Accelerating Robot Dynamics Gradients on a CPU, GPU, and FPGA

Place Zoom Headshot Here

Brian Plancher[1], Sabrina M. Neuman[1], Thomas Bourgeat[2], Scott Kuindersma[1,3], Srini Devadas[2], Vijay Janapa Reddi[1]

1: Harvard University John A. Paulson School of Engineering and Applied Sciences,
2: MIT Computer Science and Artificial Intelligence Laboratory, 3: Boston Dynamics

Accelerating Robot Dynamics Gradients on a CPU, GPU, and FPGA

Place Zoom Headshot Here

Refactoring and **partitioning** the gradient of rigid body dynamics to expose different **hardware-compatible features** for GPUs and FPGAs provides as much as a **3.0x end-to-end speedup**

Accelerating Robot Dynamics Gradients on a CPU, GPU, and FPGA

Place Zoom Headshot Here

Refactoring and **partitioning** the gradient of rigid body dynamics to expose different **hardware-compatible features** for GPUs and FPGAs provides as much as a **3.0x end-to-end speedup**

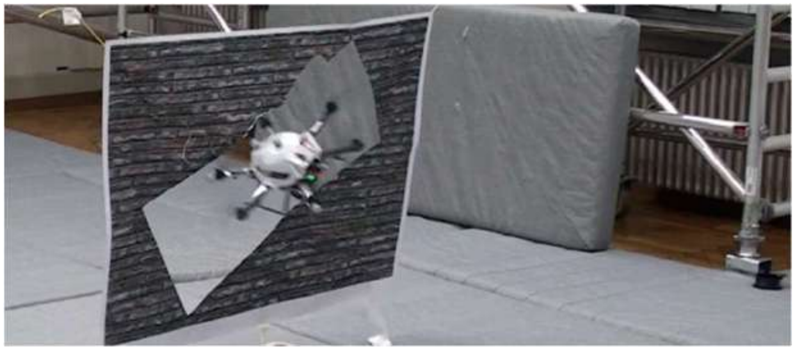
Hardware-Software Co-Design for Parallelism

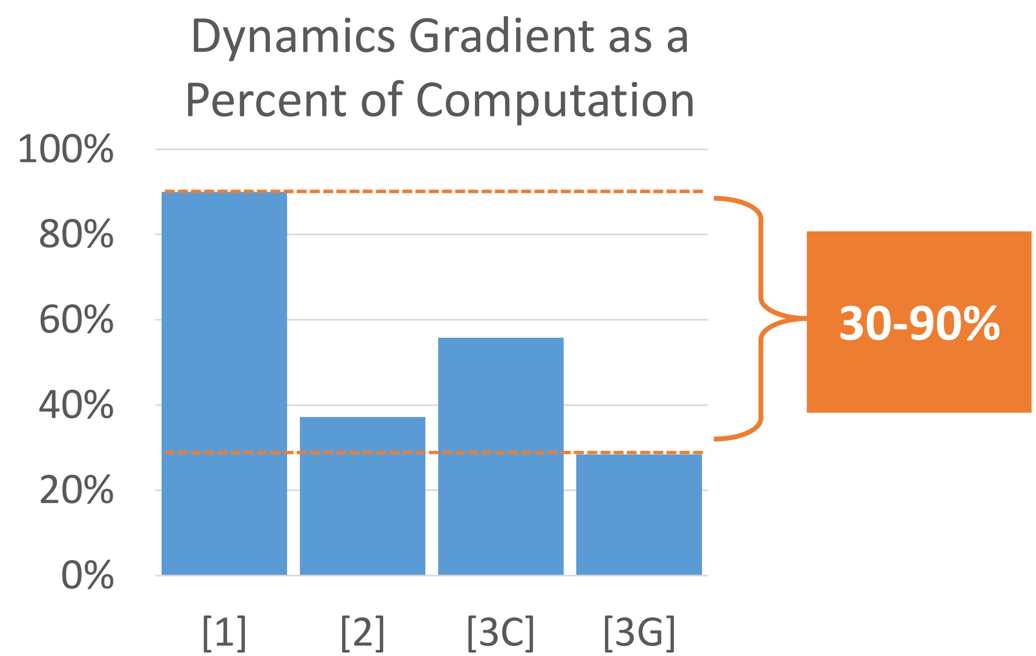
Accelerating Robot Dynamics Gradients on a CPU, GPU, and FPGA

Place Zoom Headshot Here

Rigid Body Dynamics Gradients are a bottleneck for planning and control (e.g., nonlinear MPC)

Place Zoom Headshot Here





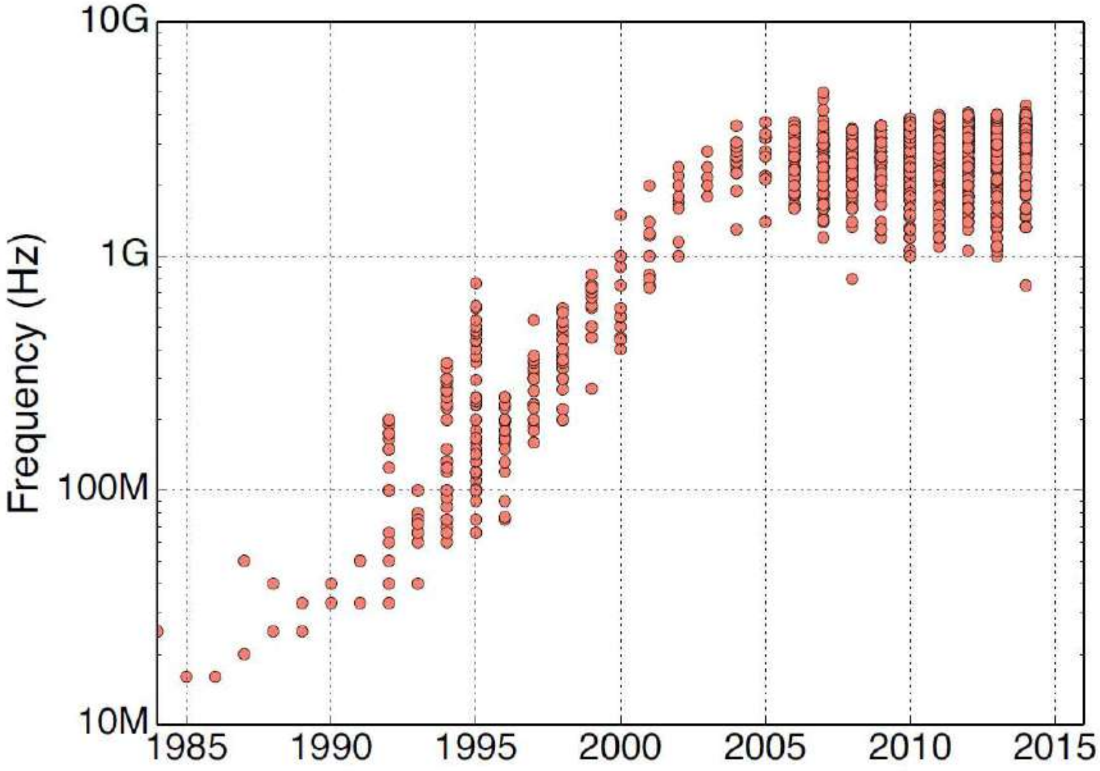
[1] J. Carpentier and N. Mansrud, "Analytical Derivatives of Rigid Body Dynamics Algorithms," RSS 2018

[2] M. Neunert, et al., "Fast nonlinear Model Predictive Control for unified trajectory optimization and tracking," ICRA 2016

[3] Best end-to-end [C]PU and [G]PU option from B. Plancher and S. Kuindersma, "A Performance Analysis of Parallel Differential Dynamic Programming," WAFR 2018

Rigid Body Dynamics Gradients are a bottleneck for planning and control (e.g., nonlinear MPC)

Place Zoom Headshot Here



- Frequency scaling is ending (CPUs aren't getting faster)
- Massive parallelism on GPUs and FPGAs may be a solution for hardware acceleration

[Shao and Brooks "Synthesis Lectures on Computer Architecture" 2015]

Accelerating Robot Dynamics Gradients on a CPU, GPU, and FPGA

Place Zoom Headshot Here

1. Motivation

2. CPUs, GPUs, and FPGAs

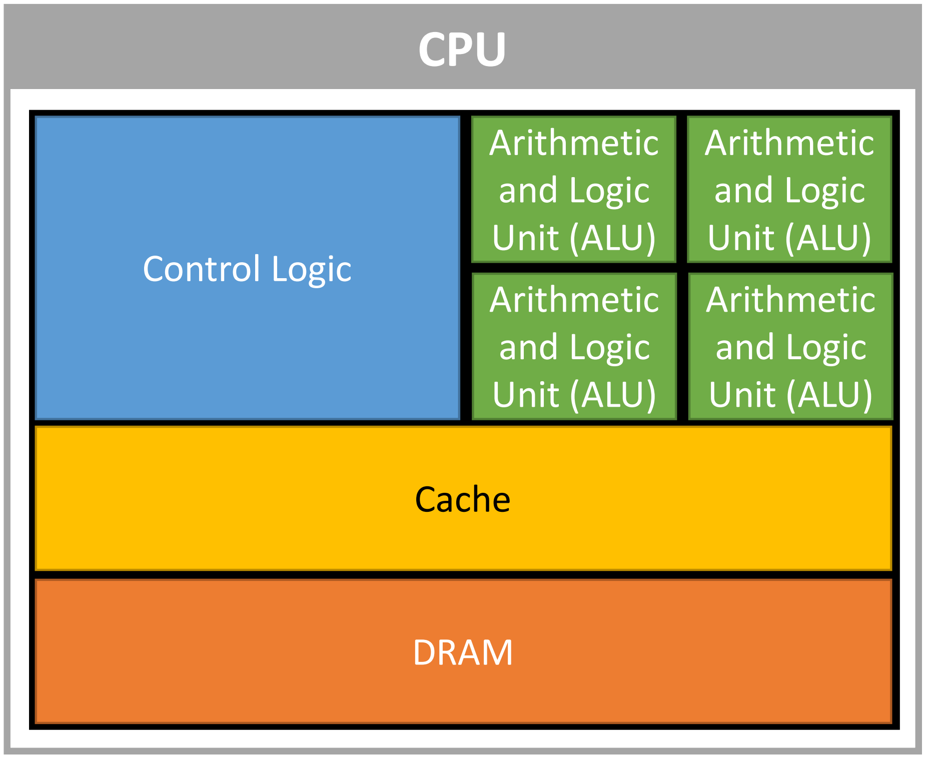
3. The Gradient of Rigid Body Dynamics

4. Accelerated Design

5. Results

CPUs, GPUs, and FPGAs have fundamentally different strengths and weaknesses

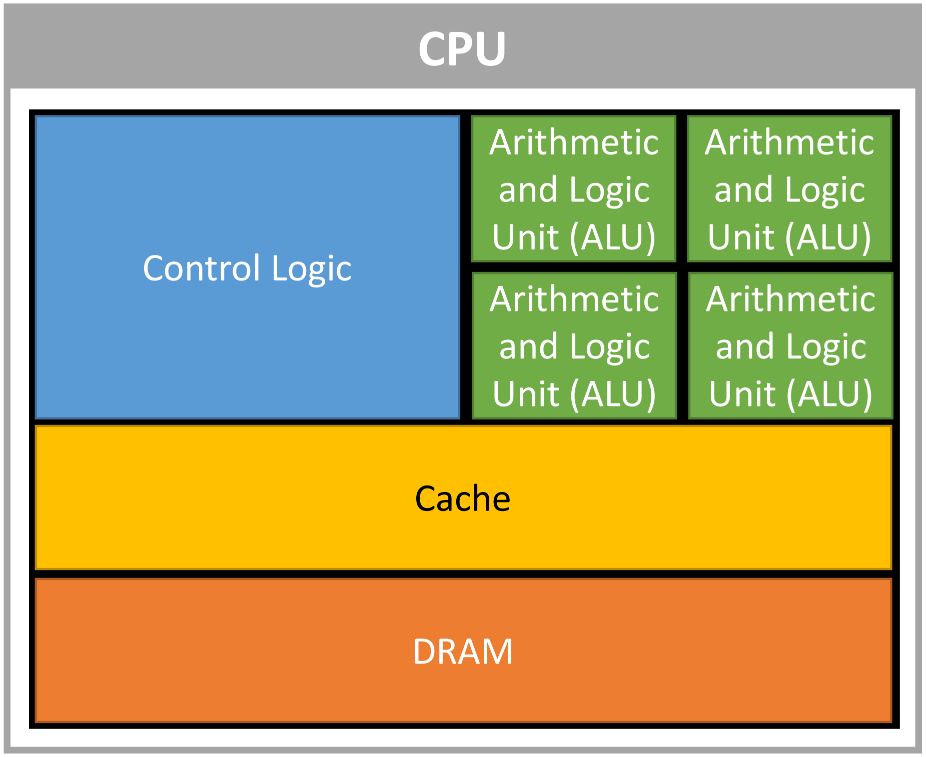
Place Zoom Headshot Here

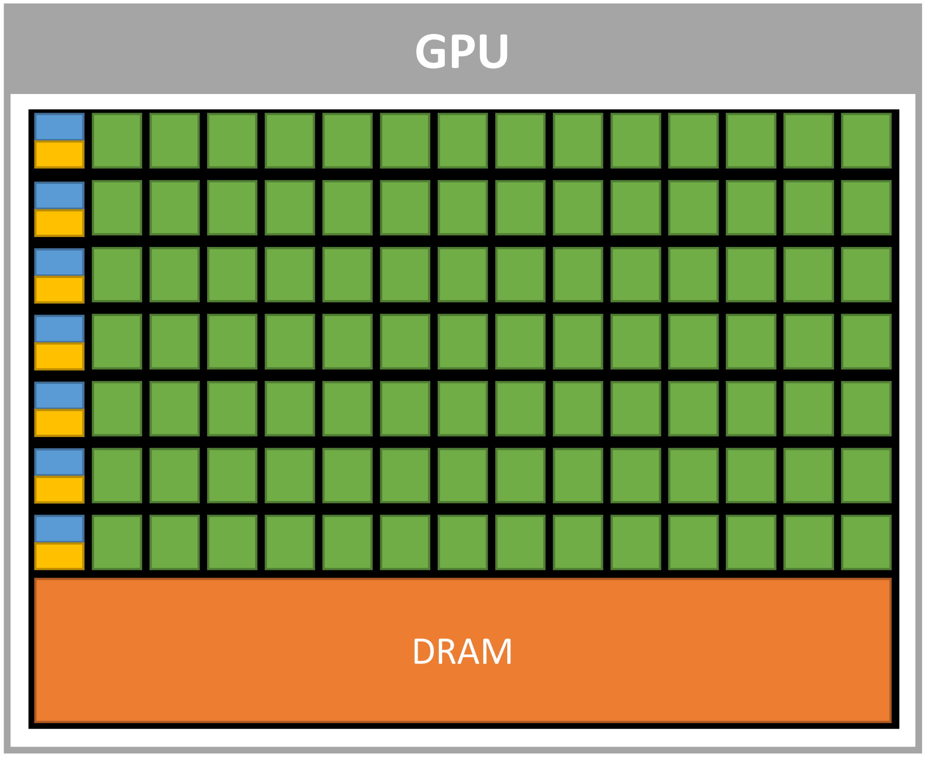


[NVIDIA]

CPUs, GPUs, and FPGAs have fundamentally different strengths and weaknesses

Place Zoom Headshot Here

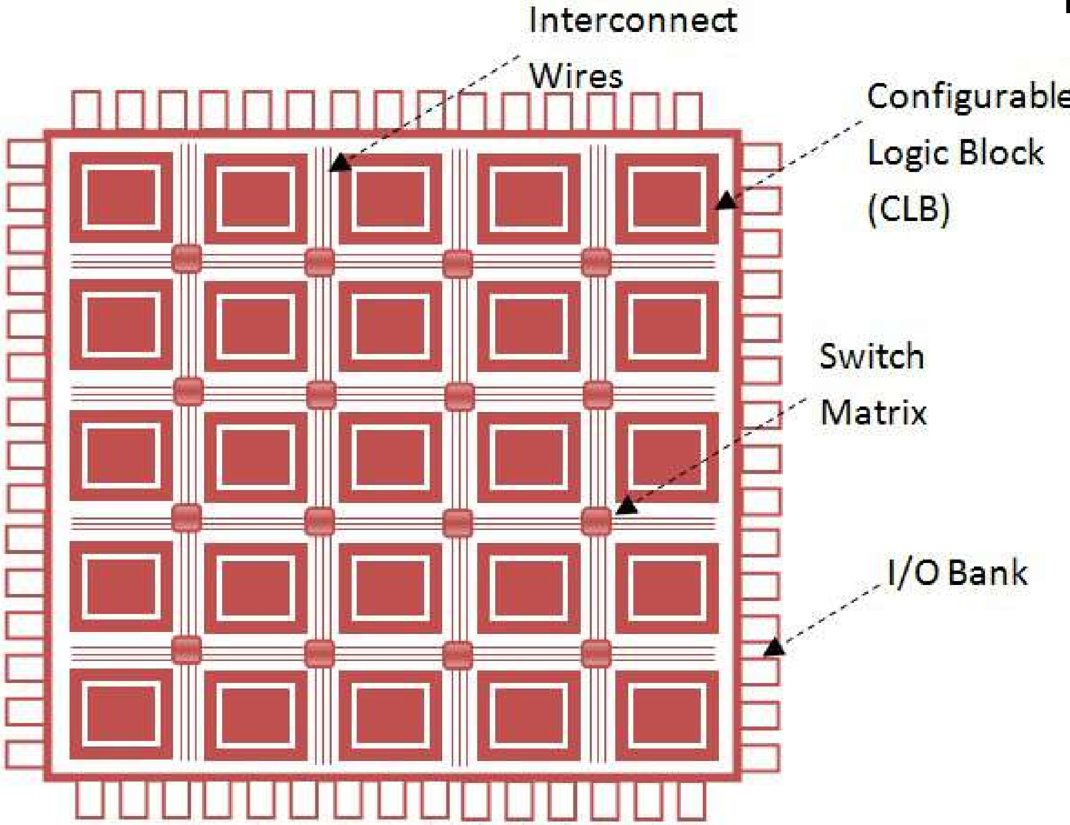




[NVIDIA]

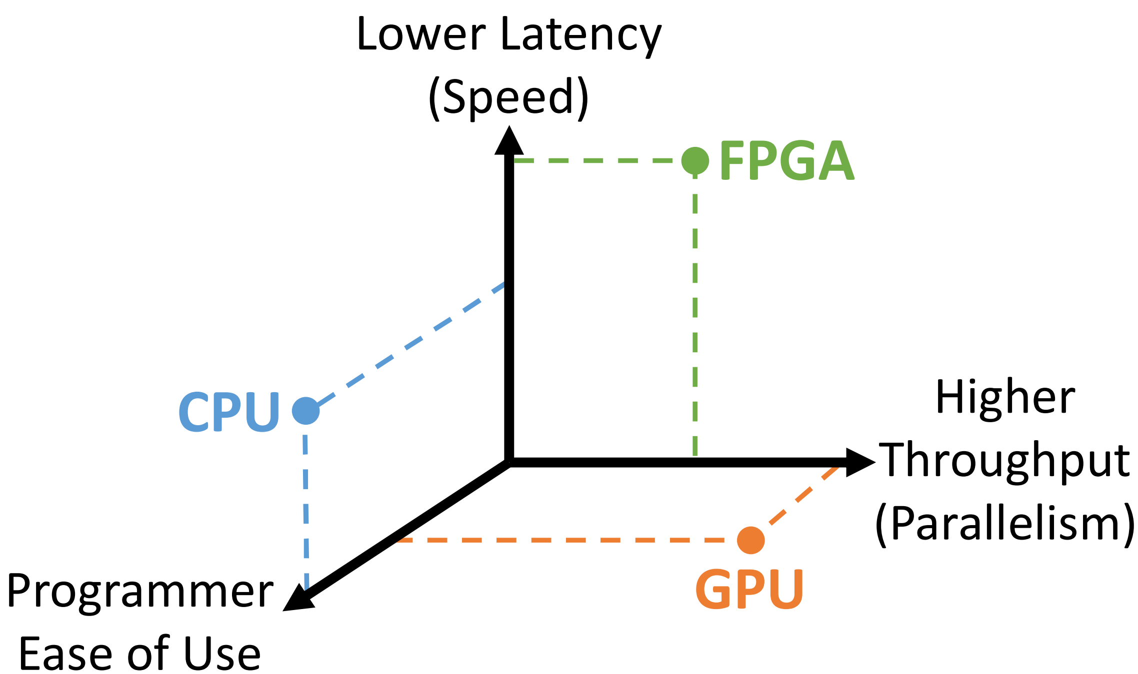
CPUs, GPUs, and FPGAs have fundamentally different strengths and weaknesses

Place Zoom Headshot Here



CPUs, GPUs, and FPGAs have fundamentally different strengths and weaknesses

Place Zoom Headshot Here



Hardware-Software Co-Design

High performance code needs to be **refactored** to take advantage of **different hardware** computational strengths and weaknesses

Accelerating Robot Dynamics Gradients on a CPU, GPU, and FPGA

Place Zoom Headshot Here

The Gradient of Rigid Body Dynamics

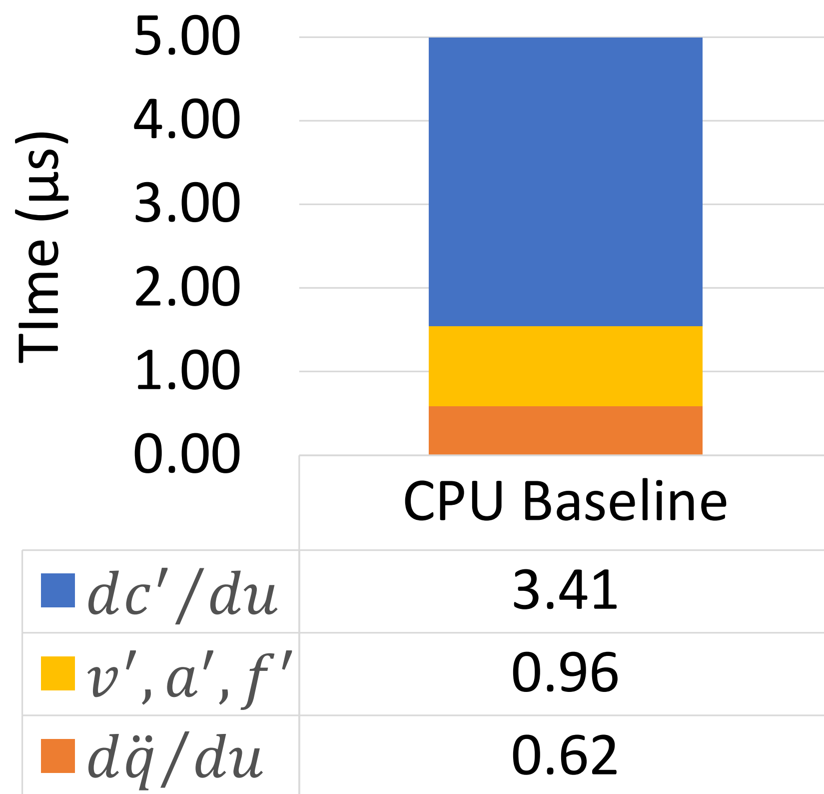
Place Zoom Headshot Here

Algorithm 3 $\nabla\text{Dynamics}(q, \dot{q}, \ddot{q}, f^{ext}) \rightarrow \partial\ddot{q}/\partial u$

1: $v', a', f', X, S, I \leftarrow \text{RNEA}(q, \dot{q}, \ddot{q}, f_{ext})$

2: $\partial c'/\partial u = \nabla\text{RNEA}(\dot{q}, v', a', f', X, S, I)$

3: $\partial\ddot{q}/\partial u = -M^{-1}\partial c'/\partial u$



The Gradient of Rigid Body Dynamics

Place Zoom Headshot Here

Algorithm 2 ∇RNEA$(\dot{q}, v, a, f, X, S, I) \rightarrow \partial c/\partial u$

for link $i = 1 : N$ **do**

$$\frac{\partial v_i}{\partial u} = {}^iX_{\lambda_i}\frac{\partial v_{\lambda_i}}{\partial u} + \begin{cases} \left({}^iX_{\lambda_i}v_{\lambda_i}\right) \times S_i & u \equiv q \\ S_i & u \equiv \dot{q} \end{cases}$$

$$\frac{\partial a_i}{\partial u} = {}^iX_{\lambda_i}\frac{\partial a_{\lambda_i}}{\partial u} + \frac{\partial v_{\lambda_i}}{\partial u} \times S_i\dot{q}_i + \begin{cases} \left({}^iX_{\lambda_i}a_{\lambda_i}\right) \times S_i \\ v_i \times S_i \end{cases}$$

$$\frac{\partial f_i}{\partial u} = I_i\frac{\partial a_i}{\partial u} + \frac{\partial v_i}{\partial u} \times^* I_i v_i + v_i \times^* I_i\frac{\partial v_i}{\partial u}$$

for link $i = N : 1$ **do**

$$\frac{\partial c_i}{\partial u} = S_i^T\frac{\partial f_i}{\partial u}$$

$$\frac{\partial f_{\lambda_i}}{\partial u} \mathrel{+}= {}^iX_{\lambda_i}^T\frac{\partial f_i}{\partial u} + {}^iX_{\lambda_i}^T\left(S_i \times^* f_i\right)$$

Algorithmic Features

The Gradient of Rigid Body Dynamics

Place Zoom Headshot Here

Algorithm 2 $\nabla \text{RNEA}(\dot{q}, v, a, f, X, S, I) \rightarrow \partial c / \partial u$

for link $i = 1 : N$ **do**

$$\frac{\partial v_i}{\partial u} = {}^{i}X_{\lambda_i} \frac{\partial v_{\lambda_i}}{\partial u} + \begin{cases} ({}^{i}X_{\lambda_i} v_{\lambda_i}) \times S_i & u \equiv q \\ S_i & u \equiv \dot{q} \end{cases}$$

$$\frac{\partial a_i}{\partial u} = {}^{i}X_{\lambda_i} \frac{\partial a_{\lambda_i}}{\partial u} + \frac{\partial v_{\lambda_i}}{\partial u} \times S_i \dot{q}_i + \begin{cases} ({}^{i}X_{\lambda_i} a_{\lambda_i}) \times S_i \\ v_i \times S_i \end{cases}$$

$$\frac{\partial f_i}{\partial u} = I_i \frac{\partial a_i}{\partial u} + \frac{\partial v_i}{\partial u} \times^* I_i v_i + v_i \times^* I_i \frac{\partial v_i}{\partial u}$$

for link $i = N : 1$ **do**

$$\frac{\partial c_i}{\partial u} = S_i^T \frac{\partial f_i}{\partial u}$$

$$\frac{\partial f_{\lambda_i}}{\partial u} \mathrel{+}= {}^{i}X_{\lambda_i}^T \frac{\partial f_i}{\partial u} + {}^{i}X_{\lambda_i}^T (S_i \times^* f_i)$$

Algorithmic Features

Fine-Grained Parallelism

Small Working Set Size

The Gradient of Rigid Body Dynamics

Place Zoom Headshot Here

Algorithm 2 $\nabla\text{RNEA}(\dot{q}, v, a, f, X, S, I) \rightarrow \partial c / \partial u$

for link $i = 1 : N$ **do**

$$\frac{\partial v_i}{\partial u} = {}^iX_{\lambda_i}\frac{\partial v_{\lambda_i}}{\partial u} + \begin{cases} \left({}^iX_{\lambda_i} v_{\lambda_i}\right) \times S_i & u \equiv q \\ S_i & u \equiv \dot{q} \end{cases}$$

$$\frac{\partial a_i}{\partial u} = {}^iX_{\lambda_i}\frac{\partial a_{\lambda_i}}{\partial u} + \frac{\partial v_{\lambda_i}}{\partial u} \times S_i\dot{q}_i + \begin{cases} \left({}^iX_{\lambda_i} a_{\lambda_i}\right) \times S_i \\ v_i \times S_i \end{cases}$$

$$\frac{\partial f_i}{\partial u} = I_i\frac{\partial a_i}{\partial u} + \frac{\partial v_i}{\partial u} \times^* I_i v_i + v_i \times^* I_i \frac{\partial v_i}{\partial u}$$

for link $i = N : 1$ **do**

$$\frac{\partial c_i}{\partial u} = S_i^T \frac{\partial f_i}{\partial u}$$

$$\frac{\partial f_{\lambda_i}}{\partial u} \mathrel{+}= {}^iX_{\lambda_i}^T \frac{\partial f_i}{\partial u} + {}^iX_{\lambda_i}^T \left(S_i \times^* f_i\right)$$

Algorithmic Features

Fine-Grained Parallelism

Structured Sparsity

Small Working Set Size

The Gradient of Rigid Body Dynamics

Place Zoom Headshot Here

Algorithm 2 $\nabla\text{RNEA}(\dot{q}, v, a, f, X, S, I) \rightarrow \partial c / \partial u$

for link $i = 1 : N$ **do**

$$\frac{\partial v_i}{\partial u} = {}^{i}X_{\lambda_i} \frac{\partial v_{\lambda_i}}{\partial u} + \begin{cases} \left({}^{i}X_{\lambda_i} v_{\lambda_i}\right) \times S_i & u \equiv q \\ S_i & u \equiv \dot{q} \end{cases}$$

$$\frac{\partial a_i}{\partial u} = {}^{i}X_{\lambda_i} \frac{\partial a_{\lambda_i}}{\partial u} + \frac{\partial v_{\lambda_i}}{\partial u} \times S_i \dot{q}_i + \begin{cases} \left({}^{i}X_{\lambda_i} a_{\lambda_i}\right) \times S_i \\ v_i \times S_i \end{cases}$$

$$\frac{\partial f_i}{\partial u} = I_i \frac{\partial a_i}{\partial u} + \frac{\partial v_i}{\partial u} \times^* I_i v_i + v_i \times^* I_i \frac{\partial v_i}{\partial u}$$

for link $i = N : 1$ **do**

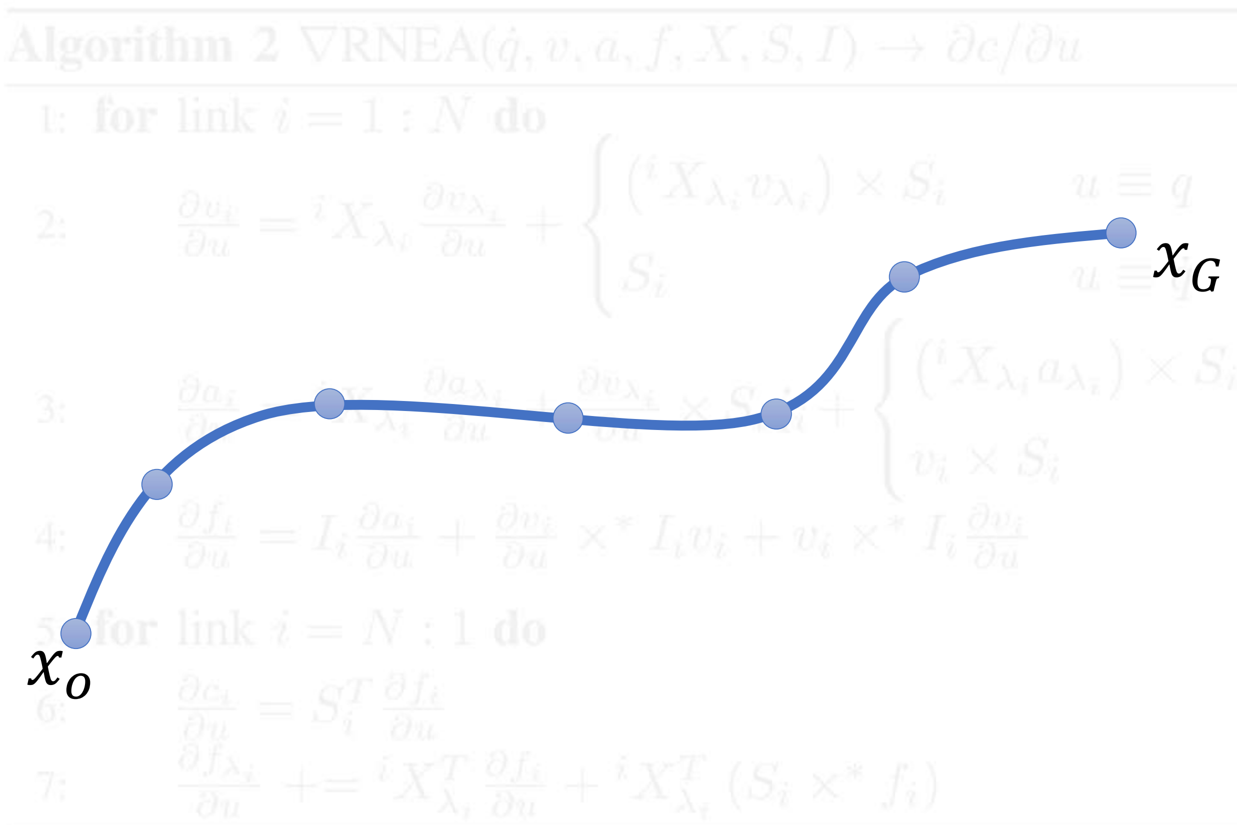
$$\frac{\partial c_i}{\partial u} = S_i^T \frac{\partial f_i}{\partial u}$$

$$\frac{\partial f_{\lambda_i}}{\partial u} \mathrel{+}= {}^{i}X_{\lambda_i}^T \frac{\partial f_i}{\partial u} + {}^{i}X_{\lambda_i}^T \left(S_i \times^* f_i\right)$$

- Algorithmic Features
- Fine-Grained Parallelism
- Structured Sparsity
- Irregular Data Patterns
- Sequential Dependencies
- Small Working Set Size

The Gradient of Rigid Body Dynamics
as a step of an MPC algorithm

Place Zoom Headshot Here



Algorithmic Features
Coarse-Grained Parallelism
Fine-Grained Parallelism
Structured Sparsity
Irregular Data Patterns
Sequential Dependencies
Small Working Set Size
I/O Overhead

Accelerating Robot Dynamics Gradients on a CPU, GPU, and FPGA

Place Zoom Headshot Here

1. Motivation
2. CPUs, GPUs, and FPGAs
3. The Gradient of Rigid Body Dynamics
4. **Accelerated Design**
5. Results

The Gradient of Rigid Body Dynamics as a step of an MPC algorithm

Place Zoom Headshot Here

Algorithmic Features	CPU
Coarse-Grained Parallelism	moderate
Fine-Grained Parallelism	poor
Structured Sparsity	good
Irregular Data Patterns	moderate
Sequential Dependencies	good
Small Working Set Size	good
I/O Overhead	excellent

The Gradient of Rigid Body Dynamics as a step of an MPC algorithm

Place Zoom Headshot Here

Algorithmic Features	CPU	GPU
Coarse-Grained Parallelism	moderate	excellent
Fine-Grained Parallelism	poor	moderate
Structured Sparsity	good	moderate
Irregular Data Patterns	moderate	poor
Sequential Dependencies	good	poor
Small Working Set Size	good	moderate
I/O Overhead	excellent	poor

Algorithmic Refactoring is needed to effective target GPUs and FPGAs

Place Zoom Headshot Here

Algorithm 2 $\nabla\text{RNEA}(\dot{q}, v, a, f, X, S, I) \rightarrow \partial c/\partial u$

for link $i = 1 : N$ **do**

$$\frac{\partial v_i}{\partial u} = {}^iX_{\lambda_i}\frac{\partial v_{\lambda_i}}{\partial u} + \begin{cases} ({}^iX_{\lambda_i}v_{\lambda_i}) \times S_i & u \equiv q \\ S_i & u \equiv \dot{q} \end{cases}$$

$$\frac{\partial a_i}{\partial u} = {}^iX_{\lambda_i}\frac{\partial a_{\lambda_i}}{\partial u} + \frac{\partial v_{\lambda_i}}{\partial u} \times S_i\dot{q}_i + \begin{cases} ({}^iX_{\lambda_i}a_{\lambda_i}) \times S_i \\ v_i \times S_i \end{cases}$$

$$\frac{\partial f_i}{\partial u} = I_i\frac{\partial a_i}{\partial u} + \frac{\partial v_i}{\partial u} \times^* I_i v_i + v_i \times^* I_i \frac{\partial v_i}{\partial u}$$

for link $i = N : 1$ **do**

$$\frac{\partial c_i}{\partial u} = S_i^T \frac{\partial f_i}{\partial u}$$

$$\frac{\partial f_{\lambda_i}}{\partial u} \mathrel{+}= {}^iX_{\lambda_i}^T \frac{\partial f_i}{\partial u} + {}^iX_{\lambda_i}^T (S_i \times^* f_i)$$

Algorithm 4 $\nabla\text{RNEA-GPU}(\dot{q}, v, a, f, X, S, I) \rightarrow \partial c/\partial u$

for link $i = 1 : n$ **in parallel do**

$$\alpha_i = {}^iX_{\lambda_i}v_{\lambda_i} \quad \beta_i = {}^iX_{\lambda_i}a_{\lambda_i} \quad \gamma_i = I_i v_i$$

$$\alpha_i = \alpha_i \times S_i \quad \beta_i = \beta_i \times S_i \quad \delta_i = v_i \times S_i$$

$$\zeta_i = f_i \times S_i \quad \eta_i = v_i \times^*$$

$$\zeta_i = -{}^iX_{\lambda_i}^T\zeta_i \quad \eta_i = \eta_i I_i$$

for link $i = 1 : n$ **do**

$$\frac{\partial v_i}{\partial u} = {}^iX_{\lambda_i}\frac{\partial v_{\lambda_i}}{\partial u} + \begin{cases} \alpha_i & u \equiv q \\ S_i & u \equiv \dot{q} \end{cases}$$

for link $i = 1 : n$ **in parallel do**

$$\mu_i = \frac{\partial v_i}{\partial u} \times^* \quad \rho_i = \frac{\partial v_{\lambda_i}}{\partial u} \times S_i\dot{q}_i + \begin{cases} \beta_i \\ \delta_i \end{cases}$$

for link $i = 1 : n$ **do**

$$\frac{\partial a_i}{\partial u} = {}^iX_{\lambda_i}\frac{\partial a_{\lambda_i}}{\partial u} + \rho_i$$

for link $i = 1 : n$ **in parallel do**

$$\frac{\partial f_i}{\partial u} = I_i\frac{\partial a_i}{\partial u} + \mu_i\gamma_i + \eta_i\frac{\partial v_i}{\partial u}$$

for link $i = n : 1$ **do**

$$\frac{\partial f_{\lambda_i}}{\partial u} \mathrel{+}= {}^iX_{\lambda_i}^T\frac{\partial f_i}{\partial u} + \zeta_i$$

for link $i = n : 1$ **in parallel do**

$$\frac{\partial c_i}{\partial u} = S_i^T\frac{\partial f_i}{\partial u}$$

The Gradient of Rigid Body Dynamics as a step of an MPC algorithm

Place Zoom Headshot Here

Algorithmic Features	CPU	GPU	FPGA
Coarse-Grained Parallelism	moderate	excellent	moderate
Fine-Grained Parallelism	poor	moderate	excellent
Structured Sparsity	good	moderate	excellent
Irregular Data Patterns	moderate	poor	excellent
Sequential Dependencies	good	poor	good
Small Working Set Size	good	moderate	excellent
I/O Overhead	excellent	poor	poor

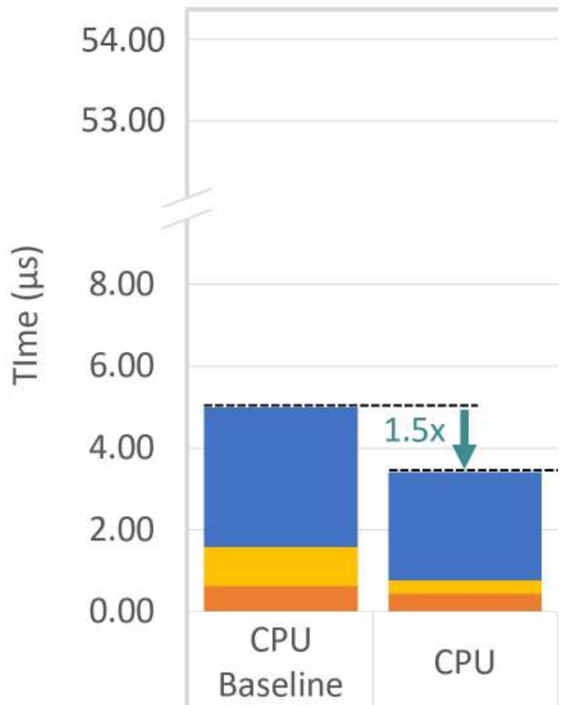
Accelerating Robot Dynamics Gradients on a CPU, GPU, and FPGA

Place Zoom Headshot Here

1. Motivation
2. CPUs, GPUs, and FPGAs
3. The Gradient of Rigid Body Dynamics
4. Accelerated Design
5. **Results**

These code optimizations and refactoring greatly improved single computation latency

Place Zoom Headshot Here

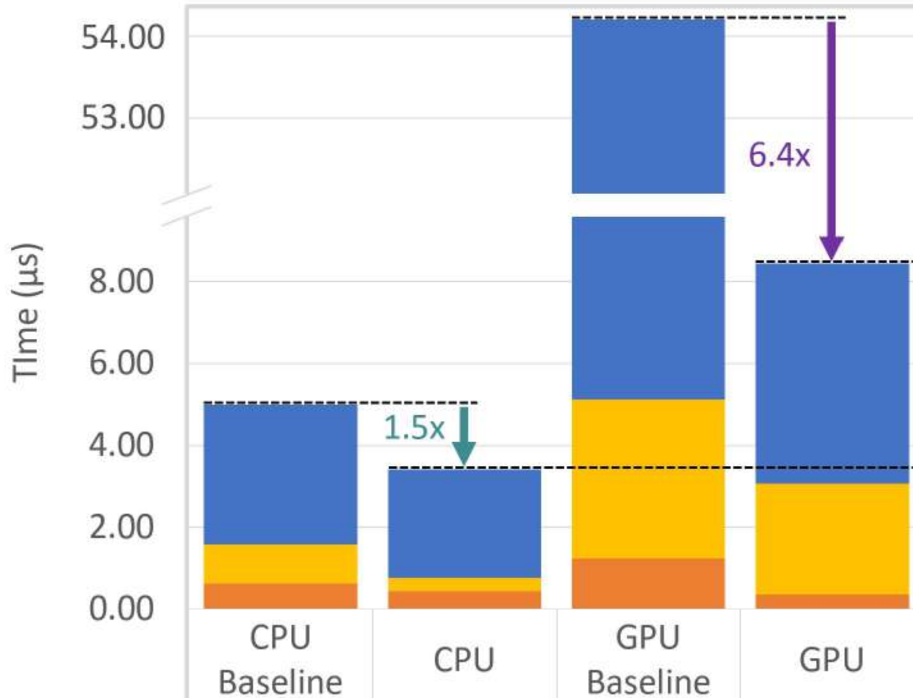


	CPU Baseline	CPU
Total	4.99	3.42
dc'/du	3.41	2.65
v', a', f'	0.96	0.34
$d\ddot{q}/du$	0.62	0.44

Hardware optimizations even improve CPU performance

These code optimizations and refactoring greatly improved single computation latency

Place Zoom Headshot Here

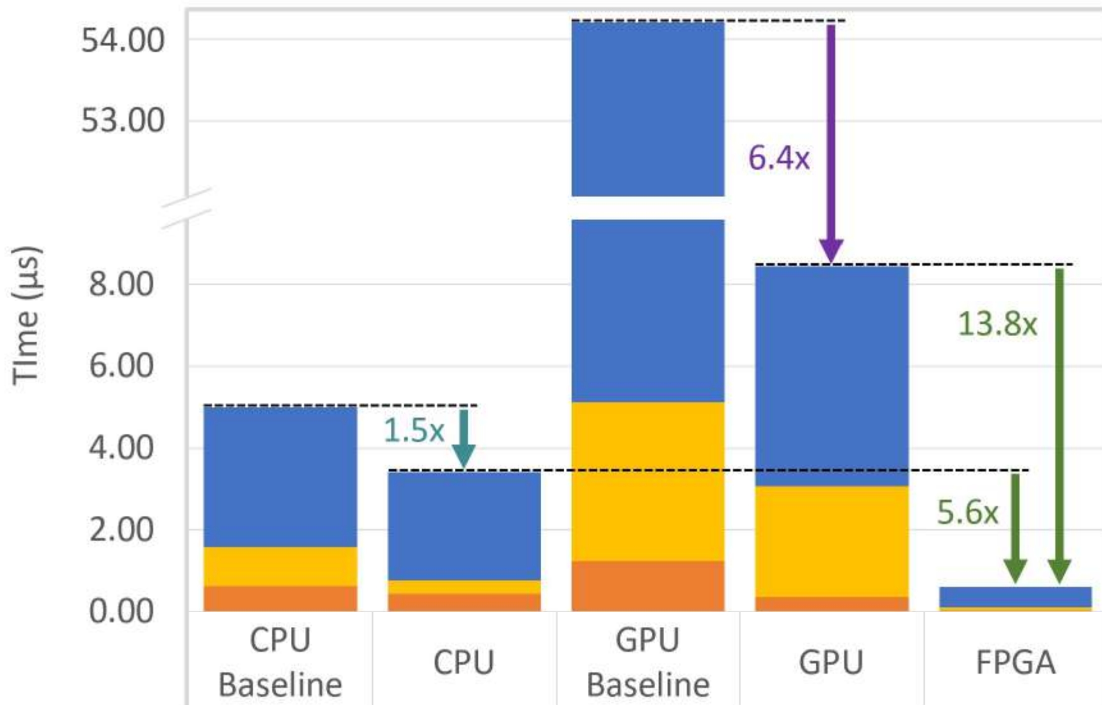


	CPU Baseline	CPU	GPU Baseline	GPU
Total	4.99	3.42	54.41	8.44
dc'/du	3.41	2.65	49.29	5.37
v', a', f'	0.96	0.34	3.89	2.71
$d\ddot{q}/du$	0.62	0.44	1.23	0.36

The GPU is built for large scale parallelism

These code optimizations and refactoring greatly improved single computation latency

Place Zoom Headshot Here

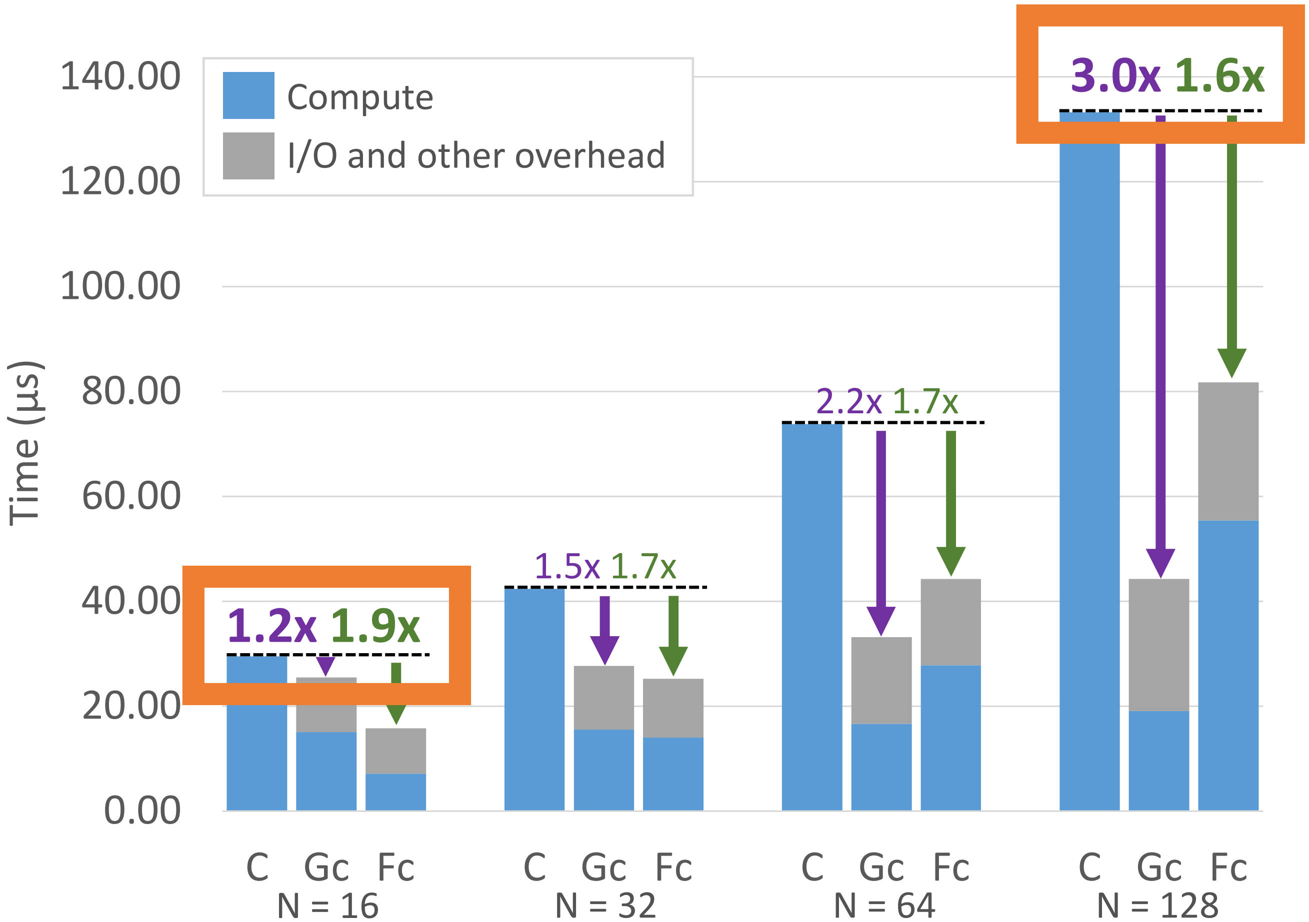


	CPU Baseline	CPU	GPU Baseline	GPU	FPGA
Total	4.99	3.42	54.41	8.44	0.61
dc'/du	3.41	2.65	49.29	5.37	0.50
v', a', f'	0.96	0.34	3.89	2.71	0.07
$d\ddot{q}/du$	0.62	0.44	1.23	0.36	0.04

Custom circuits are incredibly fast!

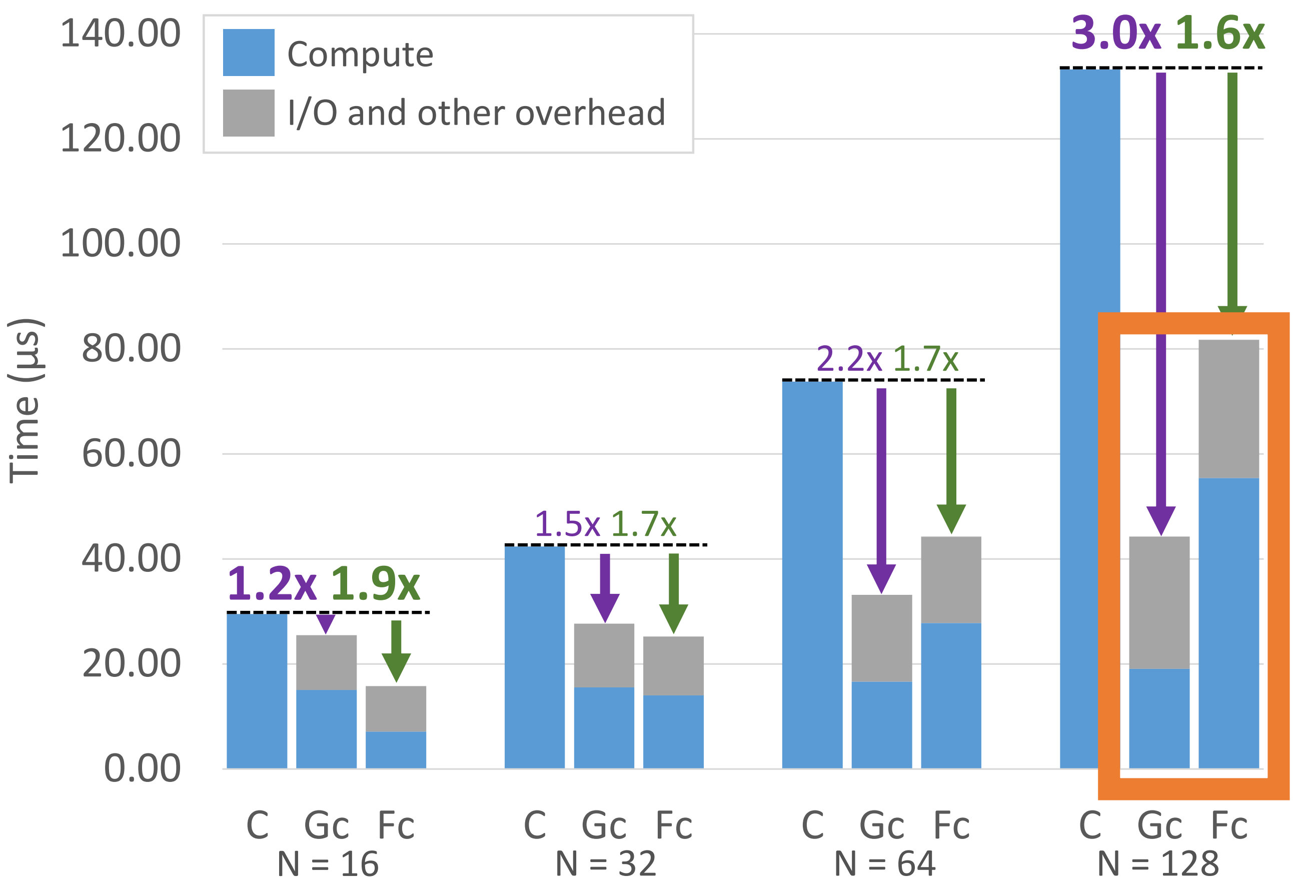
The GPU scales best and the FPGA is the fastest at low numbers of computations

Place Zoom Headshot Here



The GPU scales best and the FPGA is the fastest at low numbers of computations

Place Zoom Headshot Here



Move everything onto the accelerator (if possible)!

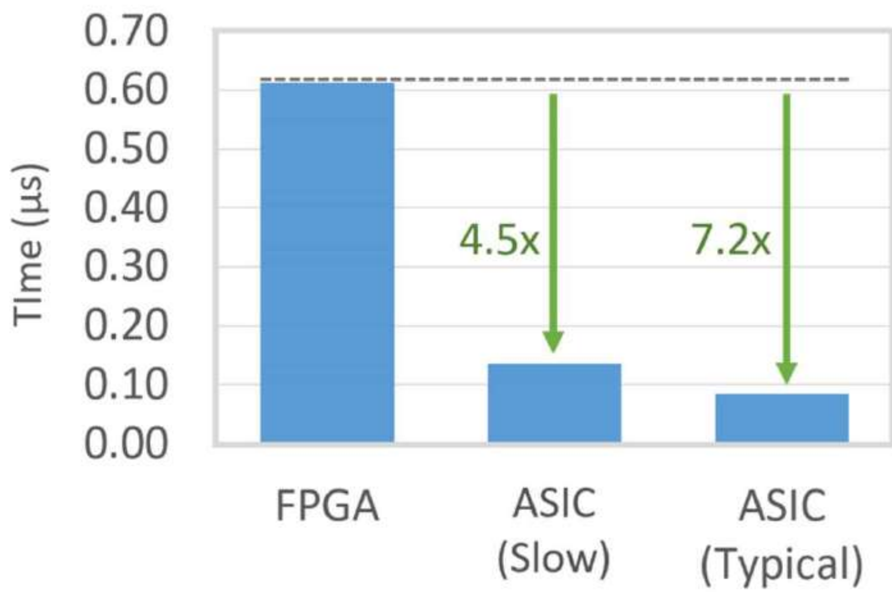
What's next?

Place Zoom Headshot Here

What’s next?

Place Zoom Headshot Here

1. ASIC acceleration to improve both **latency** and **coarse-grained parallelism**



[S.M. Neuman et al. "Robomorphic Computing: A Design Methodology for Domain-Specific Accelerators Parameterized by Robot Morphology,” ASPLOS 2021]

What’s next?

Place Zoom Headshot Here

1. ASIC acceleration to improve both latency and coarse-grained parallelism

2. **Code generation from URDFs**

Actively in progress but/and our current code can be found at:

http://bit.ly/fast-rbd-grad

Accelerating Robot Dynamics Gradients on a CPU, GPU, and FPGA

Place Zoom Headshot Here

http://bit.ly/fast-rbd-grad brian_plancher@g.harvard.edu

Refactoring and **partitioning** the gradient of rigid body dynamics to expose different **hardware-compatible features** for GPUs and FPGAs provides as much as a **3.0x end-to-end speedup**

Hardware-Software Co-Design for Parallelism

Harvard John A. Paulson School of Engineering and Applied Sciences

MIT CSAIL

Massachusetts Institute of Technology

This material is based upon work supported by the National Science Foundation (under Grant DGE1745303 and Grant 2030859 to the Computing Research Association for the CIFellows Project), and the Defense Advanced Research Projects Agency (under Grant HR001118C0018). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the funding organizations