



🌀 MASCOT — MATLAB Stability and Control Toolbox

MASCOT software was created to provide the conceptual aircraft designer accurate predictions of air vehicle stability and control characteristics. The code takes as input mass property data in the form of an inertia tensor, aerodynamic loading data, and propulsion (i.e. thrust) loading data. Using fundamental non-linear equations of motion, MASCOT then calculates vehicle trim and static stability data for any desired flight condition. Common predefined flight conditions are included. The predefined flight conditions include six horizontal and six landing rotation conditions with varying options for engine out, crosswind and sideslip, plus three takeoff rotation conditions. Results are displayed through a unique graphical interface developed to provide stability and control information to the conceptual design engineers using a qualitative scale indicating whether the vehicle has acceptable, marginal, or unacceptable static stability characteristics.

This software allows the user to prescribe the vehicle's CG location, mass, and inertia tensor so that any loading configuration between empty weight and maximum take-off weight can be analyzed. The required geometric and aerodynamic data as well as mass and inertia properties may be entered directly, passed through data files, or come from external programs such as Vehicle Sketch Pad (VSP). The current version of MASCOT has been tested with VSP used to compute the required data, which is then passed directly into the program. In VSP, the vehicle geometry is created and manipulated. The aerodynamic coefficients, stability and control derivatives, are calculated using VorLax, which is now available directly within VSP. MASCOT has been written exclusively using the technical computing language MATLAB®.

This innovation is able to bridge the gap between low-fidelity conceptual design and higher-fidelity stability and control analysis. This new tool enables the conceptual design engineer to include detailed static stability and trim constraints in the conceptual design loop. The unique graphical interface devel-

oped for this tool presents the stability data in a format that is understandable by the conceptual designer, yet also provides the detailed quantitative results if desired.

This work was done by Sean Kenny of Langley Research Center and Luis Crespo of the National Institute of Aerospace. Further information is contained in a TSP (see page 1), LAR-16581-1

🌀 MIRO Continuum Calibration for Asteroid Mode

MIRO (Microwave Instrument for the Rosetta Orbiter) is a lightweight, uncooled, dual-frequency heterodyne radiometer. The MIRO encountered asteroid Steins in 2008, and during the flyby, MIRO used the Asteroid Mode to measure the emission spectrum of Steins. The Asteroid Mode is one of the seven modes of the MIRO operation, and is designed to increase the length of time that a spectral line is in the MIRO pass-band during a flyby of an object. This software is used to calibrate the continuum measurement of Steins' emission power during the asteroid flyby.

The MIRO raw measurement data need to be calibrated in order to obtain physically meaningful data. This software calibrates the MIRO raw measurements in digital units to the brightness temperature in Kelvin. The software uses two calibration sequences that are included in the Asteroid Mode. One sequence is at the beginning of the mode, and the other at the end. The first six frames contain the measurement of a cold calibration target, while the last six frames measure a warm calibration target. The targets have known temperatures and are used to provide reference power and gain, which can be used to convert MIRO measurements into brightness temperature.

The software was developed to calibrate MIRO continuum measurements from Asteroid Mode. The software determines the relationship between the raw digital unit measured by MIRO and the equivalent brightness temperature by analyzing data from calibration frames. The found relationship is applied to non-calibration frames, which are the measurements of an object of interest such as asteroids and other

planetary objects that MIRO encounters during its operation.

This software characterizes the gain fluctuations statistically and determines which method to estimate gain between calibration frames. For example, if the fluctuation is lower than a statistically significant level, the averaging method is used to estimate the gain between the calibration frames. If the fluctuation is found to be statistically significant, a linear interpolation of gain and reference power is used to estimate the gain between the calibration frames.

This work was done by Seungwon Lee of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

The software used in this innovation is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47075.

🌀 GOATS Image Projection Component

When doing mission analysis and design of an imaging system in orbit around the Earth, answering the fundamental question of imaging performance requires an understanding of the image products that will be produced by the imaging system. At the highest level, this understanding can be gained by a first-principles analysis of the geometric image projections involved in image capture from space. Going back to first principles allows for requirements to be quickly analyzed and the fundamental specifications of the imaging systems to be tweaked and traded quickly to aid in rapid mission design and analysis. The problem then becomes encapsulating these first-principle algorithms in a set of easy-to-use and modular functions to be used in coordination with other mission design tools.

GOATS software represents a series of MATLAB functions to provide for geometric image projections. Unique features of the software include function modularity, a standard MATLAB interface, easy-to-understand first-principles-based analysis, and the ability to perform geometric image projections of framing type imaging systems. The software modules are created for maximum analysis utility, and can all be used inde-

pendently for many varied analysis tasks, or used in conjunction with other orbit analysis tools.

By basing the tools purely on first principles and utilizing a high degree of generalization, the same set of tools can be applied to imaging systems not in orbit (i.e., helicopters, high-altitude planes, etc.). The image projection tools begin with a simple description of the imaging system to be analyzed. This description is based on the highest level of imaging parameters (system F-number, pixel pitch, aperture size, etc.), which are among the first parameters settled upon when contemplating a new system. The tools are structured this way to allow for early analysis of the highest level requirements and the performance of rather detailed trade studies to best settle upon these high-level requirements early on.

Using these high-level parameters and the imaging geometry of the proposed system, one can geometrically project the imaging system onto the ground and determine many of the features of the collected imagery. These features include ground sample distances, actual image size and geolocation, image stretching and warping, etc. With these features calculated, there is a direct link between imaging system hardware parameters and performance requirements. Additionally, the same tools can be used to do some level of image processing. Existing image data

sets can be processed and re-projected to simulate what they would look like had they been taken with an imaging system with different parameters. The functions provide a robust toolset to quickly answer several types of imaging questions that typically arise when considering missions around framing imaging systems.

This work was done by Benjamin M. Haber and Joseph J. Green of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47281.

Coded Modulation in C and MATLAB

This software, written separately in C and MATLAB as stand-alone packages with equivalent functionality, implements encoders and decoders for a set of nine error-correcting codes and modulators and demodulators for five modulation types. The software can be used as a single program to simulate the performance of such coded modulation.

The error-correcting codes implemented are the nine accumulate repeat-4 jagged accumulate (AR4JA) low-density parity-check (LDPC) codes, which have been approved for international standardization by the Consultative Committee for Space Data Systems, and

which are scheduled to fly on a series of NASA missions in the Constellation Program. The software implements the encoder and decoder functions, and contains compressed versions of generator and parity-check matrices used in these operations.

The software supports the modulations of binary phase-shift keying (BPSK), quadrature PSK (QPSK), 8-PSK, 16-ary amplitude PSK (16-APSK), and 32-APSK. For each modulation type, the software modulator supports various bit-to-modulation-symbol mappings, including the natural order, the Gray code, the anti-Gray code, and the ordering specified by the Digital Video Broadcast Satellite Second Generation standard for 16-APSK and 32-APSK. The software supports hard and soft demodulation, and when soft, it supports both an exact log likelihood computation and an approximate log likelihood computation based on nearest neighbors.

The software supports all nine AR4JA LDPC codes of the CCSDS standard and all five modulations of the DVB-S2 standard, in any combination.

This work was done by Jon Hamkins and Kenneth S. Andrews of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47171.