Autonomous Vehicle Simulation (AVS) Laboratory, University of Colorado

Basilisk Technical Memorandum

Document ID: Basilisk-thrFiringSchmitt

SCHMITT TRIGGER THRUSTER FIRING LOGIC MODULE

Prepared by	H. Schaub

Status: Public Release

Scope/Contents

A Schmitt trigger logic is implemented to map a desired thruster force value into a thruster on command time. The module reads in the attitude control thruster force values for both on- and off-pulsing scenarios, and then maps this into a time which specifies how long a thruster should be on. The thruster configuration data is read in through a separate input message in the reset method. The Schmitt trigger allows for an upper and lower bound where the thruster is either turned on or off.

Rev	Change Description	By	Date
1.0	First Release	H. Schaub	2019-03-29

Contents

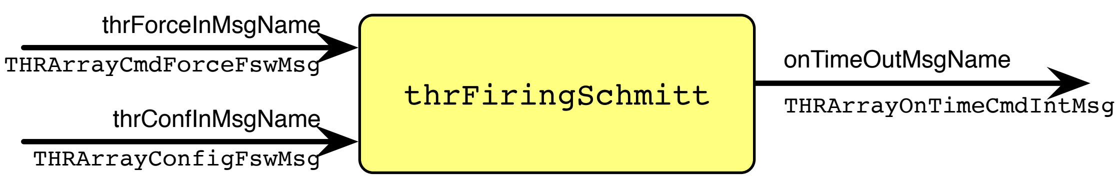

Fig. 1: Illustration of the module input and output messages.

1 Model Description

This module implements a Schmitt trigger thruster firing logic. More details can be found in Reference 1. Here if the minimum desired on-time t_{min} is specified. If the commanded on-time $t_i > t_{\text{min}}$ then the thruster is turned off for the duration of t_i. If $t_i < t_{\text{min}}$ then the Schmitt trigger logic is invoked. Let l be the current thruster duty cycle relative to this minimum thruster firing time.

$$l = \frac{t_i}{t_{\text{min}}}$$

If l is larger than a threshold l_{on} then the thruster control time is set to $t_i = t_{\text{min}}$. Once on, the thruster level must drop below a lower threshold l_{off} to turn off again. The benefit of this logic is that it provides a good balance between fuel efficiency and pointing accuracy.

1.1 Module Input and Output Messages

As illustrated in Figure 1, the module reads in two messages. One message contains the thruster configuration message from which the maximum thrust force value for each thruster is extracted and stored in the module. This message is only read in on `Reset()`.

The second message reads in an array of requested thruster force values with every `Update()` function call. These force values F_i can be positive if on-pulsing is requested, or negative if off-pulsing is required. On-pulsing is used to achieve an attitude control torque onto the spacecraft by turning on a thruster. Off-pulsing assumes a thruster is nominally on, such as with doing an extended orbit correction maneuver, and the attitude control is achieved by doing periodic off-pulsing.

The output of the module is a message containing an array of thruster on-time requests. If these on-times are larger than the control period, then the thruster remains on only for the control period upon which the on-criteria is reevaluated.

1.2 Reset() Functionality

- The control period is dynamically evaluated in the module by comparing the current time with the prior call time. In `reset()` the `prevCallTime` variable is reset to 0.
- The thruster configuration message is read in and the number of thrusters is stored in the module variable `numThrusters`. The maximum force per thruster is stored in `maxThrust`.
- The last thruster state variable `lastThrustState` is set to off (i.e. false)

1.3 Update() Functionality

The goal of the `update()` method is to read in the current attitude control thruster force message and map these into a thruster on-time output message using the Schmitt trigger logic.[1] The module sets a desired minimum thruster on time t_{min}. This is typically not set to the lower limit of the thruster resolution, but rather to a value that provides a good duty cycle and avoids excessive short on-off switching. The cost naturally is a reduced pointing capability. Let the duty cycle l be defined as the ratio between the commanded on time t_i and t_{min}. The thruster is turned on for a period $t_i = t_{\text{min}}$ if l is larger than l_{on}. The thruster then remains on until l drops below a lower threshold l_{off}. The benefit of this method is that it provides a good balance between fuel usage and pointing accuracy.[1]

If the `update()` method is called for the first time after reset, then there is no knowledge of the control period Δt. In this case the thruster on-time values are set either to 0 (on-pulsing case) or 2 seconds (off-pulsing case). After writing out this message the module is exited. This logic is essence turns off the thruster-based attitude control for one control period.

If this is a repeated call of the `update()` method then the control period Δt is evaluated by differencing the current time with the prior call time. Next a loop goes over each installed thruster to map the requested force F_i into an on-time t_i. The following logic is used.

- If off-pulsing is used then $F_i \leqslant 0$ and we set

  $$F_i + = F_{\text{max}}$$

  to a reduced thrust to achieve the negative F_i force.
- Next, if $F_i < 0$ then it set to be equal to zero. This can occur if an off-pulsing request is larger than the maximum thruster force magnitude F_{max}.
- The nominal thruster on-time is computed using

  $$t_i = \frac{F_i}{F_{\text{max}}} \Delta t$$
- If $\Delta t > t_i \geqslant t_{\text{min}}$ the thruster on time is set to t_i
- If $t_i > \Delta t$ then the thruster is saturated. In this case the on-time is set to $t_i = 1.1\Delta t$ such that the thruster remains on through the control period.

- The Schmitt trigger logic occurs if $t_i < t_{\text{min}}$. If $l > l_{\text{on}}$ then $t_i = t_{\text{min}}$. This command of $t_i = t_{\text{min}}$ remains on as long as $l > l_{\text{off}}$. If $l < l_{\text{off}}$ then $t_i = 0$ and the thruster is turned off again for the control period.

- The final step is to store the thruster on-time into and write out this output message

2 Module Functions

- **Read in thruster configuration message**: This is used to determine the number of installed thrusters and what the maximum force is for each.

- **Convert thruster force requested into an on-time request**: Knowing how strong the thruster is, the on-time is scaled such that the effectively applied force is equal to the requested force.

3 Module Assumptions and Limitations

The module assumes that the incoming forces F_i can be both positive or negative, depending if an on- or off-pulsing mode is being implemented. The particular mode is set through `baseThrustState`.

4 Test Description and Success Criteria

The unit test creates a desired thruster force input vector and then runs the simulation for 3 seconds. If the `resetCheck` flag is true then a `reset()` method is called and the simulation is repeated for another 2.5 seconds. If the `dvOn` flag is set than the off-pulsing mode is checked.

5 Test Parameters

The simulation sets up 8 thrusters. All permutations with the `resetCheck` and `dvOn` states are run. The output is checked to the tolerance shown in Table 2.

Table 2: Error tolerance for each test.

Output Value Tested	**Tolerated Error**
`OnTimeRequest`	1e-12

6 Test Results

All of the tests passed:

Table 3: Test results

`resetCheck`	`dvOn`	**Pass/Fail**
False	False	PASSED
False	True	PASSED
True	False	PASSED
True	True	PASSED

7 User Guide

The following variables are all required parameter to operate this module:

- `thrForceInMsgName`: string containing the thruster force input message

- `thrConfInMsgName`: string containing the thruster configuration input message

- `onTimeOutMsgName`: string containing the thruster on-time output message

- `thrMinFireTime`: Minimum thruster on-time t_{min} in seconds
- `baseThrustState`: Flag indicating either an on-pulsing (0) or off-pulsing (1) configuration
- `level_on`: Upper duty cycle percentage threshold relative to t_{min} to turn on thrusters
- `level_off`: Lower duty cycle percentage threshold relative to t_{min} to turn off thrusters

REFERENCES

[1] John Alcorn, Hanspeter Schaub, and Scott Piggott. Steady-state attitude and control effort sensitivity analysis of discretized thruster implementations. *AIAA Journal of Spacecraft and Rockets*, 54(5):1161–1169, 2017.