Virtual Autonomous Navigation Environment

76th MORS Symposium
10-12 June 2008

Mr. Christopher Cummins
Geotechnical and Structures Laboratory
Approved for public release distribution.
### Virtual Autonomous Navigation Environment

**Geotechnical and Structures Laboratory**

VANE Research Focus

- Integrate vehicle mobility, ground physics, terrain physics and sensor response models into a High Performance Computing computational testbed to facilitate virtual testing of UMS for autonomous navigation performance
**Purpose:**
- Represent mechanical system interactions with the CTB
- Realistic movement
- Provide an interface for mechanical systems and sensor models
- Allow easy configuration of mission scenario

**Results:**
- JAUS Compliance
- Dynamics engine for VANE simulation
- Simultaneous viewing of sensor output, vehicle mobility, and ANS
- Mission rehearsal and playback

**Payoff:**
- Interaction with guest JAUS compliant subsystems.
- Faster debugging of components
- Viewing options for output data

<table>
<thead>
<tr>
<th>Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Milestones</strong></td>
</tr>
<tr>
<td>Mechanical systems</td>
</tr>
<tr>
<td>UI design and construction</td>
</tr>
<tr>
<td>CTB Environment Interface</td>
</tr>
<tr>
<td>JAUS Applications</td>
</tr>
</tbody>
</table>
Major Needs Identified by Stakeholders

Identified Needs

<table>
<thead>
<tr>
<th>Identified Needs</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interfaces and Integration with Other Models, Simulations, Systems</td>
<td>18</td>
</tr>
<tr>
<td>Terrain/Model Fidelity</td>
<td>6</td>
</tr>
<tr>
<td>Runtime Performance</td>
<td>6</td>
</tr>
<tr>
<td>Incorporation of Human in the Loop for Teleoperation/Enhanced Teleoperation</td>
<td>3</td>
</tr>
<tr>
<td>Graphics/Scene Quality</td>
<td>1</td>
</tr>
</tbody>
</table>
Objective: Provide a high-fidelity, high-resolution environment for assessment of UGV systems and subsystems across concepts, designs, and operations to achieve implementation of the best systems.
Common Open Architecture

• Open Architecture Characteristics:
  ➢ Based on Open, Publicly Available Specifications — Preferably Maintained as Standards by a Consensus Process, e.g. By an Internationally Recognized Governing Group
  ➢ Well-defined, Widely Used Non-Proprietary (Standard) Interfaces, Services and Formats
  ➢ Durable (Stable or Slowly Evolving) Component Interfaces That Facilitate Component Replacement and Addition of New Capabilities
  ➢ Upgradeable Through Incorporation of Additional or More Capable Components With Minimal Impact on the System
Vehicle/Object Modeling

- Objects exist in several contexts
  - Collision geometries
  - Joint constraints
  - GUI visualization
  - Sensor detection

- Every object needs to make sense in each of these contexts.
- The testbed manages the objects so that the contexts can be resolved to one entity
- Some object data can be manipulated graphically though the testbed.
GUI Design

• OGRE rendering engine
  ➢ Object Oriented Graphics Rendering Engine
  ➢ Open source
  ➢ Object Oriented for manageability
  ➢ Includes shadow, shaders, object and scene loading, and other functions to reduce programming time.
  ➢ Portable
  ➢ DirectX/OpenGL

• WXWidgets interface
  ➢ Free software license
  ➢ Powerful
  ➢ Easy to program
  ➢ Multiple windows
  ➢ Portable
  ➢ Uses the native GUI to reduce the learning curve
Ridged Body Dynamics Simulation

- ODE (Open Dynamics Engine)
  - Open Source
  - Mature
  - Widely used
  - Fast or Accurate
  - Portable
- Different levels of accuracy are possible
  - Allows for different uses for the VANE test bed depending on the mission.
- The option exists if replacement of ODE with other physics solutions such as PhysX is desired.
- ODE has an active user base to help in solving programming issues and to ensure continuing updates.

Robot picking an object
Actuator Modeling

- **Actuator joints and motors**
  - Linear Actuators
  - Rotational Servos
  - Lever Arms
  - Motors/Engines

- **The joints use either a force or a speed solver.**
  - The speed solution is used on joints that are not torque limited. This allows for easy and accurate modeling of speed limited joints.

- **Lever arm linkages are modeled internally to the actuator to decrease degrees of freedom in the physics solver.**
The terrain is represented by a series of nodes.

The forces on the wheel are computed using the continuous spring tire model.

The traction element sinkage is determined and used to calculate the sinkage at the current time step that applies to each terrain node in contact with the traction element.

\[
S_c = \frac{(V \times T)}{C} \times S
\]

- $S$ = Predicted total sinkage (in) for entire wheel
- $C$ = Chord Length (in) from P1 to P2
- $T$ = Time step (sec)
- $V$ = Vehicle’s instantaneous velocity (in/sec)
- $S_c$ = Sinkage (in) this time step
Terrain Generation

• Conversion to formats appropriate to dynamics and visualization
  - High resolution for dynamics modeling
  - Low resolution for GUI display

• The Terrain formats allow for paging of large data sets.

• The formats also allow for easy deformation of the terrain by the soil sinkage models.
Operating over scanned terrain
Path Forward

- Operator Control Unit (OCU) construction
- Interaction between sensor models and vehicle simulation
- Easier integration of terrain into dynamics engine
- Importing of larger data sets
Path Forward

Testbed / CTB interaction

Vehicle Simulation

Contexts

Modules

Adaptive Scene Generation

LIDAR

Physics

GUI

Tire/Track Model
Path Forward

Testbed / CTB interaction

Sensor Motion

Sensor Position/Orientation

Sensor Output

ANS

CTB

Sensor Simulation
• Demonstration of the VANE Dynamics