

Attitude Visualization and Animation Tool

User Manual **Version 1.3**

Ben K. Bradley and Suneel I. Sheikh
3 May 2013



© 2009-2013 ASTER Labs, Inc.
All Rights Reserved.
www.asterlabs.com

Contents

OVERVIEW	3
SYSTEM REQUIREMENTS	3
INSTALLATION	4
IMPORTANT NOTES	4
WHAT'S NEW IN VERSION 1.3?	5
SOFTWARE OVERVIEW	6
MAIN MENU	6
ANIMATE PRINCIPAL ROTATION VECTOR (PRV)	7
ANIMATE EULER	7
ANIMATE FILE	7
ANIMATION FILE SETTINGS	7
SETTINGS	8
MATHEMATICAL OVERVIEW.....	9
DIRECTION COSINE MATRIX.....	9
EULER ANGLES	9
PRINCIPAL ROTATION VECTOR (PRV)	10
QUATERNION (EULER PARAMETERS)	10
ALTERNATE QUATERNION.....	11
CLASSIC RODRIGUES PARAMETERS (CRPs).....	11
MODIFIED RODRIGUES PARAMETERS (MRPs)	11
REFERENCES.....	12

Overview

Aptus is a graphical user interface (GUI) written in MATLAB accomplished at displaying attitude rotations and animating attitude time-histories.

This software is a perfect tool for students and professors of attitude dynamics. It is also an excellent tool for use in industry for quick visualizations of attitude time-histories. The user interface is intuitive and powerful, taking only seconds to play back a high quality attitude simulation usually reserved for expensive commercial tools.

System Requirements

Aptus is designed to work within MATLAB version 7.5 (2007b) and higher.

MATLAB versions earlier than 2007b are not supported due to changes in *.fig files and pre-parsed pseudocode, or p-code, encryption that MATLAB has changed over the years.

However, if you have a strong need for using versions earlier than MATLAB version 7.5 for your application, please contact our support group for further assistance:

support@asterlabs.com

Aptus will work on any platform, including Mac OS, Windows, or Linux, which supports the MATLAB versions listed above.

MATLAB® is a high-level language and interactive development environment. It is produced by The MathWorks, Inc.

Users are referred to their website for more information:

<http://www.mathworks.com/products/matlab/>

Installation

- (1) Download the zipped folder from the ASTER Labs web store after your purchase:
<http://www.asterlabs.com/store>
- (2) If not done automatically by your web browser, unzip the downloaded file.
- (3) Put the unzipped Aptus folder into a directory or folder of your choice.
- (4) Open up MATLAB and navigate to the Aptus folder
- (5) At the MATLAB command line, type the following command and then hit enter:

```
>> Install_Aptus
```
- (6) Aptus is now installed. A message should appear in the command window with a summary of the Aptus license information.
- (7) This Aptus license is tied to the MATLAB license number that was entered when the software was purchased.
- (8) To start Aptus, type the following in the MATLAB command line and hit enter:

```
>> Aptus
```

Important Notes

- (a) If user interface windows appear bigger than your screen, you may need to change the resolution of your monitor.
- (b) Information about your Aptus license and version can be viewed by entering the following at the MATLAB command line:

```
>> About_Aptus
```
- (c) We would happily take suggestions or noted corrections. Please submit these directly to us via email at support@asterlabs.com.

What's New in Version 1.3?

There are some great new features that have been added to Aptus for this new release. All of the updates are listed below:

- Aptus now supports multiple vehicle image types that can replace the standard spacecraft model in the animation window. Current set includes satellites and spacecraft, as well as aircraft.
- The user now has the option to insert vectors or arrows as needed to help visualize inertial and body-fixed locations during the attitude animation.
- The Settings options have been improved for additional functionality by the user.
- A save/load feature has been added, allowing the user to save animation settings and load them back in at a later date.
- A number of functions have been sped up through increased vectorization of the code.

Software Overview

Main Menu

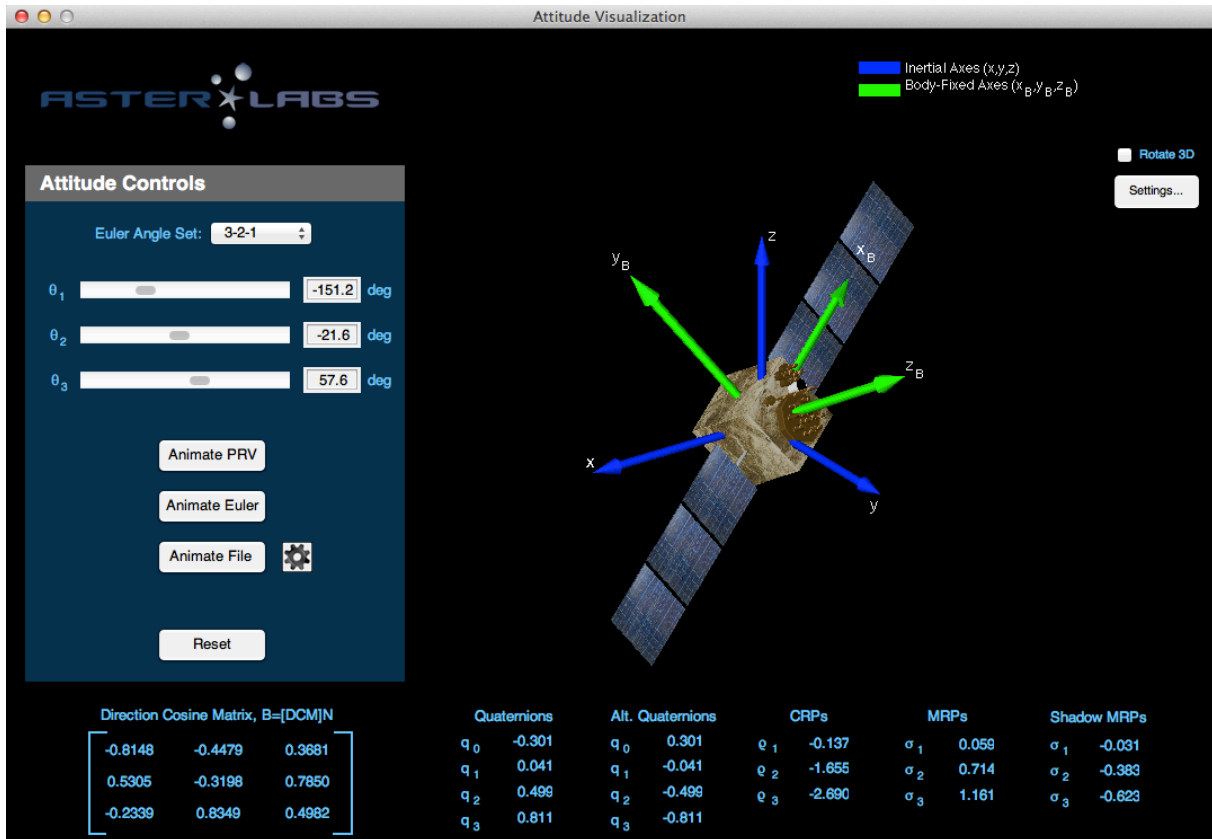


Figure 1. Main screen of Aptus.

Figure 1 shows the first screen that appears after starting the Aptus program. As one of the significant features of the program, Aptus allows the user to specify Euler angles to describe the orientation of a spacecraft in inertial space. The user can either input specific values, or utilize the slider bar to vary the Euler angles.

As part of Aptus, the user can animate the spacecraft to the current attitude using the Principal Rotation Vector (PRV), or through the Euler angle sequence, or by successively pressing the slider buttons next to each Euler angle value.

Alternate attitude sets are displayed below the satellite (or other vehicle) image and updated each instance the attitude is adjusted. In addition to the Euler Angles, these alternate sets include Direction Cosine Matrices, Quaternions, Alternate Quaternions, Classic Rodriguez Parameters, Modified Rodriguez Parameters, or Shadow Modified Rodriguez Parameters.

Animate Principal Rotation Vector (PRV)

This button will animate the vehicle in the animation window about the Principal Rotation Vector in a single rotation sequence until the final attitude orientation is achieved.

Animate Euler

This button will animate the vehicle in the animation window about the selected Euler axis, sequentially rotating the vehicle about each of the three axes until the final attitude orientation is achieved.

Animate File

By selecting this button, the user can elect to use Aptus to display an animation of their vehicle using their own recorded attitude parameter file. The file must be created according to the attitude parameter method selected. Example formatted files are provided with Aptus for users to develop their own files, which typically are a time value and the attitude parameters listed in rows within the file (see demo_MRP.mat, demo_MRP.txt, demo_quaternion.mat, and demo_quaternion.txt for examples).

Animation File Settings

The button beside the “Animate File” button within Aptus provides user inputs to the operation of the program, including Animation File Settings and Additional Arrows.

The Animation File Settings defines the specific files, attitude parameter type, and display frequency when using the Animate File button. This settings window is shown in Figure 2 below.

The Additional Arrows window allows the user to define arrows in the animation window used to help determine inertial locations during the animation process. Existing arrows for the inertial and body-fixed coordinate frame are provided, and the user can define their own arrows as needed. The included file, demo_arrow.txt, is an example file for a time-varying inertial arrow.

Lastly, the user can save and load animation settings via the “Save” and “Load” buttons in the bottom left of the Animation File Settings window. An example animation with arrows is given in the demo_animation1.mat file. Simply load this and run the simulation to get a better idea of how saving/loading works and how arrows can be used to display the information you want.

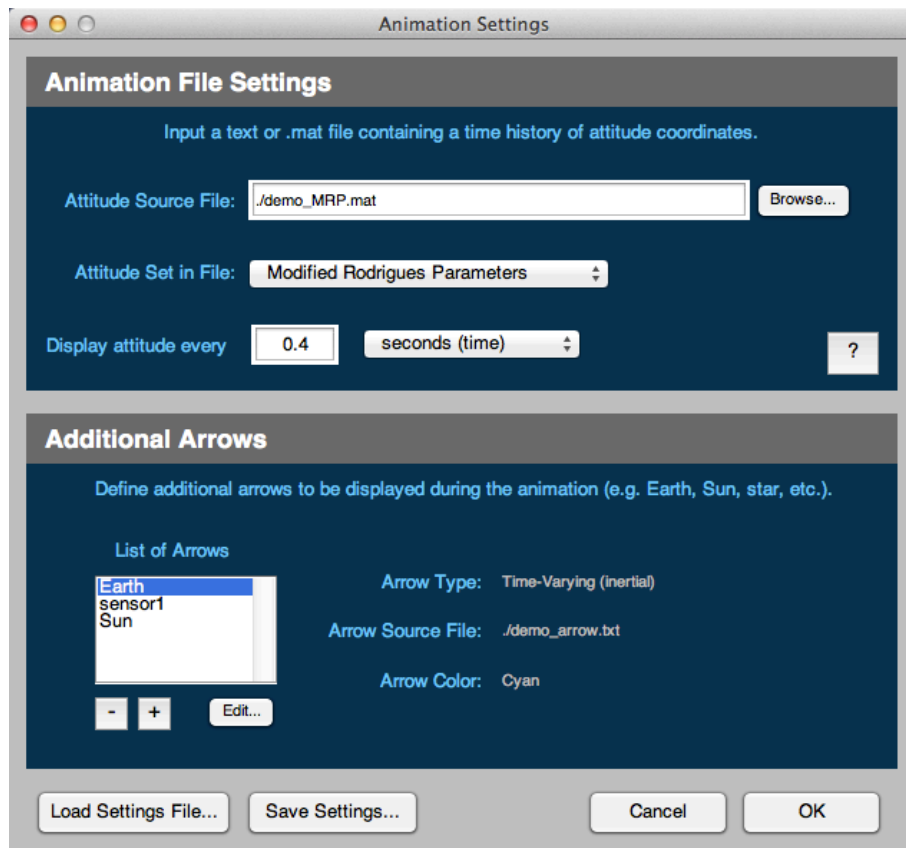


Figure 2. Animation Settings window of **Aptus**. The user can specify attitude time-history files, add/remove arrows, and save/load all simulation settings.

Settings

This button within the **Aptus** main window allows the user to select a new Object Type to be displayed in the animation window. It also allows the user to modify the orientation of the Object with respect the body-fixed axes. Object types currently include a GPS satellite, the Hubble Space Telescope, Boeing 747 aircraft, and Global Hawk UAV to name a few. Figure 3 demonstrates what these four objects look like in **Aptus**.

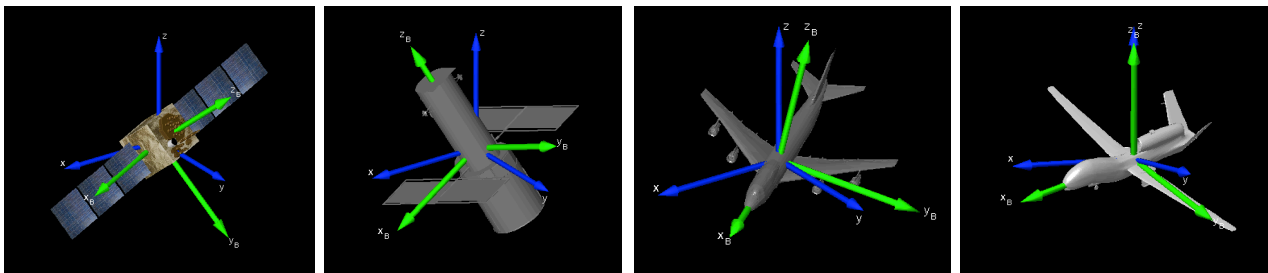


Figure 3. Example Objects types that can be displayed by **Aptus**. From left to right, GPS satellite, Hubble Space Telescope, Boeing 747, Global Hawk UAV.

Mathematical Overview

Direction Cosine Matrix

The Direction Cosine Matrix (DCM), denoted by $[C]$, is also displayed in the lower left corner of the window such that,

$$\mathbf{B} = [C]\mathbf{N} \quad (1)$$

where \mathbf{N} is a vector in the inertial frame and \mathbf{B} is a vector in the body-fixed frame. Each of the following subsections briefly overview the computation of each attitude set for completeness.

Euler Angles

Euler angles are very common attitude parameters that relate the body-fixed frame to the inertial frame through the rotation of three angles ($\theta_1, \theta_2, \theta_3$). Each rotation is about a body-fixed axis (x, y, z) expressed as (1, 2, 3), respectively. Each axis rotation is defined by,

$$[M_1(\theta)] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \quad (2a)$$

$$[M_2(\theta)] = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad (2b)$$

$$[M_3(\theta)] = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2c)$$

An Euler angle set (α, β, γ) using the angles ($\theta_1, \theta_2, \theta_3$) is written as

$$[C] = [M_\gamma(\theta_3)][M_\beta(\theta_2)][M_\alpha(\theta_1)] \quad (3)$$

which is used in Eq. (1).

Principal Rotation Vector (PRV)

The Principal Rotation Vector describes any orientation with a rotation axis \hat{e} and angle about that axis Φ . The rotation axis and angle are computed using components of the Direction Cosine Matrix $[C]$ as,

$$\cos \Phi = \frac{1}{2}(C_{11} + C_{22} + C_{33} - 1)$$

$$\hat{e} = \frac{1}{2 \sin \Phi} \begin{pmatrix} C_{23} - C_{32} \\ C_{31} - C_{13} \\ C_{12} - C_{21} \end{pmatrix} \quad (4)$$

where the rotation axis is described in the inertial frame as,

$$\hat{e} = e_1 \hat{n}_1 + e_2 \hat{n}_2 + e_3 \hat{n}_3 . \quad (5)$$

Quaternion (Euler Parameters)

Quaternions (or *Euler parameters*) are also commonly used parameters due to the fact that they are nonsingular and can handle large rotations. Quaternions are a redundant attitude set consisting of four parameters. Sheppard [2] introduced an algorithm for computing the quaternions from the DCM by first calculating the squares as,

$$q_0^2 = \frac{1}{4}(1 + \text{trace}[C])$$

$$q_1^2 = \frac{1}{4}(1 + 2C_{11} - \text{trace}[C])$$

$$q_2^2 = \frac{1}{4}(1 + 2C_{22} - \text{trace}[C])$$

$$q_3^2 = \frac{1}{4}(1 + 2C_{33} - \text{trace}[C]) \quad (6)$$

which satisfy the constraint

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1. \quad (7)$$

To compute the individual quaternions, we first take the square root of the largest q_i^2 in Eq. (6) above. The rest of the q_j values are computed by dividing three of the equations below by q_i .

$$\begin{aligned}
 q_0 q_1 &= (C_{23} - C_{32}) / 4 \\
 q_0 q_2 &= (C_{31} - C_{13}) / 4 \\
 q_0 q_3 &= (C_{12} - C_{21}) / 4 \\
 q_2 q_3 &= (C_{23} + C_{32}) / 4 \\
 q_3 q_1 &= (C_{31} + C_{13}) / 4 \\
 q_1 q_2 &= (C_{12} + C_{21}) / 4
 \end{aligned} \tag{8}$$

Alternate Quaternion

An alternate set of quaternions can also be defined as $\mathbf{q}' = -\mathbf{q}$. These are provided in the lower display of alternate attitude parameters. This *alternate* quaternion (or *complex conjugate* quaternion) represents the same attitude that is achieved with a rotation about the opposite rotation axis, $-\hat{e}$, and with a negative angle, $-\Phi$, from the Principal Rotation Vector.

Classic Rodrigues Parameters (CRPs)

The classical Rodrigues parameter vector is a minimal set of three parameters (ρ_1, ρ_2, ρ_3) . The CRPs have a singularity at a principal rotation of $\pm 180^\circ$ and behave more linearly for small angle rotations than Euler angle sets. The CRPs can be computed from the DCM by

$$\boldsymbol{\rho} = \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{pmatrix} = \frac{1}{\xi^2} \begin{pmatrix} C_{23} - C_{32} \\ C_{31} - C_{13} \\ C_{12} - C_{21} \end{pmatrix} \tag{9}$$

where

$$\xi^2 = \text{trace}([C]) + 1. \tag{10}$$

Modified Rodrigues Parameters (MRPs)

The MRPs are another three-parameter set with a singularity only for principal rotations of $\pm 360^\circ$. The MRPs are defined by the quaternions

$$\sigma_i = \frac{q_i}{1 + q_0}, \quad i = 1, 2, 3. \tag{11}$$

References

There are numerous very good text and papers on attitude representation and dynamics. Some of those that we have found helpful are provided below.

- [1] Hughes, P. C., *Spacecraft Attitude Dynamics*, Dover, New York, 2004.
- [2] Kuipers, J. B., *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*, Princeton University Press, Princeton, New Jersey, 1999.
- [3] Schaub, H. and Junkins, J.L., *Analytical Mechanics of Space Systems*, 2nd ed., American Institute of Aeronautics and Astronautics Inc., 2009.
- [4] Sheppard, S.W., “Quaternion from Rotation Matrix,” *Journal of Guidance and Control*, Vol. 1, No. 3, 1978, pp. 223-224.
- [5] Sidi, M. J., *Spacecraft Dynamics and Control*, Cambridge University Press, Cambridge UK, 1997.
- [6] Wertz, J. R. Ed., *Spacecraft Attitude Determination and Control*. Kluwer Academic Publishers, Boston MA, 1978.

Enjoy!

We hope you find Aptus useful.

We would happily take suggestions or noted corrections. Please submit these directly to us via email at support@asterlabs.com.

Development Group

Aptus is developed by ASTER Labs, Inc.

Our development team includes Ben K. Bradley and Suneel I. Sheikh.