster combination can be selected based on thruster availability, firing time, and propellant usage. This type of database would require more checks in the control logic to select the best thruster combinations.

V. Conclusion

A spacecraft control system was designed that does not require knowledge of vehicle mass properties or thruster performance. The pattern recognition control logic uses a database of thruster firing commands and vehicle rates to efficiently control a spacecraft maneuver. The control logic requires inputs for the maneuver attitude limit, maneuver rate, rate error limit, minimum firing time, and hold time, many of which can be deduced when generating the lookup table. The control logic replaced a phase plane controller in a generic spacecraft model, with similar thruster firings and performance. A Monte Carlo analysis was performed to verify the robustness of the lookup table to mass property and thruster dispersions. The pattern recognition software TRAM was utilized to generate mappings of the dispersions and firing commands to failures. Thrust and moment of inertia had the largest impact on runs where the rate error did not converge. The logic was also modified to improve attitude hold capability in response to an external disturbance. The pattern recognition control logic can be expanded by further considering torque coupling and non-linear thrust profiles. An adaptive controller can also be designed that modifies the lookup table in the event of thrust and mass property changes. This controller design study verified the functionality and usefulness of a lookup table based control system.

A. Original Lookup Table

Case	Jet 1	Jet 2	Jet 3	Jet 4	Jet 9	Jet 10	Jet 11	Jet 12	Jet 13	Jet 14	Jet 15	Jet 16
1	0.28	0	0.28	0	0.28	0	0.28	0	0.24	0	0.24	0
2	0.36	0	0.36	0	0.36	0	0.36	0	0	0	0	0
3	0.28	0	0.28	0	0.28	0	0.28	0	0	0.28	0	0.28
4	0.36	0	0.36	0	0	0	0	0	0.32	0	0.32	0
5	0.52	0	0.52	0	0	0	0	0	0	0	0	0
6	0.36	0	0.36	0	0	0	0	0	0	0.32	0	0.32
7	0.28	0	0.28	0	0	0.28	0	0.28	0.28	0	0.28	0
8	0.36	0	0.36	0	0	0.36	0	0.36	0	0	0	0
9	0.28	0	0.28	0	0	0.28	0	0.28	0	0.24	0	0.24
10	0	0	0	0	0.32	0	0.32	0	0.32	0	0.32	0
11	0	0	0	0	0.48	0	0.48	0	0	0	0	0
12	0	0	0	0	0.36	0	0.36	0	0	0.32	0	0.32
13	0	0	0	0	0	0	0	0	0.4	0	0.4	0
14	0	0	0	0	0	0	0	0	0	0.4	0	0.4
15	0	0	0	0	0	0.36	0	0.36	0.32	0	0.32	0
16	0	0	0	0	0	0.48	0	0.48	0	0	0	0
17	0	0	0	0	0	0.36	0	0.36	0	0.32	0	0.32
18	0	0.28	0	0.28	0.28	0	0.28	0	0.24	0	0.24	0
19	0	0.36	0	0.36	0.32	0	0.32	0	0	0	0	0
20	0	0.28	0	0.28	0.28	0	0.28	0	0	0.28	0	0.28
21	0	0.36	0	0.36	0	0	0	0	0.32	0	0.32	0
22	0	0.52	0	0.52	0	0	0	0	0	0	0	0
23	0	0.36	0	0.36	0	0	0	0	0	0.32	0	0.32
24	0	0.28	0	0.28	0	0.28	0	0.28	0.28	0	0.28	0
25	0	0.36	0	0.36	0	0.36	0	0.36	0	0	0	0
26	0	0.28	0	0.28	0	0.28	0	0.28	0	0.24	0	0.24

Table 6. Thruster firing times in seconds.

B. Updated Lookup Table

Case	Jet 1	Jet 2	Jet 3	Jet 4	Jet 9	Jet 10	Jet 11	Jet 12	Jet 13	Jet 14	Jet 15	Jet 16
1	0.28	0	0.28	0	0.28	0	0.28	0	0.24	0	0.24	0
2	0.36	0	0.36	0	0.36	0	0.36	0	0	0	0	0
3	0.28	0	0.28	0	0.28	0	0.28	0	0	0.24	0	0.24
4	0.36	0	0.36	0	0	0	0	0	0.28	0	0.28	0
5	0.52	0	0.52	0	0	0	0	0	0	0	0	0
6	0.36	0	0.36	0	0	0	0	0	0	0.28	0	0.28
7	0.28	0	0.28	0	0	0.28	0	0.28	0.24	0	0.24	0
8	0.36	0	0.36	0	0	0.36	0	0.36	0	0	0	0
9	0.28	0	0.28	0	0	0.28	0	0.28	0	0.24	0	0.24
10	0	0	0	0	0.32	0	0.32	0	0.28	0	0.28	0
11	0	0	0	0	0.48	0	0.48	0	0	0	0	0
12	0	0	0	0	0.36	0	0.36	0	0	0.28	0	0.28
13	0	0	0	0	0	0	0	0	0.4	0	0.4	0
14	0	0	0	0	0	0	0	0	0	0.4	0	0.4
15	0	0	0	0	0	0.36	0	0.36	0.28	0	0.28	0
16	0	0	0	0	0	0.48	0	0.48	0	0	0	0
17	0	0	0	0	0	0.36	0	0.36	0	0.28	0	0.28
18	0	0.28	0	0.28	0.28	0	0.28	0	0.24	0	0.24	0
19	0	0.36	0	0.36	0.32	0	0.32	0	0	0	0	0
20	0	0.28	0	0.28	0.28	0	0.28	0	0	0.24	0	0.24
21	0	0.36	0	0.36	0	0	0	0	0.28	0	0.28	0
22	0	0.52	0	0.52	0	0	0	0	0	0	0	0
23	0	0.36	0	0.36	0	0	0	0	0	0.28	0	0.28
24	0	0.28	0	0.28	0	0.28	0	0.28	0.24	0	0.24	0
25	0	0.36	0	0.36	0	0.36	0	0.36	0	0	0	0
26	0	0.28	0	0.28	0	0.28	0	0.28	0	0.24	0	0.24

Table 7. Updated thruster firing times in seconds.

C. Maneuvers

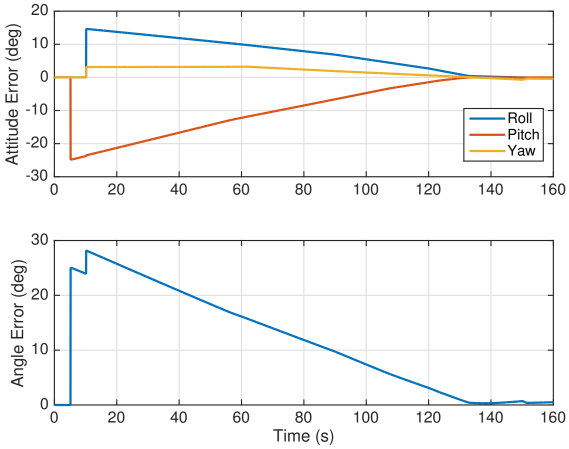

Figure 21. Large maneuver attitude error.

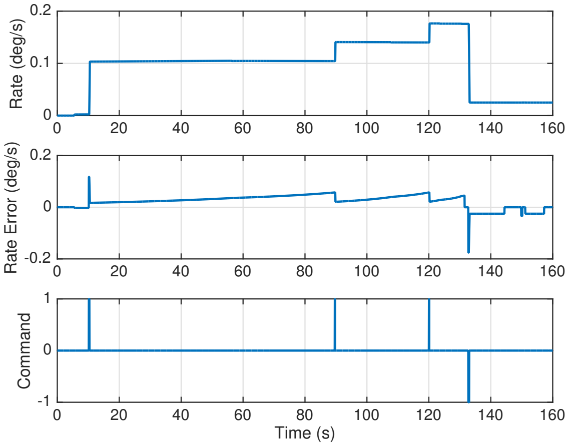

Figure 22. Large maneuver roll thruster commands.

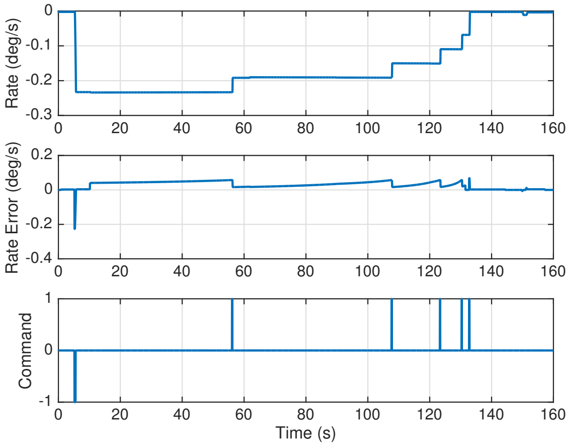

Figure 23. Large maneuver pitch thruster commands.

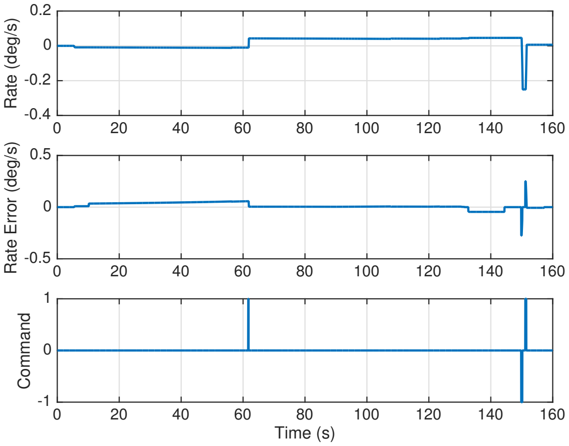

Figure 24. Large maneuver yaw thruster commands.

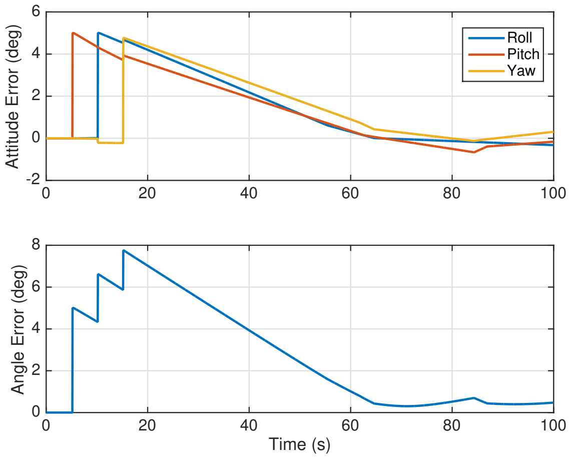

Figure 25. Slow maneuver attitude error.

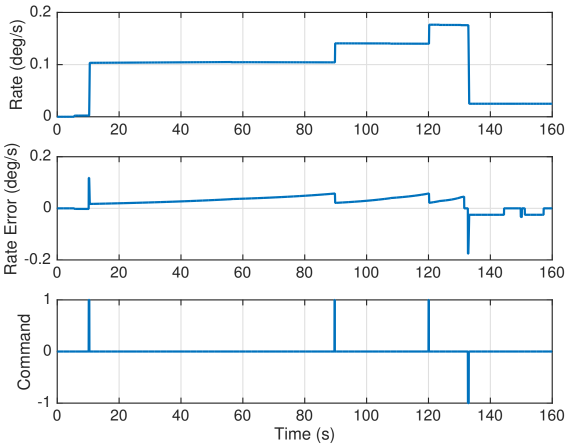

Figure 26. Slow maneuver roll thruster commands.

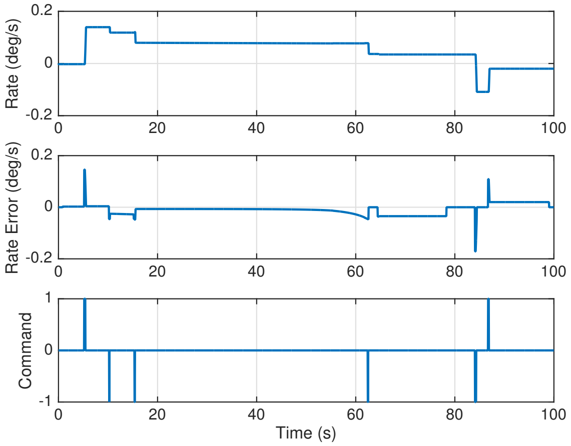

Figure 27. Slow maneuver pitch thruster commands.

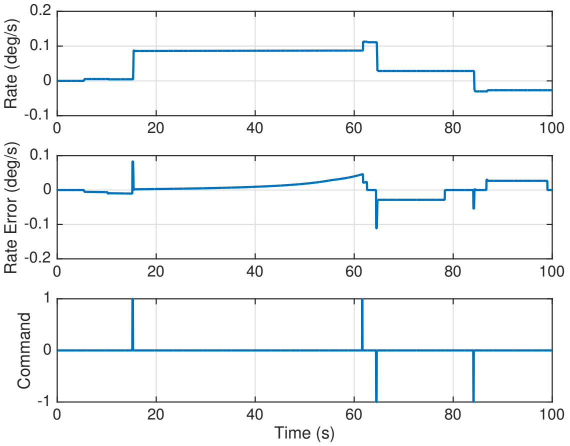

Figure 28. Slow maneuver yaw thruster commands.

Acknowledgments

Thanks to Professor John-Paul Clarke of Georgia Tech for providing advisement, resources, and expertise for this control system design. Thanks to Louis Nguyen and John Ruppert of NASA Johnson Space Center for providing direction and resources for this analysis.

References

[1] Gambone, E., "Morpheus Lander Roll Control System and Wind Modeling," *American Institute of Aeronautics and Astronautics*, 2015.

[2] Feron, E., "Attitude Control with Thrusters," *Massachusetts Institute of Technology*, 2004.

[3] Jang, J. W., Yang, L., and Fritz, M., "Design and Analysis of Morpheus Lander Flight Control System," *American Institute of Aeronautics and Astronautics*, 2014.

[4] Restrepo, C., McCall, K. E., and Hurtado, J. E., "Tool for Rapid Analysis of Monte Carlo Simulations," *American Institute of Aeronautics and Astronautics*, 2013.

[5] Restrepo, C. and Hurtado, J. E., "Pattern Recognition for a Flight Dynamics Monte Carlo Simulation," *American Institute of Aeronautics and Astronautics*, 2011.

[6] Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, 2007.

[7] Wittwer, J. W., "A Practical Guide to Monte Carlo Simulation," 2004.

[8] Seron, M., "Receding Horizon Control," *The University of Newcastle*, 2004.

[9] Sölveling, G. and Clarke, J. P., "Scheduling of airport runway operations using stochastic branch and bound methods," *Transportation Research Part C*, Vol. 45, 2014, pp. 119–137.