Autonomous Flight Termination Systems

USING THE

CORE AUTONOMOUS SAFETY SOFTWARE (CASS)

AND THE

CASS HAZARD EVALUATION READINESS REVIEW ANALYSIS (CHERRY) TOOL

AFTS Overview

An Autonomous Flight Termination System (AFTS) is an on-board unit that monitors the flight vehicle by comparing sensor inputs to pre-defined Range Safety criteria. If the flight vehicle violates safety criteria, the AFTS issues a flight termination command.

AFTS Advantages

- Faster response time allows larger safe fly zones.
- Reduced launch costs due to less dependence on Range infrastructure and instrumentation.
- Increased launch availability.
- 24/7 launch with no Range support.
- Launch in remote locations – off Range.

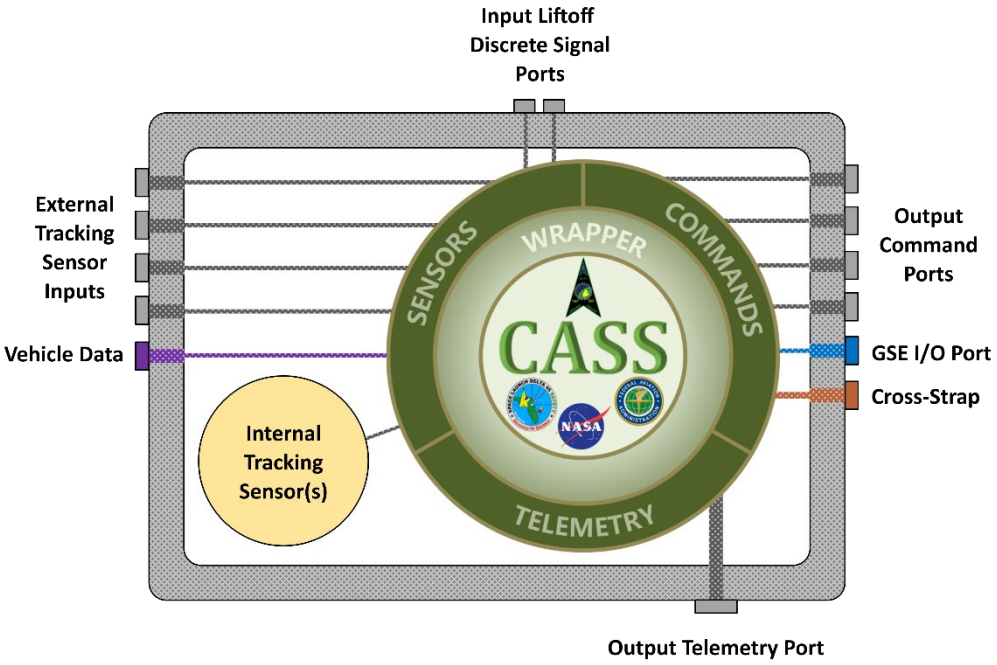
Topics

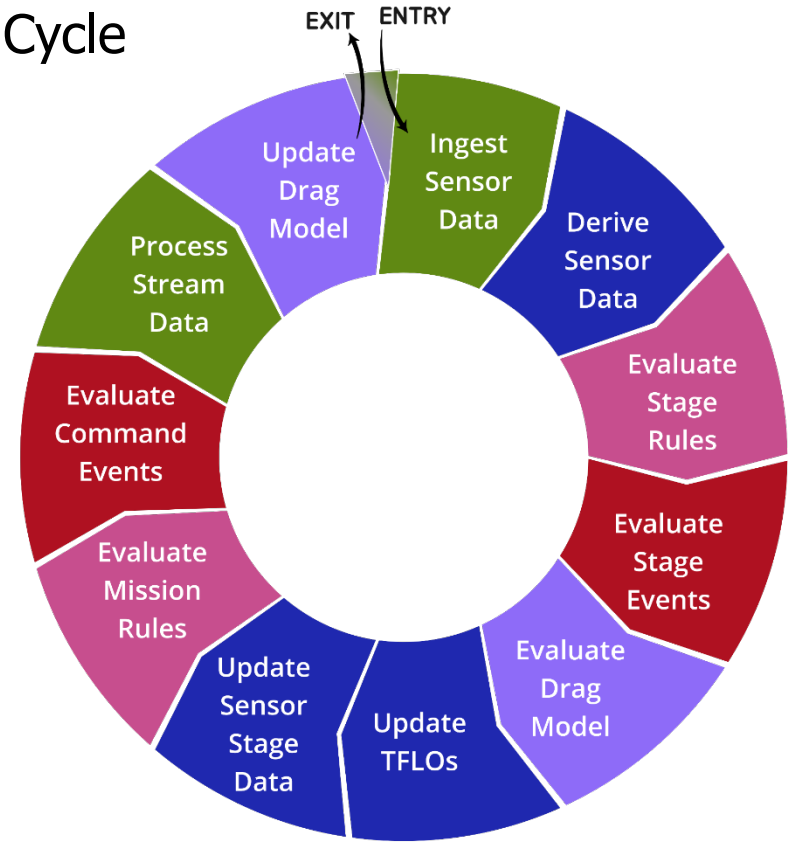
- Core Autonomous Safety Software (CASS)
- Mission Rules Development
- Testing and Simulation (CHERRY Tool)
- AFTS Unit
- Getting the Software

CASS

- CASS is composed of:
  - Flight Software
  - Utilities Software
  - Documentation
- CASS is GFE
  - Core software for an AFTS.
  - Used like any third-party software product approved for use in a Safety Critical system.
- CASS Requires
  - Safety criteria; Mission Data Load (MDL).
  - Sensor data; 2 or more adequate and independent sources of vehicle performance data at a regular, periodic rate.
  - Wrapper software; bridges gap between hardware sensors and CASS, and between CASS and flight termination system.

CASS Update Cycle





	Data is input from, or output to, the wrapper.
	Derived data computations, indepdendent of Mission Rules.
	Rule evaluations; Mission Rule dependent.
	Event/Command evaluations; Mission Rule dependent.
	Drag model processing; Mission Rule dependent.

Mission Rules Development

- Mission Rules defined in XML text file
  - Format defined by Range Safety Operations Markup Language (rsoML) schema definition.
  - Mission Rules file is composed of 11 Major sections.

Section	Description
Mission	Contains descriptive data for mission.
UserDefines	User defined variables and named constants.
Settings	Global parameter settings and constants.
Commands	Commands to Activate or deactivate the FTS.
Sensors	Sensor definitions and parameters.
AtmosphericRegions	Define region to include atmospheric computations.
FlightEvents	Important flight events like ignition, liftoff, and staging.
ReferenceFrames	User defined moving reference frames.
Boundaries	User defined boundary definitions.
Tables	User defined lookup tables.
MissionRules	Rules for Safing the flight and Terminating the flight.

Sensors Sample

```
<Sensors>
   <Sensor id="GPS_A">
      <Filter>
         <Limit_Filter>
            <Minimum_Time>              @GPST 2141 216000.0                  </Minimum_Time>
            <Minimum_Position>  DistanceFromCenterMassEarthToDeadSea         </Minimum_Position>
            <Maximum_Position>  DistanceFromCenterMassEarthToLowEarthOrbit   </Maximum_Position>
            <Maximum_Velocity>          VelocityToGoOrbital                  </Maximum_Velocity>
         </Limit_Filter>
      </Filter>
      <liftOffThreshVel> Velocity_Threshold_GPS </liftOffThreshVel>  <!-- In meters/second. -->
      <velDotParam> Acceleration_Filter_Coefficient </velDotParam>  <!-- Unitless. -->
      <QualifyLogic default="false">
         <cond> GPS_A.isValidNavData is true                </cond> <and/>
         <cond> GPS_A.isValidGPSData is true                </cond> <and/>
         <cond> GPS_A.svCount &gt; Minimum_Satellite_Count  </cond> <and/>
         <cond> GPS_A.PDOP &lt; Maximum_PDOP                </cond> <and/>
         <cond> GPS_A.Time &lt; System_Time                 </cond>
      </QualifyLogic>
   </Sensor>
      •
      •
      •
</Sensors>
```

FlightEvents Rules Sample: Stage Ignition

```
<FlightEvents>
   <Rules>
      <GenericRule id="Stage_Ignition">
         <Compute>
            <Assign id="Ignition_Acceleration_Threshold"> Ignition_Acceleration_Threshold_Table(Stage_Number) </Assign>
         </Compute>
         <Subset> GPS_A GPS_B HYBRID_A HYBRID_B INS_A INS_B    </Subset>
         <Output>
            <InvalidWhen default="true">
               <cond> isGoodSensorData is false                    </cond> <or/>
               <cond> haveLiftOff is false                         </cond>
            </InvalidWhen>
            <Result default="false">
               <cond> accelTotal &gt; Ignition_Acceleration_Threshold </cond>
            </Result>
         </Output>
      </GenericRule>
        •
        •
        •
   </Rules>
</FlightEvents>
```

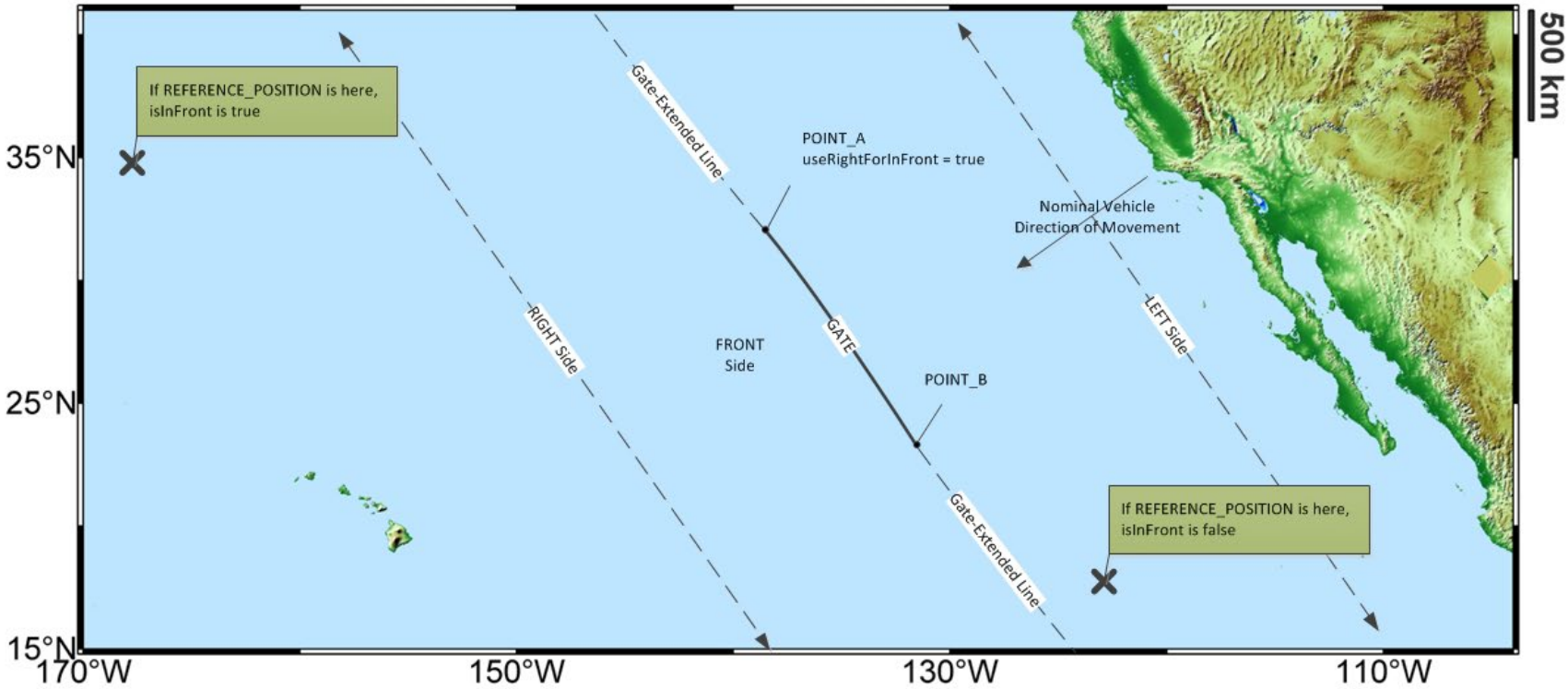
FlightEvents Rules Sample: Liftoff

```
<FlightEvents>
   <Rules>
      <GenericRule id="Liftoff_Hardware_Only">
         <Output>
            <InvalidWhen default="true">
               <cond> haveLiftOff is true                  </cond> <or/>
               <cond> GPS_A.isGoodSensorData is true       </cond> <or/>
               <cond> GPS_B.isGoodSensorData is true       </cond> <or/>
               <cond> HYBRID_A.isGoodSensorData is true    </cond> <or/>
               <cond> HYBRID_B.isGoodSensorData is true    </cond> <or/>
               <cond> INS_A.isGoodSensorData is true       </cond> <or/>
               <cond> INS_B.isGoodSensorData is true       </cond>
            </InvalidWhen>
            <Result default="false">
               <cond> haveLiftOffA is true               </cond> <and/>
               <cond> haveLiftOffB is true               </cond>
            </Result>
         </Output>
      </GenericRule>
        •
        •
        •
   </Rules>
</FlightEvents>
```

FlightEvents Stages Sample: Stage

```
<FlightEvents>
   <Stages>
      <Stage id="Stage_1">
         <!--
            Detect Stage 1 ignition (liftoff) if we have two hardware indications of liftoff, or
            we have one hardware indication and two or more sensors detected liftoff using elevated acceleration, or
            we have no hardware indication and two or more sensors detect liftoff using definitive acceleration.
         -->
         <IgnitionLogic default="false">
            <cond> Liftoff_Hardware_Only.Result is true                                    </cond> <or/>
            <vote zero="false" one="false" two="and" tie="true"> Liftoff_Sensors_And_Hardware </vote> <or/>
            <vote zero="false" one="false" two="and" tie="true"> Liftoff_Sensors_Only         </vote>
         </IgnitionLogic>
         <!--
            Detect Stage 1 burnout after the minimum burn time has elapsed and two or more sensors
            detect an acceleration drop below the level acceptable for Stage 1 burnout.
         -->
         <BurnoutLogic default="false">
            <cond> Stage_1.TimeSinceIgnition &gt; Minimum_Burn_Time_Stage_1                 </cond> <and/>
            <vote zero="false" one="false" two="and" tie="true"> Stage_Burnout              </vote>
         </BurnoutLogic>
      </Stage>
        •
        •
        •
   </Stages>
</FlightEvents>
```

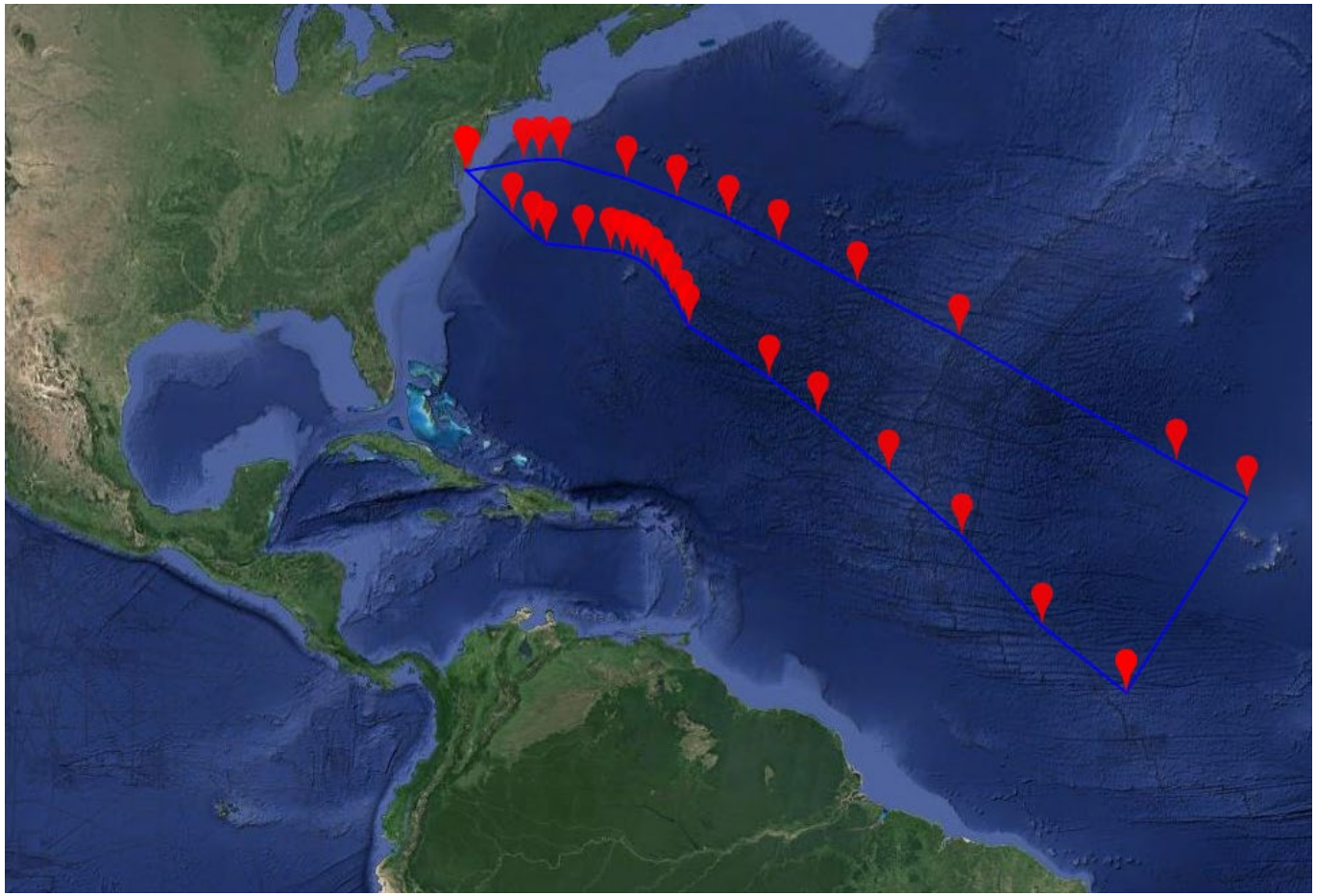
Gate Rules


If REFERENCE_POSITION is here, isInFront is true
Gate-Extended Line
POINT_A
useRightForInFront = true
Nominal Vehicle Direction of Movement
RIGHT Side
FRONT Side
GATE
POINT_B
LEFT Side
Gate-Extended Line
If REFERENCE_POSITION is here, isInFront is false
500 km
35°N
25°N
15°N
170°W
150°W
130°W
110°W

Gate Example: Headon Exit Gate

```
<MissionRules>
   <GateRule id="Headon_Exit_Gate">
      <Subset> GPS_A GPS_B HYBRID_A HYBRID_B INS_A INS_B </Subset>
      <Output>
         <InvalidWhen default="true">
            <cond> haveLiftOff is false                </cond> <or/>
            <cond> isGoodSensorData is false           </cond> <or/>
            <cond> impactComputed is false             </cond> <or/>
            <cond> Perigee &lt; Maximum_Perigee_Headon </cond>
         </InvalidWhen>
         <Result default="false">
            <cond> Headon_Exit_Gate.isCrossed is true </cond>
         </Result>
      </Output>
      <GateCoordinates>
         <PointA> <Lat> 10.547600 </Lat> <Lon> -37.226000 </Lon> </PointA>
         <PointB> <Lat> 21.731800 </Lat> <Lon> -30.005600 </Lon> </PointB>
      </GateCoordinates>
      <TripMode> CrossLeftToRight </TripMode>
      <CrossPersist> 1 </CrossPersist>
      <RefCoordinates> Impact </RefCoordinates>
   </GateRule>
         •
         •
         •
</MissionRules>
```

Map Boundary Rules: Destruct Lines



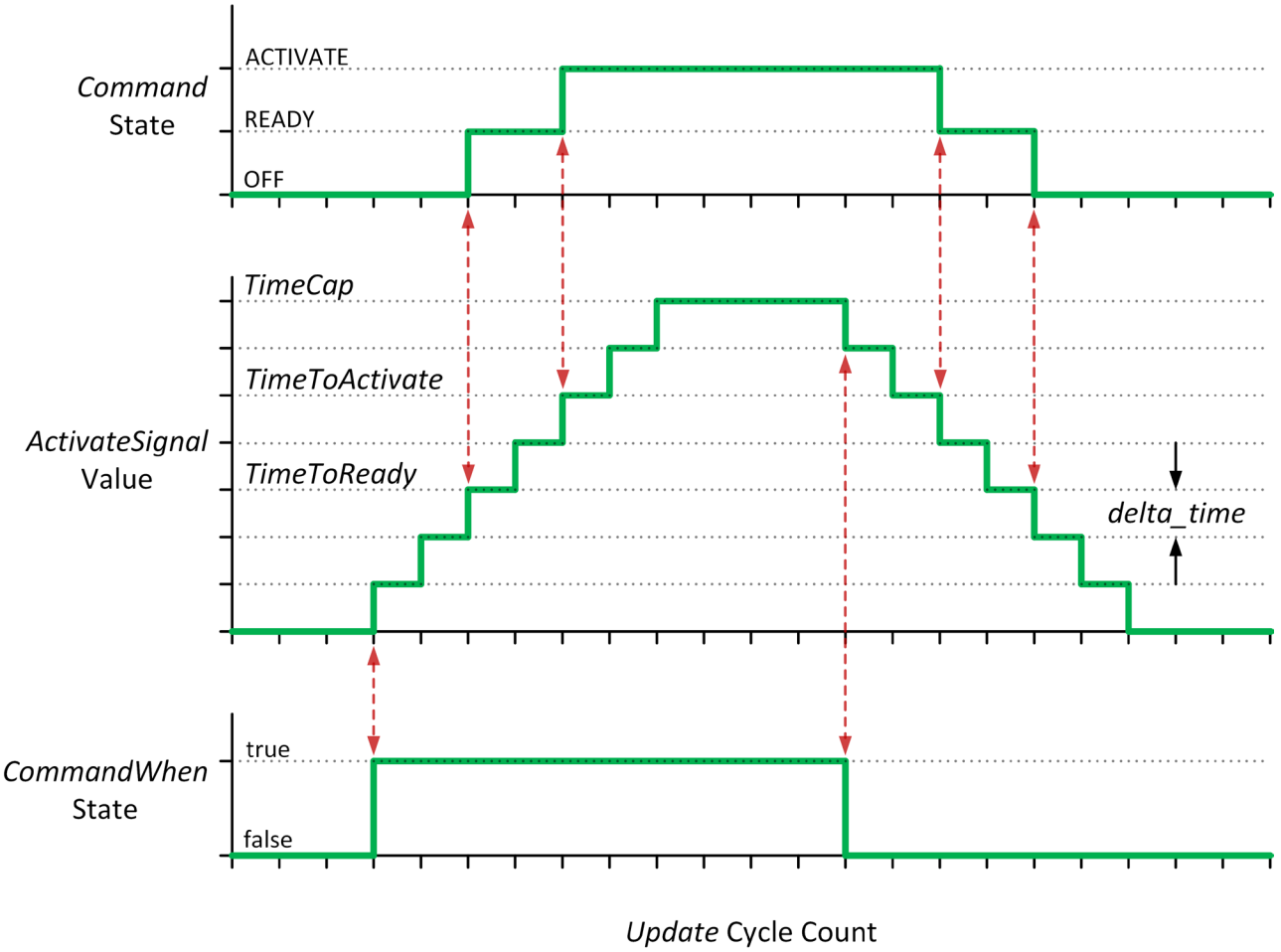
Map Boundary Sample: Destruct Lines

```
<Boundaries>
      <!-- Note: Boundary coordinates are positive East geodetic in decimal degrees. -->
   <Static_Boundary id ="Destruct_Lines" orientation="Clockwise" outerLat="39.0" outerLon="-77.0">
      <Vertices>
         <Vertex> <Lat>  7.007661 </Lat> <Lon> -32.692886 </Lon> </Vertex>
         <Vertex> <Lat> 11.236036 </Lat> <Lon> -38.107641 </Lon> </Vertex>
         <Vertex> <Lat> 16.894337 </Lat> <Lon> -43.349243 </Lon> </Vertex>
         <Vertex> <Lat> 20.767154 </Lat> <Lon> -48.056303 </Lon> </Vertex>
         <Vertex> <Lat> 24.252088 </Lat> <Lon> -52.638379 </Lon> </Vertex>
         <Vertex> <Lat> 26.412423 </Lat> <Lon> -55.826182 </Lon> </Vertex>
         <Vertex> <Lat> 29.380887 </Lat> <Lon> -60.954146 </Lon> </Vertex>
               •
               •
               •
         <Vertex> <Lat> 36.413871 </Lat> <Lon> -61.759196 </Lon> </Vertex>
         <Vertex> <Lat> 35.286065 </Lat> <Lon> -58.470732 </Lon> </Vertex>
         <Vertex> <Lat> 34.002417 </Lat> <Lon> -55.184778 </Lon> </Vertex>
         <Vertex> <Lat> 31.709156 </Lat> <Lon> -50.122673 </Lon> </Vertex>
         <Vertex> <Lat> 28.738015 </Lat> <Lon> -43.494405 </Lon> </Vertex>
         <Vertex> <Lat> 21.433278 </Lat> <Lon> -29.443836 </Lon> </Vertex>
         <Vertex> <Lat> 19.204944 </Lat> <Lon> -24.834595 </Lon> </Vertex>
         <Vertex> <Lat>  7.007661 </Lat> <Lon> -32.692886 </Lon> </Vertex>
      </Vertices>
   </Static_Boundary>
         •
         •
         •
</Boundaries>
```

Map Boundary Sample: Destruct Lines

```
<MissionRules>
   <MapBoundaryRule id="Destruct_Line_Debris_Rule">
     <Subset> GPS_A GPS_B HYBRID_A HYBRID_B INS_A INS_B </Subset>
     <Output>
         <InvalidWhen default="true">
             <cond> haveLiftOff is false             </cond> <or/>
             <cond> isGoodSensorData is false        </cond> <or/>
             <cond> impactComputed is false          </cond>
             <!-- As noted above, the following line is applicable to Wallops Flight Facility. -->
             <or/> <cond> Launch_Plane.impact.posDownRange &lt; Minimum_Destruct_Line_Distance </cond>
         </InvalidWhen>
         <Result default="false">
            <cond> Destruct_Line_Debris_Rule.isInside is false </cond>
         </Result>
     </Output>
     <RefBoundary> Destruct_Lines </RefBoundary>
     <RefCoordinates> Impact </RefCoordinates>
   </MapBoundaryRule>
         •
         •
         •
</MissionRules>
```

Stair Step Command



Voting (Boolean Case)

- A quorum is defined as 3 non-abstaining votes (In parenthesis is what is used on slide 13 sample file):
  - Zero is 0 non-abstaining votes (No decision can be made – result is false)
  - One is 1 non-abstaining vote (ballot_result is used – the 1 non-abstaining vote is used as result)
  - Two is 2 non-abstaining votes ("or" logic is used – if either say true, then result is true)
  - Tie is a quorum with equal amounts of non-abstaining votes for each outcome (true is result – true wins in a tie)

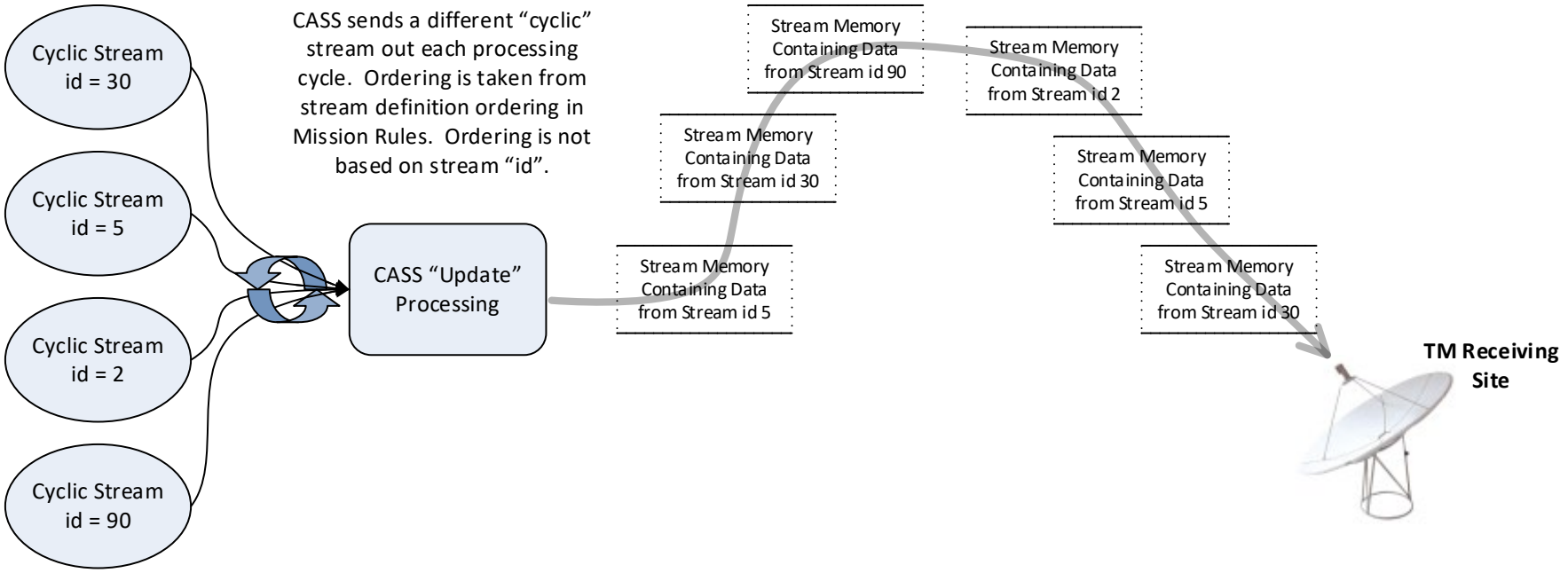
Mission Rules Development

Commands Sample: Activate FTS using:

- Stair step command
- Voting

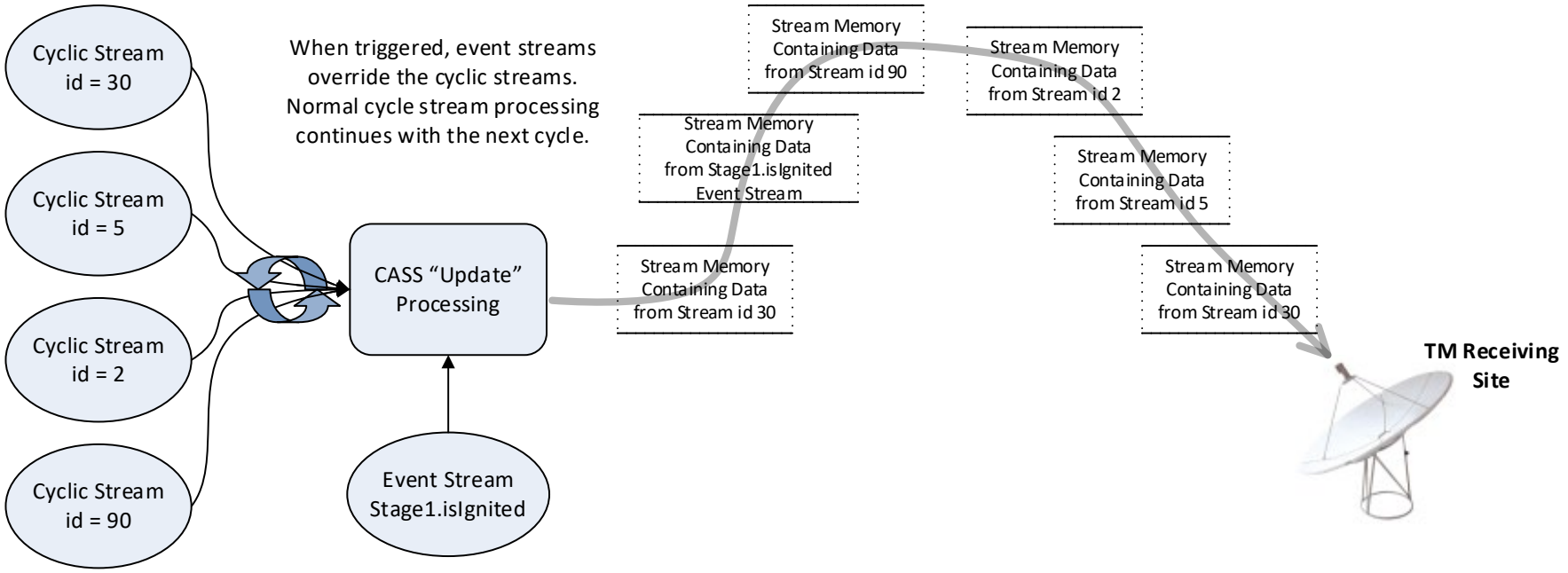
```
<Commands>
   <!--
      Activate the Flight Termination System
      This is a stair step command (tri-state: OFF ==> READY ==> ACTIVATE, with return, ACTIVATE ==> READY ==> OFF).
   -->
   <Command id="Activate_Flight_Termination_System">
      <TimeToReady>    0.3    </TimeToReady>     <!-- In seconds. -->
      <TimeToActivate> 0.5    </TimeToActivate>  <!-- In seconds. -->
      <TimeCap>        0.7    </TimeCap>         <!-- In seconds. -->
      <TimeSlop>       0.0001 </TimeSlop>        <!-- In seconds. -->
      <CommandWhen default="false">
         <vote zero="false" one="ballot-result" two="or" tie="true"> Straight_Up_Violation      </vote> <or/>
         <vote zero="false" one="ballot-result" two="or" tie="true"> Explosive_Impact_Rule      </vote> <or/>
         <vote zero="false" one="ballot-result" two="or" tie="true"> Chevron_Rule               </vote> <or/>
         <vote zero="false" one="ballot-result" two="or" tie="true"> Azimuth_Check_Rule         </vote> <or/>
         <vote zero="false" one="ballot-result" two="or" tie="true"> Destruct_Line_Debris_Rule  </vote> <or/>
         <cond>
            <cond> Pre_Stage_4_Terminate_Rule.Invalid is false </cond> <and/>
            <cond> Pre_Stage_4_Terminate_Rule.Result is true   </cond>
         </cond>
      </CommandWhen>
   </Command>
        •
        •
        •
</Commands>
```

Cyclic Streams



Streams allow the user to obtain information from the CASS flight software within the AFTU during testing and flight.

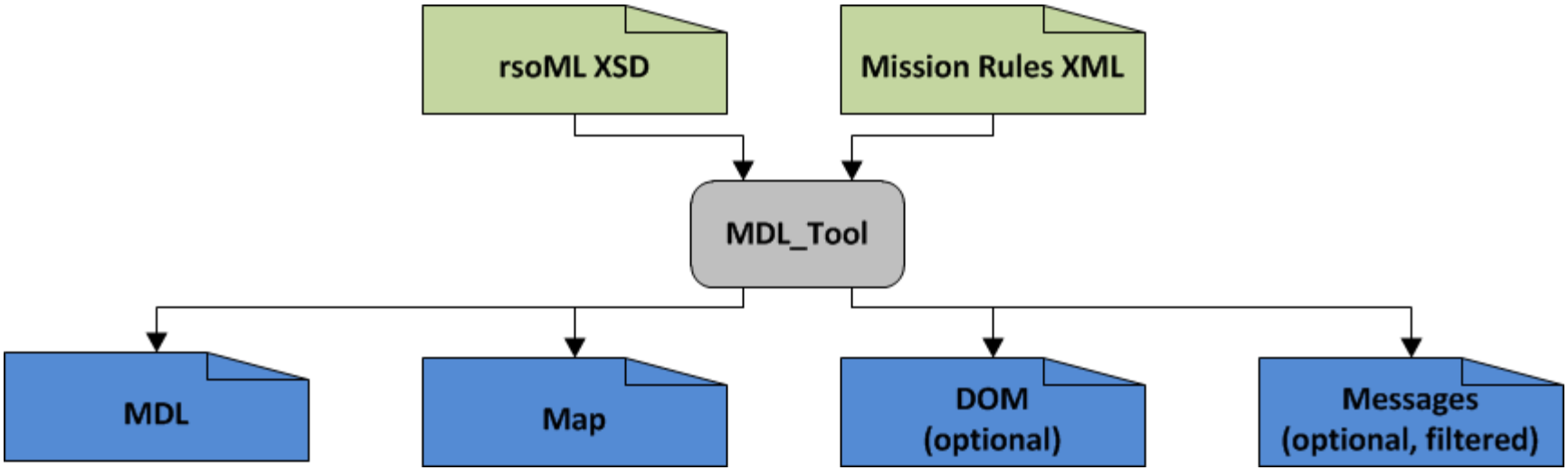
Event Streams



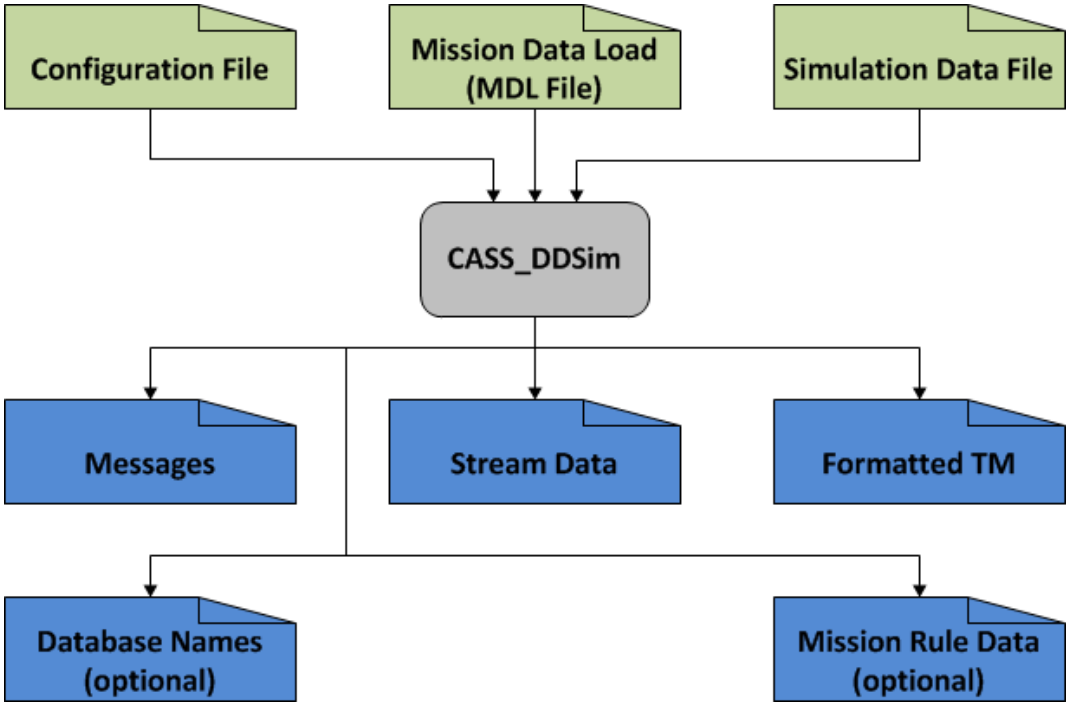
Streams allow the user to obtain information from the CASS flight software within the AFTU during testing and flight.

Mission Rules Development

- CASS Utility MDL_Tool
  - Reads and validates XML file against rsoML schema definition.
  - Performs data validation and data consistency checks.
  - Performs data dependency and data reference checks.
  - Provides diagnostic messages.
  - Generates MDL if no errors were detected during analysis.
  - Generates Stream Map file if any Streams are defined in XML file.

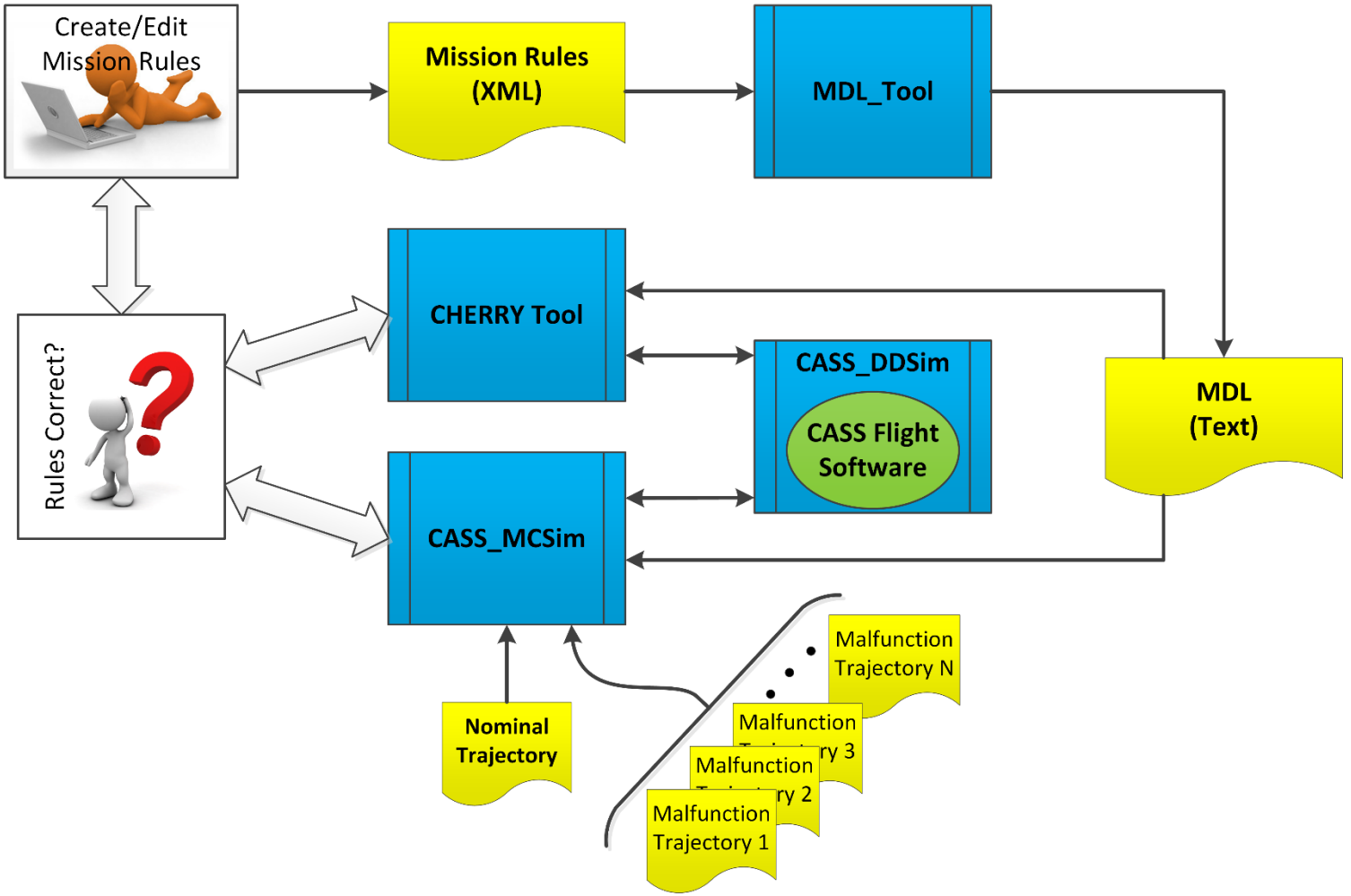


- CASS Data Driven Simulation (CASS_DDSim) Tool
  - A CASS Flight Software wrapper for the PC.
  - Passes simulated sensor data to CASS Flight Software and captures output for analysis.

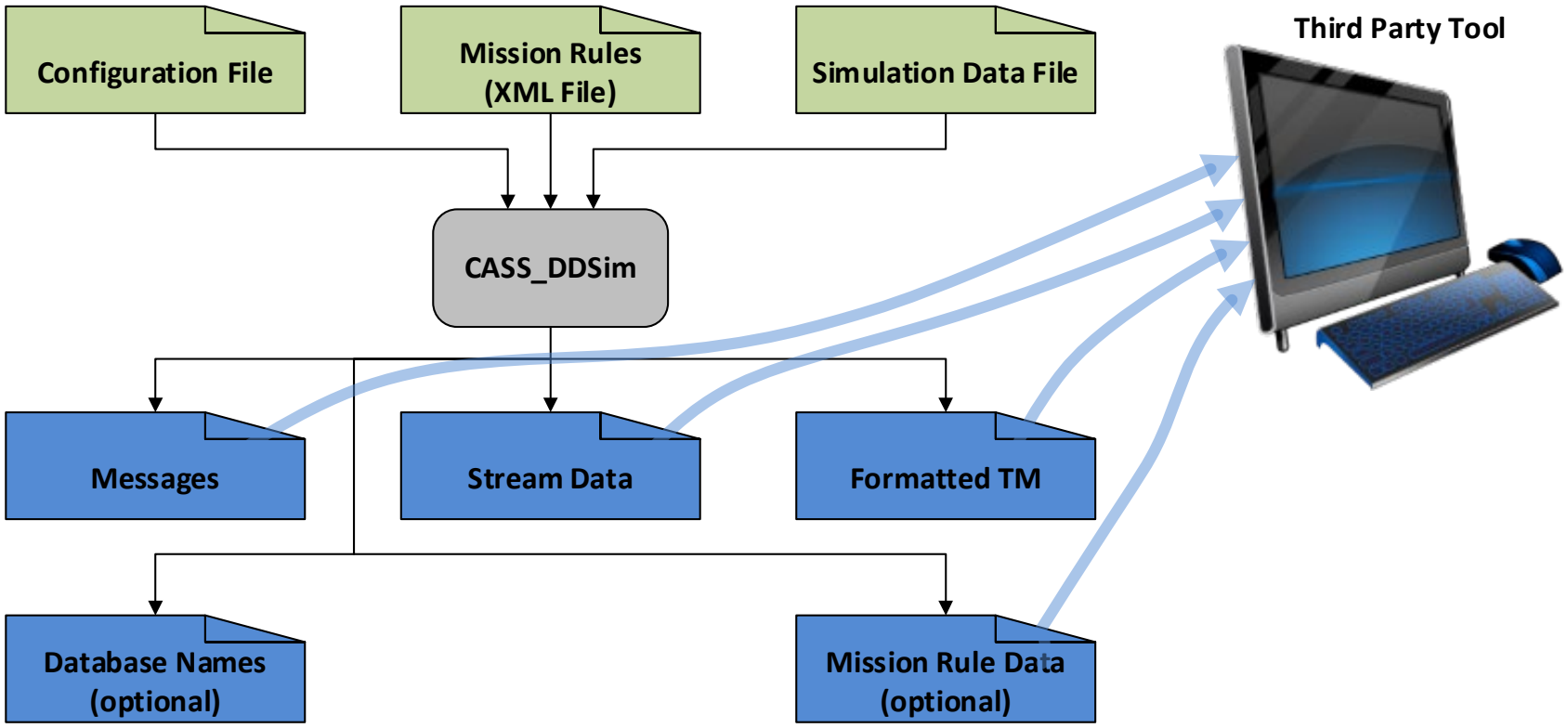


- CHERRY Tool
  - Executes under MATLAB without any special toolboxes.
  - Utilizes latest CASS_DDSim and MDL_Tool, or can be configured to use older versions of CASS_DDSim or MDL_Tool.
  - Converts many trajectory file formats into sensor input messages needed by CASS_DDSim.
  - Executes selected version of MDL_Tool to generate an MDL from an input XML Mission Rules file, then executes the selected version of CASS_DDSim.
  - Monitors CASS_DDSim results (static Mission Rule data and formatted telemetry data) and translates results to Earth grid maps.

MISSION PREPARATION

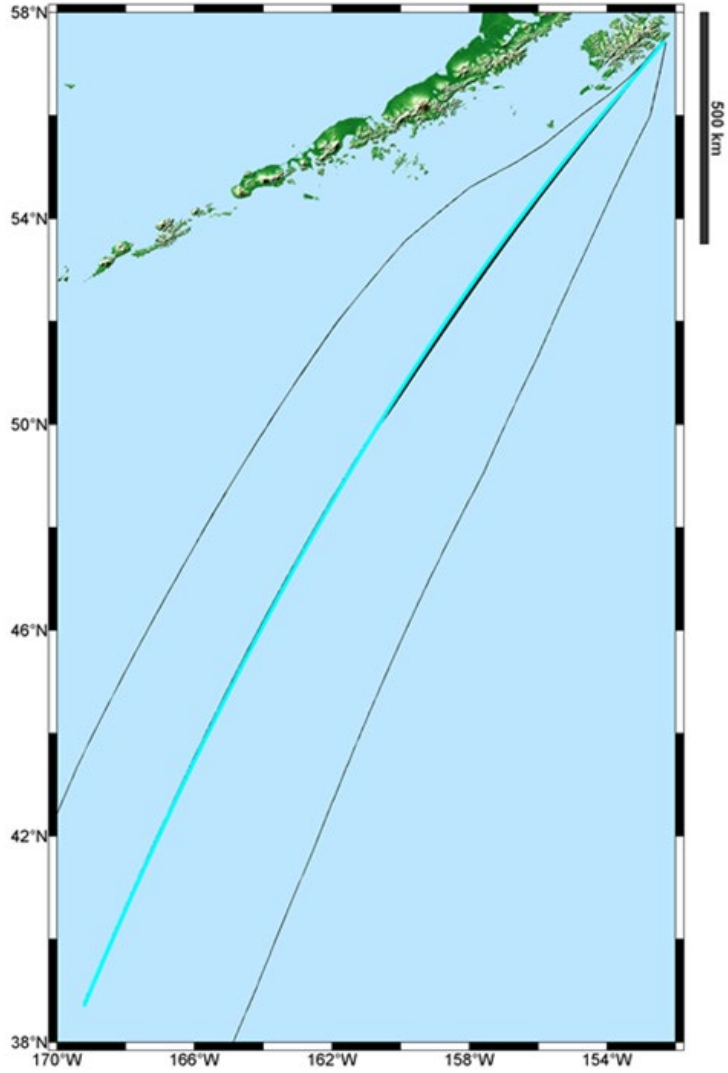


Typical CHERRY Tool Configuration

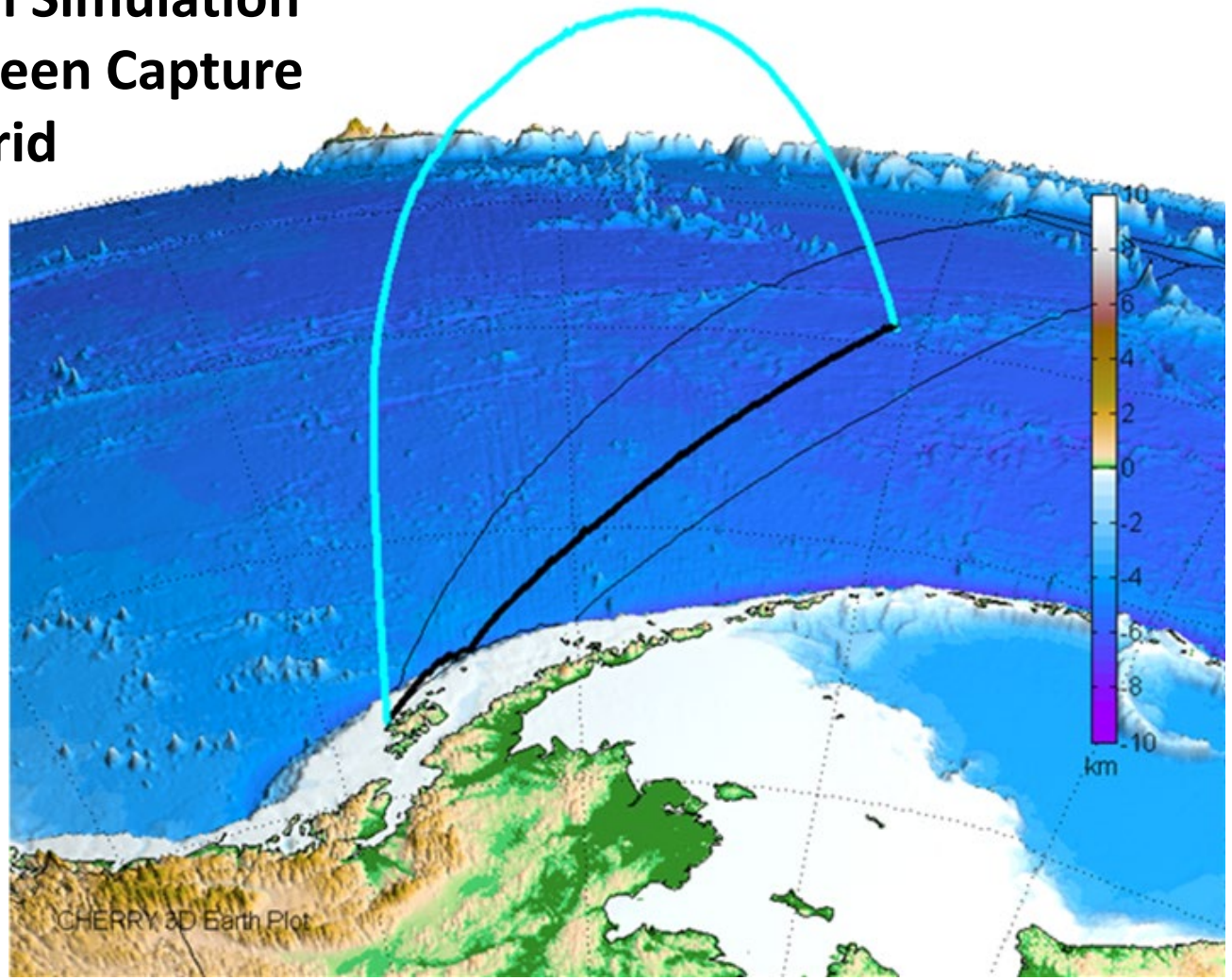


Testing and Simulation

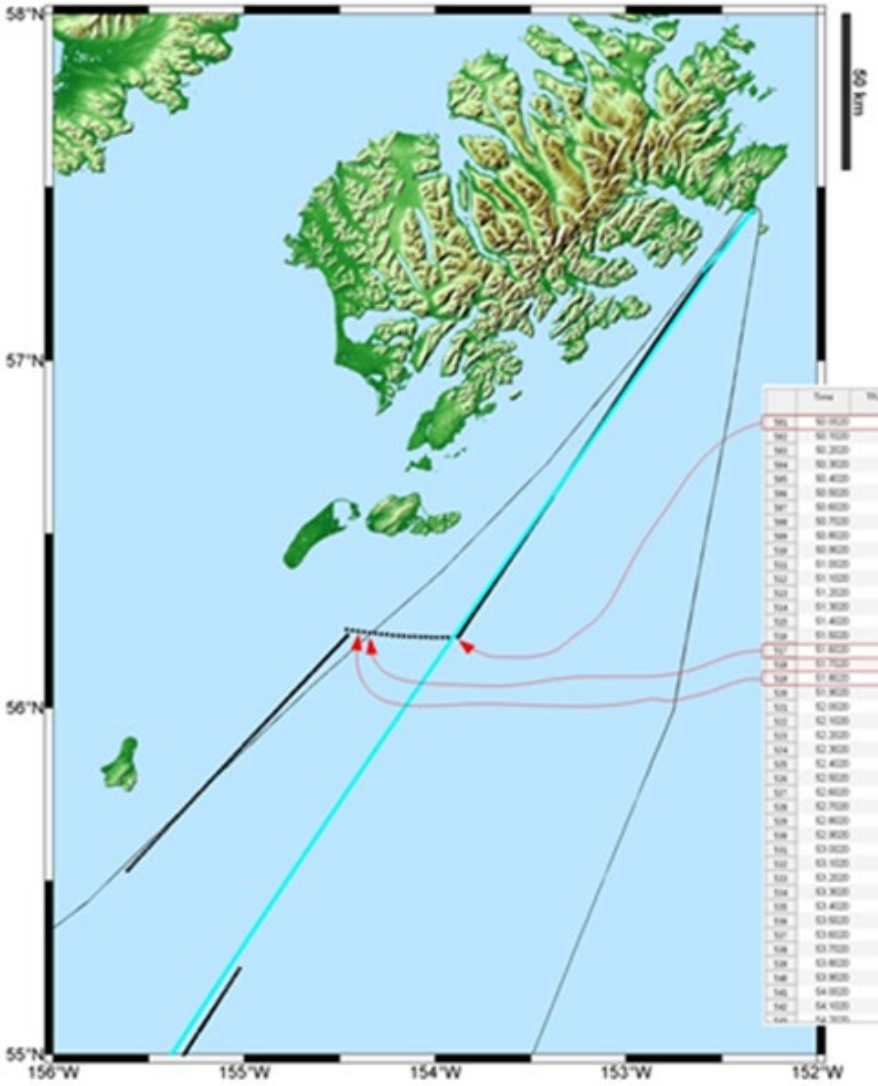
Nominal Mission Simulation
CHERRY Tool Screen Capture



Nominal Mission Simulation CHERRY Tool Screen Capture with 3D Earth Grid



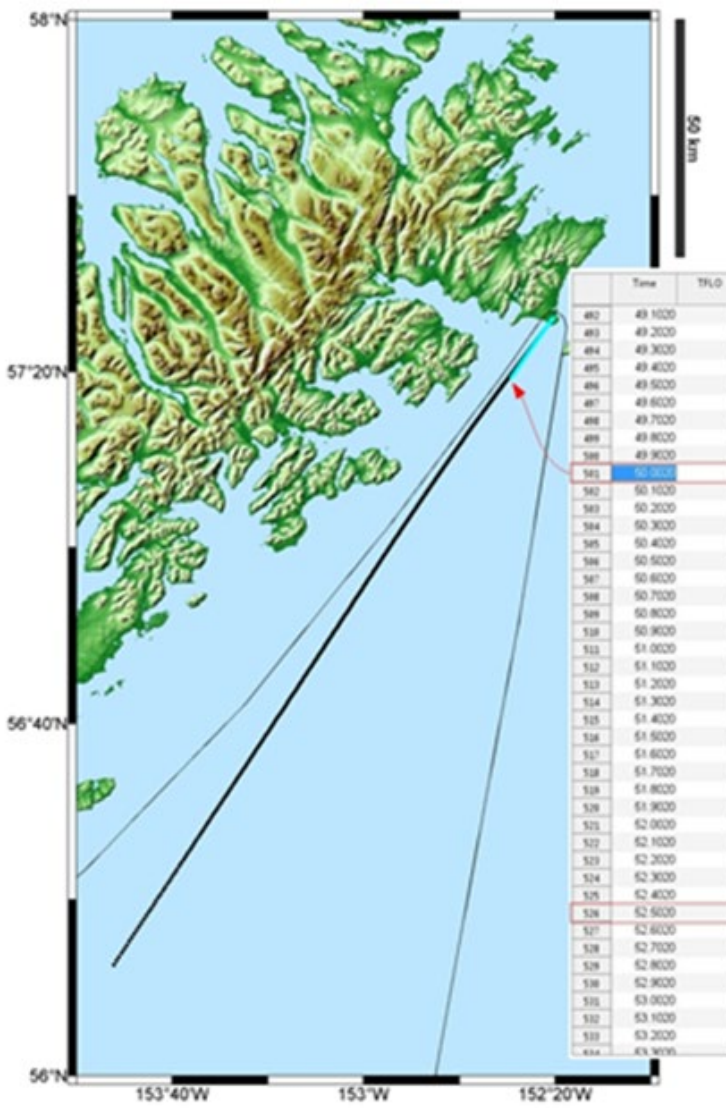
TESTING AND SIMULATION



Nominal Mission Simulation CHERRY Tool Screen Capture with 2D Earth Grid and Forced Malfunction Turn

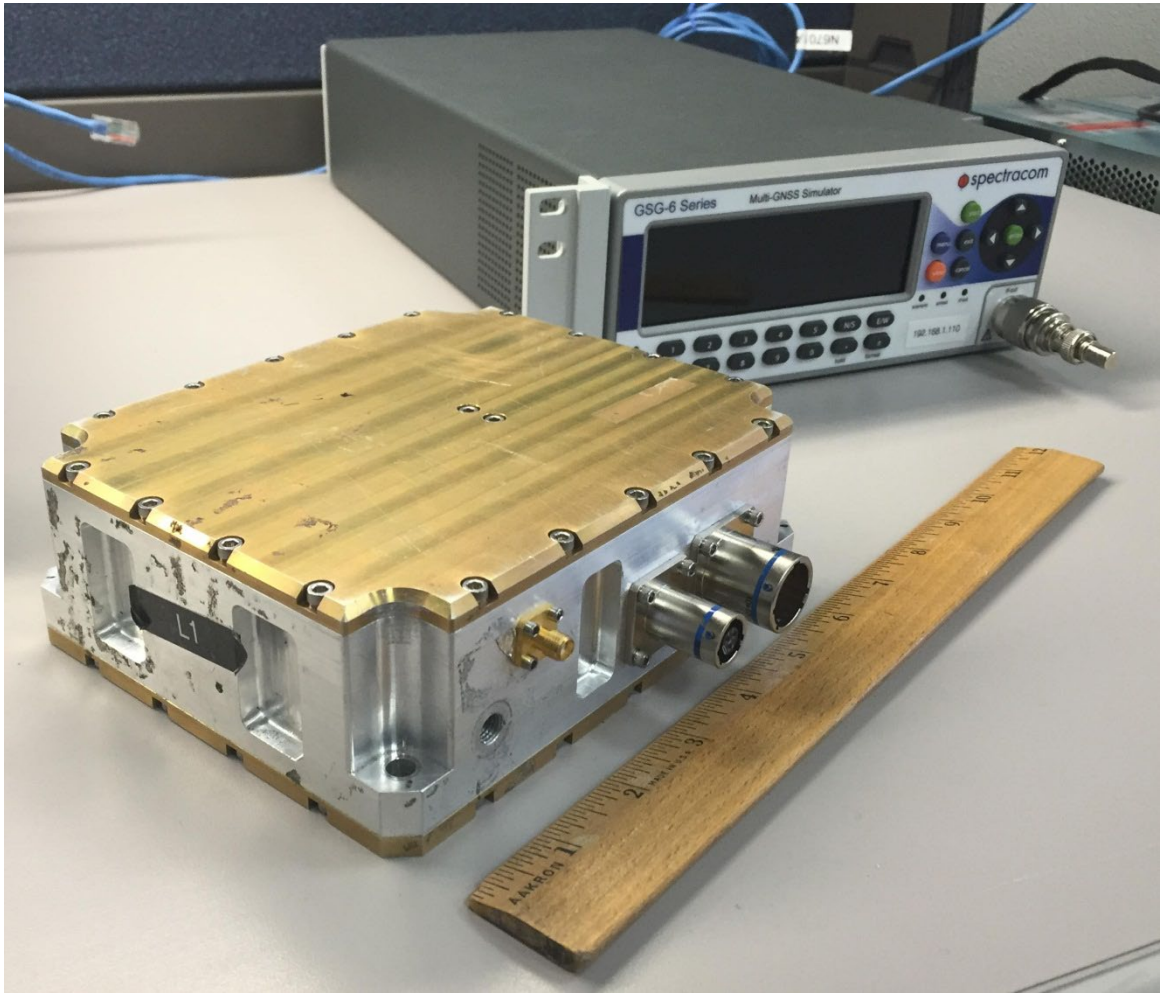
TESTING AND SIMULATION

Nominal Mission Simulation
CHERRY Tool Screen Capture
with 2D Earth Grid and
Forced Loss of Sensor Data



	Time	TFLO	Last Valid Data Time	Valid Sensor Count	Master Green Time	Latitude IP	Longitude IP	Latitude PP	Longitude PP	Have Destruct Condition	Have Liftoff	Have Safing Condition
492	49.1020	0	49.1000	1	2.3232	56.2687	-153.8001	57.3275	-152.4791	☐	☐	☐
493	49.2020	0	49.2000	1	2.3460	56.2615	-153.8088	57.3269	-152.4798	☐	☐	☐
494	49.3020	0	49.3000	1	2.3686	56.2544	-153.8175	57.3263	-152.4806	☐	☐	☐
495	49.4020	0	49.4000	1	2.3912	56.2472	-153.8263	57.3257	-152.4813	☐	☐	☐
496	49.5020	0	49.5000	1	2.4137	56.2400	-153.8351	57.3251	-152.4820	☐	☐	☐
497	49.6020	0	49.6000	1	2.4360	56.2327	-153.8439	57.3246	-152.4828	☐	☐	☐
498	49.7020	0	49.7000	1	2.4583	56.2254	-153.8529	57.3240	-152.4835	☐	☐	☐
499	49.8020	0	49.8000	1	2.4806	56.2181	-153.8618	57.3234	-152.4843	☐	☐	☐
500	49.9020	0	49.9000	1	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
501	50.0020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
502	50.1020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
503	50.2020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
504	50.3020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
505	50.4020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
506	50.5020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
507	50.6020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
508	50.7020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
509	50.8020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
510	50.9020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
511	51.0020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
512	51.1020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
513	51.2020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
514	51.3020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
515	51.4020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
516	51.5020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
517	51.6020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
518	51.7020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
519	51.8020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
520	51.9020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
521	52.0020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
522	52.1020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
523	52.2020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
524	52.3020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
525	52.4020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☐	☐	☐
526	52.5020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☑	☐	☐
527	52.6020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☑	☐	☐
528	52.7020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☑	☐	☐
529	52.8020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☑	☐	☐
530	52.9020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☑	☐	☐
531	53.0020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☑	☐	☐
532	53.1020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☑	☐	☐
533	53.2020	0	49.9000	0	2.5027	56.2107	-153.8708	57.3228	-152.4850	☑	☐	☐

AFTS UNIT



GETTING THE SOFTWARE

Request a copy of the CASS Software Usage Agreement (SUA) form or the CHERRY SUA. The form should be signed by a person that can legally obligate the company (i.e., recipient). Once the form is signed, send a PDF version of the scanned, signed, document to:

Primary POC	Alternate POC
Jeffrey D. Cherry SLD 30/SEAE Safety Engineer 805-606-5784 jeffrey.cherry.1@spaceforce.mil	Richard "Cass" Russett SLD 30/SEAE Safety Engineer 805-605-1724 richard.russett@spaceforce.mil

The CASS can be downloaded once received via a secure download site as a compressed archive containing source code for the CASS Flight Software, CASS Utilities, scripts, demonstration software, and documentation. CHERRY tool can be downloaded once received via a secure download site as a compressed archive containing the CHERRY tool with examples.