INVESTIGATING THE MOBILITY OF LIGHT AUTONOMOUS TRACKED VEHICLES USING A HIGH PERFORMANCE COMPUTING SIMULATION CAPABILITY

Prof. Dan Negrut
Hammad Mazhar
Daniel J. Melanz
University of Wisconsin-Madison
Madison, WI

Dr. David Lamb
Dr. Paramsothy Jayakumar
Michael Letherwood
US Army TARDEC
Warren, MI

Dr. Abhinandan Jain
Dr. Marco Quadrelli
Jet Propulsion Lab
California Institute of Technology
Pasadena, CA

DISCLAIMER: Reference herein to any specific commercial company, product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the Department of the Army (DoA). The opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or the DoA, and shall not be used for advertising or product endorsement purposes.
Investing the Mobility of Light Autonomous Tracked Vehicles using a High Performance Computing Simulation Capability

David Lamb; Paramsothy Jayakumar; Michael Letherwood; Dan Negut; Abhinandan Jain

U.S. Army TARDEC, 6501 East Eleven Mile Rd, Warren, MI, 48397-5000

Approved for public release; distribution unlimited

Submitted to 2012 NDIA Ground Vehicle Systems Engineering and Technology Symposium August 14-16 Troy, Michigan

briefing charts.
Acknowledgements

• Collaborators:
  – Alessandro Tasora – University of Parma, Italy
  – Mihai Anitescu – Argonne National Lab, USA

  – Lab Students:
    ● Aaron Bartholomew
    ● Makarand Datar
    ● Toby Heyn
    ● Naresh Khude
    ● Justin Madsen

  ● Hammad Mazhar
  ● Dan Melanz
  ● Spencer O’Rourke
  ● Arman Pazouki
  ● Andrew Seidl
  ● Rebecca Shotwell

• Financial support
  – National Science Foundation, Career Award
  – Army Research Office (ARO)
  – US Army TARDEC
  – FunctionBay, S. Korea
  – NVIDIA
  – Caterpillar
  – MSC.Software
  – Advanced Micro Devices (AMD)
Classical Computational Dynamics, Constrained Equations of Motion

Generalized Positions

Kinematic Differential Equations

Force Balance Equations

Holonomic Kinematic Constraints

Generalized Mass Matrix

Velocity Transformation Matrix

Generalized Velocities

\( \dot{q} = L(q)v \)

\( M(q)\dot{v} = f(t, q, v) - g_q^T(q, t)\lambda \)

\( g(q, t) = 0 \)

Reaction Force

Applied Force
**Purpose**: understand/optimize performance before building prototype
Multibody Dynamics: Is anything left to do?
All the good music has already been written by people with wigs and stuff.

Frank Zappa
Frictional Contact Simulation
[Commercial Solution]

• Model Parameters:
  – Spheres: 60 mm diameter and mass 0.882 kg
  – Forces: smoothing with stiffness of 1E5, force exponent of 2.2, damping coefficient of 10.0, and a penetration depth of 0.1
  – Simulation length: 3 seconds
CAE: Looking Ahead…

- How is the Rover moving along on a slope with granular material?
- What wheel geometry is more effective?
Applications transitioning from multi-body to many-body dynamics

Bodies interacting through friction/contact/impact

Bodies are compliant, sometimes undergo large deformations

Bodies might interact with fluid (FSI)

Tomorrow’s problems are in the realm of multi-physics
Simulating large engineering problems remains a challenge...
Lab’s Research Heterogeneous Computing Cluster

Legend, Connection Type:
- Gigabit Ethernet
- 4x QDR Infiniband

File Server Architecture
- CPU: Intel Xeon 5620
- RAM: 16 GB DDR3
- Infiniband HCA
- RAID 6
- 24x 2TB Hard Disks

Remote Collaborators

Internal Users

Gigabit Ethernet Switch

Head Node

4x QDR Infiniband Switch

File Server

CPU/GPU Node 1

CPU/GPU Node 2

CPU/GPU Node 14

AMD Node 1

CPU/GPU Node Architecture
- CPU 0: Intel Xeon 5520
- CPU 1: Intel Xeon 5520
- RAM: 48 GB DDR3
- Infiniband HCA
- 1.5GB RAM
- 448 Cores PCIEx16 2.0

AMD Node Architecture
- CPU 0: AMD Opteron 6276
- CPU 1: AMD Opteron 6276
- CPU 2: AMD Opteron 6276
- CPU 3: AMD Opteron 6276
- RAM: 128 GB DDR3
- Infiniband HCA
- SSD
Lab’s Research Heterogeneous Computing Cluster

- More than 25,000 GPU scalar processors
  - Can manage about 75,000 GPU parallel threads at full capacity
- More than 1000 CPU cores
- Mellanox Infiniband Interconnect, 40Gb/sec
- About 0.7 TB of RAM
- More than 20 Tflops DP
- ...

The issues is not hardware availability. Rather, it is producing modeling and solution techniques that can leverage this hardware
Goal, lab’s research effort: shape up the future of physics-based simulation
- Develop a Heterogeneous Computing Template (HCT) that leverages emerging hardware architectures and suitable algorithms to solve open engineering problems

Targeted “emerging hardware architectures”:
- Clusters of CPUs and GPUs (accelerators)
  - More than 100 CPU cores, tens of GPU cards, tens of thousands of GPU cores

Focus on “open engineering problems”
- Vehicle mobility, granular dynamics, soil modeling, tire/terrain modeling, FSI, etc.
HCT: Five Major Components

• Computational Dynamics requires
  – Advanced modeling techniques
  – Strong algorithmic (applied math) support
  – Proximity computation
  – Domain decomposition & Inter-domain data exchange
  – Post-processing (visualization)

• HCT represents the library support, the associated API, and the embedded tools that support this five component abstraction
Multi-Physics targeted Computational Dynamics requires

- **Advanced modeling techniques**
  - Strong algorithmic (applied math) support
  - Proximity computation
  - Domain decomposition & Inter-domain data exchange
  - Post-processing (visualization)
• Modeling: what does it mean?
  – The process of formulating a set of governing differential equations that captures the multi-physics associated with the engineering problem of interest

• Modeling decisions are consequential
  – Good modeling places you at an advantage when it comes to simulating hard problems
Multi-Body Dynamics w/ DVI

Generalized Positions

\[ \mathbf{q} = \mathbf{L}(\mathbf{q}) \mathbf{v} \]

Velocity Transformation Matrix

\[ \dot{\mathbf{q}} = \mathbf{L}(\mathbf{q}) \mathbf{v} \]

Generalized Mass Matrix

\[ \mathbf{M}(\mathbf{q}) \dot{\mathbf{v}} = \mathbf{f}(t, \mathbf{q}, \mathbf{v}) - \mathbf{g}_q(\mathbf{q}, t) \lambda + \sum_{i=1}^{N_c} (\gamma_n^i \mathbf{D}_n^T \dot{\mathbf{v}} + \gamma_u^i \mathbf{D}_u^T \dot{\mathbf{v}} + \gamma_w^i \mathbf{D}_w^T \dot{\mathbf{v}}) \]

Generalized Velocities

Frictional Contact Force

Force Balance Equations

\[ \mathbf{g}(\mathbf{q}, t) = \mathbf{0} \]

Kinematic Differential Equations

Reaction Force

Holonomic Kinematic Constraints

Applied Force

Contact Impulse, for Contact “i”

\[ 0 \leq \Phi^i(\mathbf{q}, t) \perp \gamma_n^i \geq 0 \quad i = 1, 2, \ldots, N_c \]

Coulomb Friction Model

Total Number of Contacts

Friction Impulse Components, for Contact “i”

Gap Function, for Contact “i”

Friction Dissipation Energy Gap Function, for Contact “i”

\[ \langle \gamma_u^i, \gamma_w^i \rangle = \arg \min_{\mu \gamma_n^i \geq (\gamma_u^i)^2 + (\gamma_w^i)^2} (\gamma_u^i \mathbf{v}^T \mathbf{D}_u^i + \gamma_w^i \mathbf{v}^T \mathbf{D}_w^i) \]
Traditional Discretization Scheme

\[
\begin{align*}
q^{(l+1)} &= q^{(l)} + hL(q^{(l)})v^{(l+1)} \\
M(v^{(l+1)} - v^l) &= hf(t^{(l)}, q^{(l)}, v^{(l)}) + \sum_{i \in A(q^{(l)}, \delta)} \gamma_i n D_{i,n} + \gamma_i u D_{i,u} + \gamma_i w D_{i,w}
\end{align*}
\]

\[
i \in A(q^{(l)}, \delta) : 0 \leq \frac{1}{h} \Phi_i(q^{(l)}) + D_{i,n}^T v^{(l+1)} \perp \gamma_n^i \geq 0,
\]

\[
(\gamma_i u, \gamma_i w) = \argmin_{\mu_i \gamma_i n \geq \sqrt{\gamma_{i,u}^2 + \gamma_{i,w}^2}} v^T (\gamma_i u D_{i,u} + \gamma_i w D_{i,w}).
\]
The Cone Complementarity Problem (CCP)

- First order optimality conditions lead to Cone Complementarity Problem

- Introduce the convex hypercone...

\[ \mathcal{Y} = \left( \bigoplus_{i \in \mathcal{A}(q^{l}, \epsilon)} \mathcal{F}C^{i} \right) \]

\( \mathcal{F}C^{i} \in \mathbb{R}^{3} \) represents friction cone associated with \( i^{th} \) contact

... and its polar hypercone:

\[ \mathcal{Y}^{\circ} = \left( \bigoplus_{i \in \mathcal{A}(q^{l}, \epsilon)} \mathcal{F}C^{i^{\circ}} \right) \]

CCP assumes following form: Find \( \gamma \) such that

\[ \gamma \in \mathcal{Y} \perp - (N\gamma + d) \in \mathcal{Y}^{\circ} \]
The Quadratic Programming Angle…

- The relaxed EOM represent a cone-complementarity problem (CCP)

- The CCP captures the first-order optimality condition for a quadratic optimization problem with conic constraints:

\[
\begin{aligned}
\min q(\gamma) &= \frac{1}{2} \gamma^T N \gamma + d^T \gamma \\
\text{subject to} \quad \gamma_i \in \mathcal{Y}_i \quad \text{for} \quad i = 1, 2, \ldots, N_c
\end{aligned}
\]

- Notation used:

\[
\gamma \equiv [\gamma_1^T, \gamma_2^T, \ldots, \gamma_{N_c}^T]^T \in \mathbb{R}^{3 \times N_c} \quad \text{and} \quad \mathcal{Y}_i : (\gamma_{u,i}^2 + \gamma_{w,i}^2) - \mu_i \gamma_{n,i}^2 \leq 0
\]
1. For each contact $i$, evaluate $\eta_i = 3/\text{Trace}(D_i^T M^{-1} D_i)$.

2. If some initial guess $\gamma^*$ is available for multipliers, then set $\gamma^0 = \gamma^*$, otherwise $\gamma^0 = \mathbf{0}$.

3. Initialize velocities: $\mathbf{v}^0 = \sum_i M^{-1} D_i \gamma_i^0 + M^{-1} \mathbf{k}$.

4. For each contact $i$, compute changes in multipliers for contact constraints:
   \[
   \gamma_i^{r+1} = \lambda \Pi_{\gamma_i} (\gamma_i^r - \omega \eta_i (D_i^T \mathbf{v}^r + b_i)) + (1 - \lambda) \gamma_i^r ;
   \]
   \[
   \Delta \gamma_i^{r+1} = \gamma_i^{r+1} - \gamma_i^r ;
   \]
   \[
   \Delta \mathbf{v}_i = M^{-1} D_i \Delta \gamma_i^{r+1}.
   \]

5. Apply updates to the velocity vector:
   \[
   \mathbf{v}^{r+1} = \mathbf{v}^r + \sum_i \Delta \mathbf{v}_i
   \]

6. $r := r + 1$. Repeat from 4 until convergence or $r > r_{\text{max}}$. 

Mixing 50,000 M&Ms on the GPU
Multi-Physics targeted Computational Dynamics requires

- Advanced modeling techniques
- **Strong algorithmic (applied math) support**
- Proximity computation
- Domain decomposition & Inter-domain data exchange
- Post-processing (visualization)
1 Million Rigid Spheres [parallel on the GPU]
## Objective Function Value

[1K bodies, 3525 contacts]

<table>
<thead>
<tr>
<th>Method</th>
<th>Iterations</th>
<th>Final Objective Function Value</th>
<th>$\gamma_{\text{min}}$</th>
<th>$\gamma_{\text{max}}$</th>
<th>Computation Time [sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPMINRES-no p</td>
<td>1000 MinRes Its. [within 100 changes of active set]</td>
<td>-2.9035</td>
<td>0.0</td>
<td>7.7487</td>
<td>6.7002</td>
</tr>
<tr>
<td>GPMINRES-no p (not plotted above)</td>
<td>10000 MinRes Its. [within 1000 changes of active set]</td>
<td>-2.9045</td>
<td>0.0</td>
<td>8.2002</td>
<td>61.0698</td>
</tr>
<tr>
<td>GPMINRES-p</td>
<td>100 MinRes Its. [within 100 changes of active set]</td>
<td>-2.8854</td>
<td>0.0</td>
<td>6.8551</td>
<td>1675</td>
</tr>
<tr>
<td>Jacobi</td>
<td>1000</td>
<td>-2.5077</td>
<td>0.0</td>
<td>4.4961</td>
<td>3.6643</td>
</tr>
</tbody>
</table>

The red line has 1000 dots on it; i.e., 1000 Jacobi sweeps

The green & blue lines have 100 dots on them; i.e., 100 changes of active set
• Multi-Physics targeted Computational Dynamics requires
  – Advanced modeling techniques
  – Strong algorithmic (applied math) support
  – **Proximity computation**
  – Domain decomposition & Inter-domain data exchange
  – Post-processing (visualization)
600,000 Bodies Moving & Colliding
[on the GPU]
Example: Ellipsoid-Ellipsoid CD

\[ d = P_1 - P_2 = \left( \frac{1}{2\lambda_1} M_1 + \frac{1}{2\lambda_2} M_2 \right) c + (b_1 - b_2) \]

\[ \frac{\partial d}{\partial \alpha_i} = \frac{\partial P_1}{\partial \alpha_i} - \frac{\partial P_2}{\partial \alpha_i}, \quad \frac{\partial^2 d}{\partial \alpha_i \partial \alpha_j} = \frac{\partial^2 P_1}{\partial \alpha_i \partial \alpha_j} - \frac{\partial^2 P_2}{\partial \alpha_i \partial \alpha_j} \]

\[ \frac{\partial P}{\partial \alpha_i} = \left( \frac{1}{2\lambda} \frac{1}{8\lambda^3} M - \frac{1}{8\lambda^3} M c c^T M \right) \frac{\partial c}{\partial \alpha_i} \]

\[ \frac{\partial^2 P}{\partial \alpha_i \partial \alpha_j} = \left( -\frac{1}{8\lambda^3} M + \frac{3}{32\lambda^5} M c c^T M c^T M \right) \frac{\partial c}{\partial \alpha_i} \frac{\partial c}{\partial \alpha_j} \]

\[ -\frac{1}{8\lambda^3} \left[ (c^T M \frac{\partial c}{\partial \alpha_i}) M + M c (-\frac{\partial c}{\partial \alpha_i})^T M \right] \frac{\partial c}{\partial \alpha_j} \]

\[ +\left( \frac{1}{2\lambda} \frac{1}{8\lambda^3} M c c^T M \right) \frac{\partial^2 c}{\partial \alpha_i \partial \alpha_j} \]

\[ \varepsilon : \frac{x^2}{r_1^2} + \frac{y^2}{r_2^2} + \frac{z^2}{r_3^2} = 1 \]

\[ A : \text{Rotation Matrix} \]

\[ M = A R^2 A^T \]

\[ R = \text{diag}(r_1, r_2, r_3) \]

\[ b : \text{Translation of ellipsoids center} \]

\[ \lambda^2 = \frac{1}{4} n^T M n \]

\[ \min_{\alpha_1, \alpha_2} \| d(\alpha_1, \alpha_2) \|^2 \]
Collision Detection

- **Broad phase**
  - Draws on an Axis Aligned Bounding Box (AABB) approach

- **Narrow phase**
  - Draws on Minkowski Portal Refinement
Assembled Quad GPU Machine

Processor: AMD Phenom II X4 940 Black
Memory: 16GB DDR2
Graphics: 4x NVIDIA Tesla C1060
Power supply 1: 1000W
Power supply 2: 750W
Software/Hardware Setup

Main Data Set

Results

Thread 0
Thread 1
Thread 2
Thread 3

GPU 0
GPU 1
GPU 2
GPU 3

Open MP
CUDA

16 GB RAM
Quad Core AMD Microprocessor
Tesla C1060
4x4 GB Memory
4x30720 threads

Open MP
CUDA

16 GB RAM
Quad Core AMD Microprocessor
Tesla C1060
4x4 GB Memory
4x30720 threads
Spheres – Contacts vs. Time

Quad Tesla C1060 Configuration

Time (Sec)

Contacts (Billions)
Speedup - GPU vs. CPU (Bullet library) [results reported are for spheres]

GPU: NVIDIA Tesla C1060
CPU: AMD Phenom II Black X4 940 (3.0 GHz)
Multi-Physics targeted Computational Dynamics requires

- Advanced modeling techniques
- Strong algorithmic (applied math) support
- Proximity computation
- **Domain decomposition & Inter-domain data exchange**
- Post-processing (visualization)
\[ h = 0.001 \text{ [s]} \]
\[ g = -9.80665 \text{ [m/s}^2\text{]} \]

20k spheres
\[ r = 3.5 \text{ mm} \]
\[ \mu = 0.46 \]
\[ \omega = \pi \text{ [rad/sec]} \]

Anchor width = 5 [cm]
200,000 Bodies & 10 kg Anchor
Anchor Penetration Depth, Function of Applied Torque

Anchor Depth vs Time

- 600 N-m
- 800 N-m
- 1000 N-m
- 1200 N-m
Depth as a Function of Pulling Force

Anchor Depth vs Time

- 600 N
- 800 N
- 1000 N
- 1600 N
- 2000 N
Depth as a Function of Pulling Force
Track Simulation

Parameters:

- Driving speed: 1.0 rad/sec
- Length: 12 seconds
- Time step: 0.005 sec
- Computation time: 18.5 hours
- Particle radius: .027273 m
- Terrain: 284,715 particles
- Inertia parameters of track are fake
Dual Track ‘Footprint’
In theory, there is no difference between theory and practice. In practice, there is.

Yogi Bera
M113 Tank Simulation
Real Masses for Both Obstacles and Terrain…
Vehicle-Track-Terrain Interaction

![Force Vs Time Graph](image)

- Force (N)
- Time (sec)

- Force peaks at various time intervals, indicating dynamic interaction forces between the vehicle, track, and terrain.
Force vs Time

Revolute Joint 1

Time (sec)

Force (N)
Vehicle-Track-Terrain Interaction

![Force vs Time Graph](image)

- Revolute Joint 1
- Revolute Joint 2
- Revolute Joint 3
- Revolute Joint 4
- Revolute Joint 5
Conclusions/Putting Things in Perspective

• Goal: investigate how computing can catalyze over the next 10 years advances in Science and innovation in Engineering

• Reaching the goal…
  – Develop an experimentally validated Heterogeneous Computing Template (HCT)
  – Use HCT to advance state of the art in physics-based simulation
Details reported in…


Thank You.

negrut@wisc.edu
http://sbel.wisc.edu
University of Wisconsin-Madison
Simulation-Based Engineering Lab
Wisconsin Applied Computing Center

More Animations at:
http://sbel.wisc.edu/Animations/