**DOTNET Framework Design Environment for System Integration of Planetary Probe Payload Sensors and Interplanetary Trajectory Optimization**

Prabhakar Subrahmanyam, Keith Schreck and Periklis Papadopoulos

San Jose State University, Department of Mechanical and Aerospace Engineering, One Washington Square, San Jose, California, USA, 95152-0087

prasub@gmail.com, keithspace@yahoo.com and periklis.papadopoulos@sjsu.edu

---

**Software Framework Architecture**

- Interplanetary Payload Sensors and Mission Design using Multi Disciplinary Design Optimization
  - Launch vehicle performance – The Energy equation
  - Planetary orbital position – Keplarian orbit elements
  - Interplanetary trajectories – Lambert’s Problem
  - $\Delta V$ computations – Patched-Conic Approximation
  - PSO derivation – Rocket engine equation
  - Optimum case determination – MDO Simplex and Matrix Experiment techniques
  - Ability to calculate conic section between planets using the Lambert targeting technique
  - Calculation of conic section from planet center to planet center in a specified time
  - Ability to calculate the required velocity impulse at each of the planets

**Interplanetary Trajectory Design Flow as employed in .NET Framework**

**Payload Sensor and Planetary Probe Relational Database Management**

- Developed a Payload Sensors RDBMS and Integrated DB with .NET Framework for Web accessibility
- Database manager allows selective data retrieval through .NET web application user interface
- Comprehensive database of existing planetary probe designs and Sensors
  - Features tables for aerothermal, geometry, trajectory, sensors, etc., information
  - Users can modify existing vehicle designs by changing geometric features available in GUI
    - Update the database for their custom test probe architectures
    - Add/Modify sensors in the UI based on the initial selection by the system driven by Planetary body/Mission

**Framework Entry Design Environment**

- SPARTA Design environment is an Automation Package for Sensor design
  - Choose a Target Planetary body and the Sensors are automatically loaded into the DOTNET design interface
  - The total power, thermal, mass requirements and other parameters are automatically calculated from the database and populated in the user interface
  - The user is empowered with an option to add or delete a sensor for packaging into the probe
    - The TOF, $\Delta V$ are all calculated and the user is alerted.
    - For Earth (Re) Entry, user is presented with KML format for reentry visualization in Google Earth

**Driving Google Visualization from DOTNET Framework**

- Earth (Re) Entry Trajectory Data converted to Keyhole Markup Language (KML)
- Framework converts the trajectory data and produces a fresh KML file for every trajectory array set.
- Google Earth interprets the KML format and produces fly-by visualization along the entry trajectory
- Google Earth provides rich interactive and attractive platform for visualizing geospatial data
- The framework employs a pure Java library which generates KML output to display the most commonly interesting forms of geospatial data.
- No prior knowledge of KML is required for the end-user.
- They are presented with a very simple API which expects data in the form of arrays, in which it is likely to already exist, and will create graphical elements of the sort which users are most likely to want

**Driving Google Visualization from DOTNET: Stardust Trajectory**

---

**DOTNET Planetary Probe Vehicle and Trajectory Design Interface**

**Interplanetary Trajectory Optimization DOTNET Framework**

**Payload Mass Calculation: Inputs**

<table>
<thead>
<tr>
<th>Departure Planet</th>
<th>Arrival Planet</th>
<th>Mass</th>
<th>1,000 km</th>
<th>Mass</th>
<th>1,000 km</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earth</td>
<td>Mars</td>
<td>20,000 kg</td>
<td>1,000 km</td>
<td>20,000 kg</td>
<td>1,000 km</td>
</tr>
<tr>
<td>Venus</td>
<td>Mars</td>
<td>1,000 kg</td>
<td>1,000 km</td>
<td>1,000 kg</td>
<td>1,000 km</td>
</tr>
<tr>
<td>Jupiter</td>
<td>Earth</td>
<td>5,000 kg</td>
<td>1,000 km</td>
<td>5,000 kg</td>
<td>1,000 km</td>
</tr>
<tr>
<td>Saturn</td>
<td>Earth</td>
<td>2,000 kg</td>
<td>1,000 km</td>
<td>2,000 kg</td>
<td>1,000 km</td>
</tr>
</tbody>
</table>

**Payload Mass Calculation Output**

<table>
<thead>
<tr>
<th>Departure Spiral</th>
<th>Trip Time</th>
<th>Fuel Consumed</th>
<th>Arrival Spiral</th>
<th>Trip Time</th>
<th>Fuel Consumed</th>
<th>Initial Mass</th>
<th>Final Mass</th>
<th>Payload Mass</th>
<th>Equivalent $\Delta V$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.00 PM</td>
<td>75.14 days</td>
<td>275.86 kg</td>
<td>13.81 days</td>
<td>275.86 kg</td>
<td>5,000 kg</td>
<td>20,000 kg</td>
<td>20,000 kg</td>
<td>20,000 kg</td>
<td>12,090 km/s</td>
</tr>
</tbody>
</table>

**Initial Mass Calculations**

<table>
<thead>
<tr>
<th>Departure Spiral</th>
<th>Trip Time</th>
<th>Fuel Consumed</th>
<th>Arrival Spiral</th>
<th>Trip Time</th>
<th>Fuel Consumed</th>
<th>Initial Mass</th>
<th>Final Mass</th>
<th>Payload Mass</th>
<th>Equivalent $\Delta V$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.00 PM</td>
<td>75.14 days</td>
<td>275.86 kg</td>
<td>13.81 days</td>
<td>275.86 kg</td>
<td>5,000 kg</td>
<td>20,000 kg</td>
<td>20,000 kg</td>
<td>20,000 kg</td>
<td>12,090 km/s</td>
</tr>
</tbody>
</table>

---

**Planetary Probe Vehicle and Trajectory Design Interface**

**Payload Sensor and Planetary Probe Relational Database Management**

- Developed a Payload Sensors RDBMS and Integrated DB with .NET Framework for Web accessibility
- Database manager allows selective data retrieval through .NET web application user interface
- Comprehensive database of existing planetary probe designs and Sensors
  - Features tables for aerothermal, geometry, trajectory, sensors, etc., information
  - Users can modify existing vehicle designs by changing geometric features available in GUI
    - Update the database for their custom test probe architectures
    - Add/Modify sensors in the UI based on the initial selection by the system driven by Planetary body/Mission

**Framework Entry Design Environment**

- SPARTA Design environment is an Automation Package for Sensor design
  - Choose a Target Planetary body and the Sensors are automatically loaded into the DOTNET design interface
  - The total power, thermal, mass requirements and other parameters are automatically calculated from the database and populated in the user interface
  - The user is empowered with an option to add or delete a sensor for packaging into the probe
    - The TOF, $\Delta V$ are all calculated and the user is alerted.
    - For Earth (Re) Entry, user is presented with KML format for reentry visualization in Google Earth

**Driving Google Visualization from DOTNET Framework**

- Earth (Re) Entry Trajectory Data converted to Keyhole Markup Language (KML)
- Framework converts the trajectory data and produces a fresh KML file for every trajectory array set.
- Google Earth interprets the KML format and produces fly-by visualization along the entry trajectory
- Google Earth provides rich interactive and attractive platform for visualizing geospatial data
- The framework employs a pure Java library which generates KML output to display the most commonly interesting forms of geospatial data.
- No prior knowledge of KML is required for the end-user.
- They are presented with a very simple API which expects data in the form of arrays, in which it is likely to already exist, and will create graphical elements of the sort which users are most likely to want

**Driving Google Visualization from DOTNET: Stardust Trajectory**

---