**GRiD: GPU-Accelerated Rigid Body Dynamics with Analytical Gradients**

**Brian Plancher**, **Sabrina M. Neuman**, **Radhika Ghosal**, **Scott Kuindersma**, **Vijay Janapa Reddi**

1. Harvard University John A. Paulson School of Engineering and Applied Sciences, 2: Boston Dynamics

---

**The Big Picture:**

- **GRiD** is a header-only, modular, open-source, GPU-accelerated library for rigid body dynamics with analytical gradients.
  - Key features include:
    - URDF parsing & code generation to deliver optimized dynamics kernels that expose GPU-friendly computational patterns
    - Delivers end-to-end computational speedups through algorithmic refactoring
    - Modular, open-source, and header-only

---

**Design Optimizations:**

We re-factored algorithms to better leverage the GPU’s strengths by:

- Exposing more natural parallelism (especially across branches)
- Reducing work done in serial loops
- Leveraging topology driven sparsity patterns in matrices
- Unifying computational operations

---

**Performance Results:**

- Benchmarked against Pinocchio, a state-of-the-art CPU library
  - Pinocchio supports optimized CPU code generation of rigid body dynamics & analytical gradients
- GRiD scales well to complex robots and multiple computations
  - As much as a 7.2x computational speedup over the CPU
  - As much as a 2.5x speedup when accounting for I/O overhead

---

**Algorithm 1 VRNEA-F:**

\[
\begin{align*}
\text{for frame } i = 1 : N, & \text{ do } \\
\varphi_i &= \lambda \times \varphi_{i-1} + \varphi_i \\
\omega_i &= \lambda \times \omega_{i-1} + \omega_i \\
\end{align*}
\]

**Algorithm 2 VRNEA-F-GRiD:**

\[
\begin{align*}
\text{for frame } i = 1 : N, & \text{ do } \\
\alpha_i &= \lambda \times \alpha_{i-1} + \alpha_i \\
\beta_i &= \lambda \times \beta_{i-1} + \beta_i \\
\gamma_i &= \lambda \times \gamma_{i-1} + \gamma_i \\
\end{align*}
\]

---

**github.com/robot-acceleration/GRiD**

- GRiD is a header-only, modular, open-source, GPU-accelerated library for rigid body dynamics with analytical gradients.
- Key features include:
  - URDF parsing & code generation to deliver optimized dynamics kernels that expose GPU-friendly computational patterns
  - E.g., Leverages both fine-grained parallelism within each computation & coarse-grained parallelism between computations
  - Delivers end-to-end computational speedups through algorithmic refactoring
  - Modular, open-source, and header-only

---

**GRiD currently supports:**

- Prismatic, fixed, and revolute joints
- ID, FD, M⁻¹
- ∇ID, ∇FD with respect to q, q, u

---

This material is based upon work supported by the National Science Foundation (under Grant DGE1745303 and Grant 2008909). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and may not reflect those of the funding organizations.