

ScramAttitude v1.1

© 2017 Andrew “ADSWNJ” Stokes, licensed under GNU GPL 3.0

Credit to Szymon “enjo” Ender for MFDButton framework, to Doug “dbeachy1” Beachy (Altea Aerospace) for the XR-series of spaceships that make this a joy to fly, and to Martin “martins” Schweiger for the whole Orbiter environment, that has spawned a great community and a passionate following around the world.

Revision History

V1.0 Nov 11th 2017

- Original Release for Orbiter 2016

V1.1 Nov 26th 2017

- Moved the tuning options into a ScramAttitude_VesselParams.cfg file
- Renamed Dump.csv to ScramAttitude_Dump.csv for consistency
- Tuned the autopilot response for DG-S, XR1, XR2, XR5
- New scenario files with no external dependencies apart from the respective vessels.

Installation

This MFD is designed to run on Orbiter 2010 or Orbiter 2017, on Windows 7, Windows 8 or Windows 10. Note that there are two separate compiled packages, one for each version of Orbiter. Unzip the right zip file for your version of Orbiter over the top of your Orbiter installation, so that the files go in the right directories and locations (e.g. ScramAttitude.dll goes in your Orbiter’s Modules\Plugin directory). Once installed, enable the ScramAttitude module in the Modules tab in your Orbiter launchpad.

Introduction

The purpose of this MFD is to maintain a good ascent profile during a SCRAM engine burn phase. The MFD does this by maneuvering elevator trim to trend the spacecraft towards a target Dynamic Pressure. This MFD is intended to be used on the XR-series of vessels, though can be further tuned for other vessels as the tuning parameters are all exposed in the scenario files.

The MFD constantly varies Elevator Trim to drive Vertical Acceleration, to indirectly drive the change in Dynamic Pressure to achieve the desired Target Dynamic Pressure. The internal algorithm uses ‘fuzzy logic’ to achieve this goal regardless of vessel weight, altitude, speed, and performance.

The MFD can be run semi-automated (Fixed VAcc mode) or fully automated (Auto VAcc mode). In Fixed VAcc mode, you manually select the desired Vertical Acceleration, to manually achieve your desired Dynamic Pressure, and the MFD adjusts the Elevator Trim to achieve your VAcc. In Auto VAcc mode, the fuzzy logic algorithm controls the VAcc goal to drive the desired Dynamic pressure target automatically.

Description of MFD Display and Controls



1. Display elements:

- **DP:** Dynamic Pressure (kPa). This represents the effective air pressure into the SCRAM engine, from the combination of static air pressure at your current altitude, plus the increase in pressure due to your airspeed. For a good XR-series ascent, this should be set to a target of 10.0 for the bulk of the ascent, and if you prefer, it can be reduced towards 4.0 for the late stages.
- **VAcc:** Vertical Acceleration (m/s^2). This is loosely correlated to the change in DP, so is a primary readout on this MFD. The target appears if the AP is active.
- **Elev:** Current and target elevator position, where -1.00 is max down, and +1.00 is max up. If the elevator hits the min or max limit, the target will go yellow, and there will be a warning in yellow at the bottom of the MFD, indicating loss of elevator trim control. (Usually this is a transient condition and will rectify automatically.)

- **Mach:** Airspeed (in Mach). For a good ascent, this value should continuously increase throughout the period of the SCRAM engine burn. If the MFD detects a slow-down, a read warning will show at the bottom, to alert you to correct the condition.
- **VSpd:** Vertical Speed (m/s). This value is not a primary control input, but for manual flight is useful to watch, to ensure it does not get too high or low (e.g. keep to -50 m/s to +150 m/s for manual SCRAM ascents).
- **DPSpd:** Rate of Change of Dynamic Pressure (kPa/s). Try to keep in the range -0.5 to +0.5 for manual SCRAM ascents.
- **DPErA:** Average Absolute Error in Dynamic Pressure relative to Target. Useful for calibration and diagnostic purposes, and can be reset by pressing REA, but not a primary control parameter.
- **AP:** Autopilot status. Can be OFF, ON AUTO VACC, or ON FIXED VACC. When off, there is no control of Trim or Dynamic Pressure from this MFD. When in AUTO VACC mode, the autopilot dynamically adjusts the VACC to achieve the DP target. When in FIXED VAC mode, the autopilot adjusts trim to maintain a manually selected VACC target (selected by A--, A1D, A0, A1U and A++ buttons).
- **Log:** Diagnostic dump log status. Can be OFF, ON, or FAILED TO OPEN. The log file is ScramAttitude_Dump.csv, in the Config/MFD/ScramAttitude folder. If the log cannot be opened, some reasons may include: bad permissions or missing folder, file open, or file not cleanly closed from a previous simulation run.
- **DE\E and next 3 lines:** this is Diagnostic data only (selected by the DIA button). See the Internals section below for further details.
- **Warning area:** the bottom of the MFD shows a variety of warnings. For example, in this screen, you have a major warning for slow air speed (Mach: CHECK SLOW SPEED), a warning for elevator at max or min (e.g. Elevator Control: AT MAX), and a warning preventing warp acceleration under autopilot (Warp: MAX 1.0 on AUTO VACC AP).

2. Controls:

- **AP = Arm or Disarm the Autopilot**
 - Simple On/Off toggle
- **DP+/DP- = Dynamic Pressure Target Increase / Decrease**
 - Manually adjusts the Dynamic Pressure target from the default 10.0
- **LOG = Start or Stop the Diagnostic Log**
 - Creates a dump file (Dump.csv) in the Config/MFD/ScramAttitude folder. See the Internals section for more information.
- **REA = Reset DPErA value (Dynamic Pressure Absolute Error Average)**
 - Diagnostic use only. Resets the absolute error accumulator for the difference between current and target DP. Used for tuning the AP response.
- **DIA = Show / Hide the Diagnostic display**
 - Shows or hides 4 lines starting DE\E. See the Internals section for more information.
- **A--, A1D, A0, A1U, A++ = Toggle Fixed VAcc AP mode and set VAcc**

- A0 sets the VAcc Fixed target to 0. A1D and A1U sets it to 1- or +1. A--and A++adjusts down/up by one.
- **APA = Enable Autopilot Auto VAcc mode**
 - If you are in AP FIXED VACC mode, pressing APA selects AP AUTO VACC mode.

User Guide for the ScramAttitude MFD

If you are using the supplied scenarios, then you just need to launch the scenario and watch the ScramAttitude autopilot control the vessel for you.

If you are doing a full mission from takeoff, then here's a suggested approach for you. Prior to take-off determine the appropriate time and heading to reach your target (e.g. the ISS). LaunchMFD is a great utility for this.

After take-off, roll smoothly to your desired heading, and climb quickly to reach 20km altitude. Level off and allow your speed it increase to the lower recommended SCRAM speed (e.g. Mach 3.5 for XR1 or XR2, Mach 4.0 for the DG-S, and Mach 4.5 for the XR5). Whilst waiting for the right speed, bring up the ScramAttitude MFD, and confirm that the AP is not yet activated.

At the right speed, open the SCRAM doors and throttle up the SCRAM engine, whilst throttling the Main engine back to idle. Press AP to enable the ScramAttitude AP in AUTO VACC mode, noting the default dynamic pressure target (e.g. 10.0 for XR-series vessels). Make sure that there are no other autopilots selected (e.g. switch off the Attitude AP or Level Horizon AP if you have been using that to maintain your vertical profile).

Monitor the ScramAttitude AP, confirming the airspeed rises smoothly. Watch and confirm that the DP trends down to the target and then maintains the DP target (e.g. within +/- 0.5 kPa).

When the SCRAM fuel is exhausted, or the SCRAM temperature limit is reached, disable the ScramAttitude AP, close the SCRAM doors and throttle up the Main engine to complete the ascent.

If you get a bad flight – e.g. excessive oscillations or badly performing AP, then see if you can recreate the scenario, but with the LOG on. The resulting log file in your Config\MFD\ScramAttitude folder will allow you to study the anomaly further. The fix will likely require you to update the Vessel Params files (in which case, continue reading the internals section below).

Known Issues

1. The AUTO VACC AP is unstable at 10.0 Warp, so this has been disabled in the MFD. If you attempt to warp, you will get a yellow warning in the MFD screen.
2. The AP has been designed for DG-S, XR1, XR2, and XR5. Usage with other SCRAM vessels may or may not work, and may need further parameter tuning to be stable. See the Internals section.

Internals

This section is of interest only to developers and technical geeks who are interested in the internal implementation of this MFD, and who may want to get their hands dirty in the parameter tuning. For end-users, most of this will not make any sense at all, and that's ok!

Please see the source at <https://github.com/ADSWNJ/ScramAttitude/> You will always find the latest main line and future feature branches up there.

1. Fuzzy Logic Engine

A typical control loop is designed for a known system, and is optimized to respond with an appropriate control input to minimize error in the desired variable. For example, in our case, the error is the difference in DP to the DP target, and the control input is the change in the elevator trim. These standard control loops are usually called PID controllers (Proportional, Integral, Differential errors – i.e. the delta value, the sum of the errors, and the rate of the errors). Simplified versions of these controllers typically drop out one of the values, to make a PD controller, or a PI controller.

When one does not know the characteristics of the system, or where the system can vary wildly according to speed, altitude, weight, responsiveness, etc., a fixed control is inappropriate. For these systems, a fuzzy logic control engine is typically better, as it can implement variable control loop parameters according to the real-time dynamic response.

For this MFD, I was interested in an old 1990's paper by Robert P Copeland and Kuldip S Rattan from Wright State University, entitled A Fuzzy Supervisor for PD Control of Unknown Systems (see: <http://cecs.wright.edu/~krattan/courses/419/louisville.pdf> and a copy in my GitHub repo). This paper examined and improved a fuzzy control matrix proposed by Macvicar and Whelan back in 1976. They proposed a 9x9 control matrix, representing the sign and size of the error and the rate of the error, and a method to blend this data into a continuous control input.

Whilst the technical implementation can be seen directly in the source code on GitHub, the diagnostic mode (DIA button) can show you a window into how the fuzzy engine is evaluating the conditions.

Once you press DIA, you will see additional data in the MFD display, similar to this:

LOG	DE\ E	PM	PL			A1U
	NX	+1.5	+0.5	40%	16%	
	NL	+0.5	+0.0	32%	13%	
REA	Out:	+0.83				A++

The fuzzy logic engine is continually monitoring the primary error (E), which is the DP – Target DP, and also monitoring the rate of change of error (DE) over time. The logic fuzzy engine breaks E and DE into 9 control states: Negative Extra-Large (NX), Negative Large (NL), Negative Medium (NM), Negative Small

(NS), Zero (Z0), Positive Small (PS), Positive Medium (PM), Positive Large (PL), and Positive Extra-Large (PX), according to these default values in the Segmentation Table:

	NX-NL	NL-NM	NM-NS	NS-Z0	Z0-PS	PS-PM	PM-PL	PL-PX
E	+15.00 to +5.00	+5.00 to +2.00	+2.00 to +0.50	+0.50 to +0.00	+0.00 to -0.50	-0.50 to -2.00	-2.00 to -3.00	-3.00 to -5.00
DE	+0.40 to +0.10	+0.10 to +0.05	+0.05 to +0.02	+0.02 to +0.00	+0.00 to -0.02	-0.02 to -0.05	-0.05 to -0.10	-0.10 to -0.40

For each reading of E and DE, the fuzzy logic engine determines the two closest control states for each of E and DE, to give 4 values in a 2x2 box from the following Control Table:

	E:NX	E:NL	E:NM	E:NS	E:Z0	E:PS	E:PM	E:PL	E:PX
DE:NX	NX	NX	NX	NL	NL	NL	NM	NS	PS
DE:NL	NX	NL	NL	NL	NL	NM	NS	Z0	PL
DE:NM	NX	NL	NL	NL	NM	NS	Z0	PS	PX
DE:NS	NX	NL	NL	NM	NS	Z0	PS	PM	PX
DE:Z0	NX	NL	NM	NS	Z0	PS	PM	PL	PX
DE:PS	NX	NM	NS	Z0	PS	PM	PL	PL	PX
DE:PM	NX	NS	Z0	PM	PM	PL	PL	PL	PX
DE:PL	NL	Z0	PS	PL	PL	PL	PL	PL	PX
DE:PX	NS	PS	PM	PL	PL	PL	PX	PX	PX

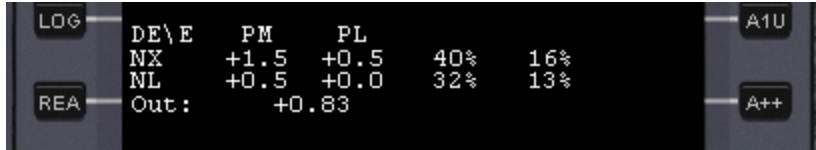
The desired VAcc output value is driven from the Desired VAcc Table:

NX	NL	NM	NS	Z0	PS	PM	PL	PX
+5.00	+3.00	+1.50	+0.50	0.00	-0.50	-1.50	-3.00	-5.00

The error in the VAcc drives the rate of adjustment of the Elevator Trim according to the Trim Control Table:

Abs VAcc Err	0.05	0.10	0.20	0.30	0.50
Trim%/sec	0.20%	0.20%	0.20%	0.30%	0.50%

Returning to this output data, we can now put this all together:



The DE\E indicates the DE is the column data and the E is the row data (i.e. this is not an equation!). For this particular instant, the E reading was roughly -2.28, so it's between PM and PL on the Segmentation Table. The DE reading was roughly +0.268, so between NX and NL on the Segmentation Table. See the red box on the Control Table, showing the currently selected 2x2 box, with corresponding output values from the Desired VAcc Table. The percentages show the weightings of the 4 values, according to how close the actual values are to the 4 corners of the table. In this case, E is 71.6% towards the E:PM value, and DE is 55.97% towards the DE:NX value, so the top left weighting is 71.60% x 55.97% ≈ 40%. And so on for the other three values. The output is the weighted sum of the 4 values, and represents the desired VAcc from the fuzzy logic engine, finally driving the elevator trim adjustment. Simple, hey?

2. Vessel Params Config File

Let's now look at the Vessel Params config file, and how you might adjust it if needed:

```
BEGIN_VESSEL_CLASS "DG-S"
#           --NX--   ---NL---   ---NM---   ---NS---   ----Z0----   ...
SEG_E_CTL   25.00   15.00 5.00    2.00 2.00    0.50 0.50    0.00 0.00    ...
SEG_DE_CTL   0.40    0.10 0.10    0.05 0.05    0.02 0.02    0.00 0.00    ...
DES_VACC_CTL 10.00    3.00      1.50      0.50      0.00      ...

#           --Rate1--   --Rate2--   --Rate3--   --Rate4--   --Rate5--
TRIM_CTL     0.05 0.40    0.10 0.80    0.20 1.20    0.30 2.40    0.50 4.00

#
HIST_COUNT   13
MIN_SIMD     0.1
DP_TGT       75
END_VESSEL_CLASS
```

Here is a sample from the ScramAttitude_VesselParams.cfg file. Note – I have trimmed the ends of the SEG_E_CTL, SEG_DE_CTL, and DES_VACC_CTL lines to fit this page. The parser for this file will ignore any comment (starts with a # sign and the rest of the line is treated as comment), as well as ignoring any spare whitespace on each line (e.g. additional spaces or tabs), so you can line up the fields as you prefer.

There are 9 commands in a vessel definition, as follows:

- **BEGIN_VESSEL_CLASS** "class". The class name is defined by the vessel author. If you are trying a new vessel, then look in the Orbiter.log and ScramAttitude will display the missing vessel class name, before reverting to default settings. If the class definition is not for our current vessel, then the following lines until END_VESSEL_CLASS are ignored.

- **SEG_E_CTL** is the 16 values from the E line of the Segmentation Table. E.g. 15.00 to 5.00 is the NX to NL transition, 15.00 to 5.00 is a constant NL rate, 5.00 to 2.00 is the NL to NM transition, and so on. The E refers to Error – i.e. the error in the dynamic pressure relative to target.
- **SEG_DE_CTL** is the 16 values from the DE line of the Segmentation Table. Same format as the SEG_E_CTL line, but applying to the delta-errors (i.e. rate of change of dynamic pressure).
- **DES_VACC_CTL** is the 9 values from the Desired VAcc Table. Adjust the acceleration rates to set acceleration response to the needs of your spacecraft (e.g. to avoid wing stress or unwanted aggressive oscillations.)
- **TRIM_CTL** is the 10 values from the Trim Control Table, reading down each column, then left to right. E.g. Absolute Vertical Acceleration Err up to 0.05 = 0.40%, then up to 0.10 = 0.90%, and so on.
- **HIST_COUNT** sets the amount of historical readings used to calculate the rates and accelerations. Set this to an odd number, between 3 and 127. What does this do? It looks at the altitude and DP from HIST_COUNT readings ago, then one in between, and then our current reading, to determine 2 rates (i.e. mid minus old and current minus mid), and then one acceleration (i.e. new rate – old rate). Setting this value odd means that we have an even number of readings in each rate calculation. Setting this number too low will make the rates and acceleration more volatile. Setting too high will damp down the rates, but also slow the responsiveness of the autopilot. Typically set this to an odd number between 9 and 29.
- **MIN_SIMD** is the smallest simulation delta time between runs of the autopilot. Typically set this between 0.05 and 1.00 seconds. This prevents the HIST_COUNTN being used up with a very small set of timed readings – i.e. there is a relationship between MIN_SIMD and HIST_COUNT, such that the min timing range for the rate calculations is HIST_COUNTN * MIN_SIMD. If you want a very relaxed AP, then you may set the MIN_SIMD to say 1.00, and a HIST_COUNT of say 3, and watch it leisurely update the readings.
- **DP_TGT** is the desired default dynamic pressure target for this vessel class.
- **END_VESSEL_CLASS** ends the vessel definition. If this was our vessel class, then the parser confirms that all 7 settings have been correctly parsed, before transferring the defaults through to the autopilot.

If at any time, the parser detects an error in the file, or the file is missing or corrupted, it will throw an error to the Orbiter.log and continue with default settings.

3. Scenario File

Let's now look at how this data ends up in the scenario file, and how you might adjust it if needed:

```
BEGIN_MFD Right
TYPE User
MODE ScramAttitude
DP_TGT 10.00
VACC_TGT 5.00
```



```
AP_MODE 1
LOG_MODE 1
DIAG_MODE 0
END_MFD
```

Here is a sample from a .SCN file from a saved scenario with ScramAttitude running. The BEGIN_MFD, TYPE, MODE, and END_MFD lines are common to every MFD scenario. The other lines are as follows:

- **DP_TGT** = Dynamic Pressure target
- **VACC_TGT** = VAcc Target (note: this is dynamically changing when AP is in AUTO, or fixed for AP FIXED VACC mode)
- **AP_MODE** = 0 for OFF, 1 for ON AUTO VACC, and 2 for ON FIXED VACC
- **LOG_MODE** = 0 for OFF, 1 for ON. (See next section on log file format)
- **DIAG_MODE** = 0 for OFF, 1 for ON

4. Log File format

if you enable the log, the MFD will create a dump file called ScramAttitude_Dump.csv in your Config\MFD\ScramAttitude folder. Make sure you toggle the LOG back off again before closing the simulation, so you have a clean CSV file.

The columns are as follows:

- **MJD** = Modified Julian Date of the reading
- **T** = Time since start of simulation
- **DT** = Delta-Time since last reading
- **AP** = State of the AutoPilot
- **Mach, TAS, Alt, VSpd, VAcc** = standard flight readings
- **DP_tgt, DP_act, DP_spd, DP_acc** = Dynamic pressure target, actual, rate of change, acceleration
- **Trim_act, Trim_tgt** = elevator trim actual to target
- **VAcc_tgt** = Vertical Acceleration target from the AP. Usually same as MW_Out (we can occasionally clip this target to prevent excessive jerk on the airframe)
- **MW_Seg(E), MW_Seg(DE)** = raw segmentation values from the fuzzy logic algo. -4 = NL to +4 = PL on a continuum scale according to the bands for each value.
- **MW_tl, MW_bl, MW_tr, MW_br** = the 4 values on the diag output (top left, bottom left, top right, bottom right respectively)
- **MW_tl%, MW_bl%, MW_tr%, MW_br%** = the 4 percentages on the diag output (top left, bottom left, top right, bottom right respectively)
- **T** = Time since start of simulation (duplicated for ease of graphing)
- **MW_Out** = the Out value on the Diag output, as the fuzzy logic Desired VAcc target.
- **DP_d** = DP delta value = DP_act - DP_tgt
- **VAcc** = Vertical Acceleration (duplicated for ease of graphing)

- **SumAbsErr** = accumulator of the absolute DP_d, incremented once per second. Used to tune the AP response, by comparing error values for identical scenarios with different tuning parameters and seeing how much error accumulates over time. This is averaged and displayed as DPErA on the MFD output, and will be reset with REA button.

On a typical calibration run, I would watch the autopilot's performance, making notes for specific simulation times (e.g. SimT at 128 secs ... AP too slow controlling the DP rate, or too big a DP overshoot). If I am tuning each oscillation of the DP_d response, I will often hit REA at each zero-point, to look at the errors for each 'bounce' independently. After the simulation is complete, I would use MS Excel to open this CSV, View Freeze Panes, Freeze Top Row to lock the column headings, and then scroll through the data to look at the response rates for periods of interest. A typical graph response would be to grab the AA and AC columns (T vs DP_d), Insert Chart, Scatter with smooth Lines ... to see the DP response against time, or AA:AD for T vs DP vs VAcc vs MW_Out and put the DP response on a secondary axis, to see the control responses overlaid on the DP delta plot. If you get into this level of analysis, then PM me and let's talk!

Thanks for reading! Regards, Andrew "ADSWNJ" Stokes