Open-Source, Extensible Spacecraft Simulation And Modeling Environment Framework

Andrew Turner
Masters Thesis Presentation
July 17th, 2003
Outline of Presentation

- Introduction
- Background
- Purpose
- Architecture Design
- Software Implementation
- Verification and Validation
- Simulation Examples
- Conclusions
Introduction

- Open-SESSAME Framework
  - Open-Source
  - Extensible
  - Spacecraft
  - Simulation And Modeling
  - Environment
  - Framework

Aerospace Engineering
- Attitude dynamics
- Orbit dynamics
- Satellite environment
- Terminology
- Aerospace tools
- Typical use cases
- Satellite Community

Spacecraft Simulation

Computer Science / Software Engineering
- Object-Oriented Design
- Computational Speed
- Networking
- Data Handling
- Extensibility
- Software development tools
- Open-Source Community
Purpose

- Provide a useful spacecraft simulation modeling tool to engineers
  - Complement existing engineering tools
  - Include underdeveloped methodologies
  - Allow for infinite extensibility
- Educate students and other users how to design and implement simulation in software
  - Readable and useful documentation and software
  - Logical architecture design
- Develop an open-source framework that can be maintained and extended longer than the original project’s lifetime
- Host on a public repository and include developers in the community
- Foster communication between developers and users
Architecture Design

- Collections of libraries
  - Interchangeable components
  - Application framework
  - ‘Assemble’ to build an application
  - Suggested Architecture
  - Reconfigurable

- Dynamics Library
- Attitude Library
- Orbit Library
- Environment Library
- Rotation Library
- Communications
- Data Handling
Software Implementation

- C++ software language
- Open-source libraries and toolkits
  - Matrix
  - Network Communications
- Documentation
  - Verbose coding
  - Doxygen
Example: Rotation Library

- **Rotation** Class
  - Abstract
  - Conversion Routines
  - Stored as a **Quaternion**

<table>
<thead>
<tr>
<th>Rotation</th>
<th>DirectionCosine Matrix</th>
<th>Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Quat</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+SetDCM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+SetEulerAngles</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+SetEulerAxisAngle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+SetMRP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+SetQuaternion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+GetDCM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+GetEulerAxisAngle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+GetMRP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+GetQuaternion</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quaternion</th>
<th>ModifiedRodriguez Parameters</th>
<th>Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other Representations</th>
<th></th>
</tr>
</thead>
</table>
Attitude Integration

AttitudeFrame → Rotation → Quaternion

AttitudeState

AttitudeHistory

Converter → History

CubicSpline Interpolator

Runge-Kutta Integrator → Attitude Dynamics

EnvFunction
Hardware-in-the-Loop

[Diagram of Hardware-in-the-Loop system]

Flight Software

- Attitude Control
  - Torquers SensorStub
  - RateGyro SensorStub
  - Magnetometer SensorStub
  - Applied Torques
  - Angular Velocity
  - B-Field

- Orbit Determination
  - GPS SensorStub
  - Position Velocity

- Orbit Control
  - Thrusters SensorStub
  - Applied Forces

Open-SESSAME Communications Layer

- Open-SESSAME Application Server
- Propagator and History
Conclusions

- No prior solution for configurable, extensible, coupled spacecraft simulation
- Open-SESSAME fulfills goals and requirements
  - Speed and accuracy comparable to commercial packages
  - Used to ease development and testing of HokieSat flight code
- Future
  - Multiple extension points for added functionality
  - Integrate software into curriculum
  - Publish use cases and include in lab activities
  - Increase connectivity with other programs
  - Optimize operation
Thank you

- Space Systems Simulation Laboratory
- Dr. Christopher Hall
- Sourceforge.net
- ION-F & HokieSat
Questions?
Object-Oriented Programming

```
Vehicle
  +Move(double) : void
  +Stop() : void
  +Turn(double) : void

Car
  -m_Mileage : double
  -m_Speed : double
  +Move(double) : void
  +Stop() : void
  +Turn(double) : void
  +SetSpeedMPH(double) : void
  +GetSpeedMPH() : double
  +GetSpeedKPH() : double
  -SetMileage(double) : void

Bicycle

RollerSkates

Axle
  +AddWheel() : void
  +Spin() : void

Wheel
  +Roll() : void

SportsCar
  +PressClutch() : void
  +UseNitro() : void

StationWagon
  +OpenTailgate() : void
```
Section: Architecture Design

- Software Architecture
- Component Libraries
- Building an Application
Attitude Library

- AttitudeState
- Composition
- Rotation, Frame, AngVel
- Attitude
  - Larger composition
  - AttitudeState
  - Kinematics & Dynamics
  - History
Orbit Library

- **OrbitState**
  - Local Composite
- **OrbitStateRep**
- **OrbitFrame**
- **Orbit**
  - Larger Composite
  - Dynamics
  - History
Dynamics Library

- **Propagator**
  - Abstract interface
  - Types of propagators
    - Numeric
    - Analytic

```
Abstract interface

Types of propagators

- Numeric
- Analytic
```

```
Propagator

NumericPropagator

Combined-NumericPropagator

EnkeCombined-NumericPropagator

Other Numeric Propagators

AnalyticPropagator

AnalyticPropagator Schemes
```
Environment Library

- Environment
- Composition and interface
- CentralBody
  - Physical parameters
  - Defines frames
  - Location with respect to other celestial bodies
- EnvFunctions
  - Interface to force and torque functions
  - Parameters: time, orbit state,
Data Handling Utilities

- **History**
  - Vectors of data
  - Associated with times
  - Interpolation
- **Data Conversion**

![Diagram](image)
Math Library

- Numerical Integration
  - Fixed step
  - Variable step
- Data Interpolation
# High Eccentricity

## Parameter

<table>
<thead>
<tr>
<th>Initial Orbit State</th>
<th>( r_0 = [7500 \ 0 \ 0]^T )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_0 = [0 \ 6.819333903 \ 6.819333903]^T )</td>
<td></td>
</tr>
<tr>
<td>Disturbances</td>
<td>Two-Body Gravity</td>
</tr>
<tr>
<td>Integrator</td>
<td>Runge-Kutta-Fehlberg 4(5)</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>20 days</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output</th>
<th>O-SESSAME</th>
<th>STK</th>
<th>FreeFlyer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Orbit State</td>
<td>([-46724.94751 \ -8276.177337 \ -8276.177337 \ 1.339061791 \ -0.85741544 \ -0.85741544]^T)</td>
<td>([-46727.35881 \ -8275.048124 \ -8275.048124 \ 1.338824 \ -0.857447 \ -0.857447]^T)</td>
<td>([-46713.96349 \ -8283.620519 \ -8283.620519 \ 1.340490327 \ -0.857151194 \ -0.857151194]^T)</td>
</tr>
<tr>
<td>Running Time</td>
<td>15 sec</td>
<td>4 sec</td>
<td>48 sec</td>
</tr>
<tr>
<td>RMS Error</td>
<td>-</td>
<td>(2.72 \times 10^{-4})</td>
<td>(1.73 \times 10^{-3})</td>
</tr>
</tbody>
</table>
Accumulation of Error

High Eccentricity Error

Position Error (km)

Altitude (km)