A Hybrid Continuous-Discrete Computing Stack for Robotics
Project Description

Yipeng Huang
yipeng@cs.columbia.edu

We live in a world that is *continuous*: our eyes see scenes defined by continuous curves and intensities of colors, hear sounds formed by continuously varying pressure waves, and we sense velocity and temperature as continuous signals. In contrast, our computers operate in a discrete world, despite the continuous nature of the world at the scale our senses perceive. We finely divide time and space into quanta in order to model the real world on our digital computers. The inherent costs of digital computing have been ameliorated by digital computers' exponential growth in computing power relative to energy consumption and size. However, as scaling in digital computers comes to an end, we must explore alternatives to digital, discrete computing.

An analog accelerator that operates on continuous signals has been proposed for use in speeding up calculation of floating-point math, nonlinear math, and solving differential equations. Continuous computation promises to deliver adequate precision, low power, low latency, and low area cost accelerators for use in systems that extensively interact with the physical world.

Continuous computation and various forms of approximate computing has been proposed and explored for vision and graphics, machine learning, and certain forms of scientific and engineering computation. Robotics computation has promise to be another domain in which continuous computing can be applied. We have found that robots have poor utilization of their on board computing resources, with excess power wasted on communication, synchronization, and on excess floating point precision. We propose using continuous computing to increase computing efficiency in robotics.

In this project, we will take previously identified kernels of robotics code that consume a high percentage of computing resources. We will design an ISA for an emerging continuous coprocessor, and we will retarget the robotics kernels of code to the continuous ISA.

This project serves as an end-to-end proof of concept for multiple layers of the continuous computing stack—from the application to just below the ISA in hardware. The most desired findings are a first draft and viable ISA, along with the findings that would lead to a proposal on intermediate layers in the continuous compiler stack. We recognize the workload suite and the hardware design are evolving, and the findings from this project will change accordingly.

To evaluate the performance benefit of the design, we will evaluate the retargeted robotics kernel in a simulated runtime. We will make an argument on power, area, and timing gains using data collected from the hardware implementation and measurements. We expect that the newly designed continuous computing stack would better match the robotics kernel code, and alleviate overhead incurred in loops, recursion, and memory accesses.

The key deliverables for this project, along with their purpose and dependencies, follows:

	Deliverable Name	**Description**	**Detailed Elaboration**	**Dependencies**
Workload	Robotics full system workload	A description and collection of robotics simulator infrastructure	Robots include Cornell Ranger and Willow Garage Turtlebot, PR2 robots	
	Microbenchmarks	A workload characterization identifying hotspot kernels in robotics software	Hotspots include Cornell Ranger ODE solver kernel, ROS simultaneous location and mapping stack, ROS navigation stack, ROS object detection and manipulation stack	
Interface	ISA	An analog, continuous, parallel ISA	For the first HCDC design, a lot will be exposed in the ISA. We will write in terms of both minimally feasible and future ideal designs	Functional unit interfaces, register set, network design, system bringup sequence, configuration and calibration routines
Implementation & Test	Robotics hotspots retargeted to continuous ISA	A manual rewrite of library source code so it calls continuous ISA instructions and follows continuous bringup sequence		
	Evaluation of retargeted performance	A simulator demonstration or analysis on performance gains		Analog functional unit latencies
	Optional: a proposal on continuous compiler toolchain components	Desiderata for an analog language, compiler, and execution model.		

Bibliography

Reference	**Relevance to project**
G.E.R. Cowan, R.C. Melville, and Y.P. Tsividis. A vlsi analog computer/digital computer accelerator. *Solid-State Circuits, IEEE Journal of*, 41(1):42 – 53, jan. 2006.	An existing implementation of bringup sequence and network on chip design for an analog accelerator.
Karthikeyan Sankaralingam, Ramadass Nagarajan, Robert McDonald, Rajagopalan Desikan, Saurabh Drolia, M. S. Govindan, Paul Gratz, Divya Gulati, Heather Hanson, Changkyu Kim, Haiming Liu, Nitya Ranganathan, Simha Sethumadhavan, Sadia Sharif, Premkishore Shivakumar, Stephen W. Keckler, and Doug Burger. 2006. Distributed Microarchitectural Protocols in the TRIPS Prototype Processor. In *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture* (MICRO 39). IEEE Computer Society, Washington, DC, USA, 480-491. DOI=10.1109/MICRO.2006.19 http://dx.doi.org/10.1109/MICRO.2006.19	A demonstration of microarchitectural chip configuration bringup and protocol
Kuhlman, M.J.; Arvelo, E.; Shuoxin Lin; Abshire, P.A.; Martins, N.C., "Mixed-signal architecture of randomized receding horizon control for miniature robotics," *Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on* , vol., no., pp.570,573, 5-8 Aug. 2012	A demonstration of retargeting a robotics library to a mixed-signal architecture