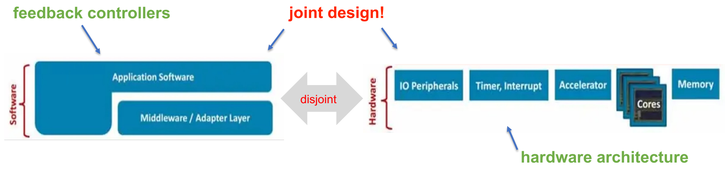
Collaborative Research: CPS: Frontier:
Computation-Aware Algorithmic Design for Cyber-Physical Systems

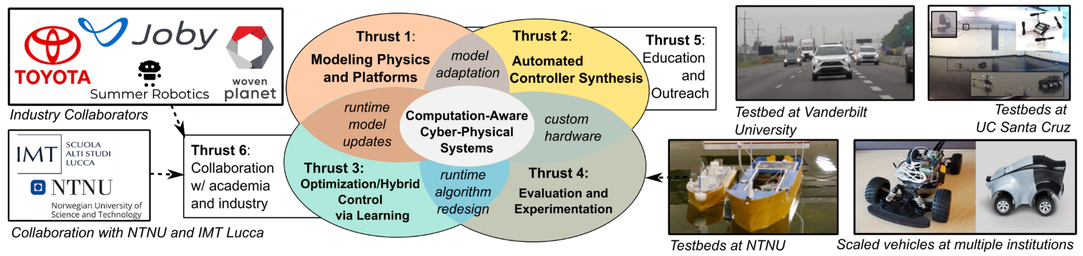
PIs: Ricardo Sanfelice (UCSC), Murat Arcak (UC Berkeley), Linh Thi Xuan Phan (Penn), Jonathan Sprinkle (Vanderbilt), Majid Zamani (CU Boulder), Abhishek Halder (UCSC), Heiner Litz (UCSC)

Overview

Project Goal

Generate tools for high-performance joint design of hardware, software, and control algorithms for cyber-physical systems in which control algorithms and computing platforms adapt to each other.





Challenges

Broad Challenges to Modeling Hardware and Software

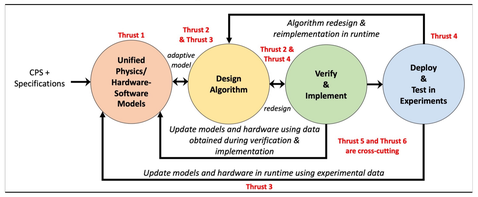
- Building models that integrate hardware and software features is challenging.
- Enabling computation-aware control algorithms requires a detailed understanding of the underlying hardware, performance monitoring capabilities, and means to modify and optimize the hardware.

Challenges to Feedback Control Design for Standard Platforms

- The implementation of the controllers resulting from symbolic approaches on a digital device requires a considerable amount of memory while those resulting from optimization approaches demand high computation power.
- The control algorithms required for intelligent transportation exceed the computational capabilities of traditional single-core processors.
- Porting existing control algorithms to multi-core processors introduces significant software development overheads while the performance benefits are often limited because some algorithms exhibit only narrow scalability.

Integrative Thrusts & Focus

The effort pertains to CPS with attributes including, but not limited to, hardware architecture, physical platforms, hardware-software models, as well as control, safety, optimization, and learning.



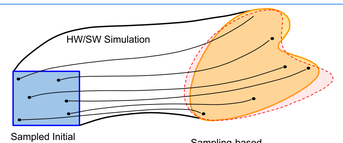
Driving Application Domain

Technical Approach

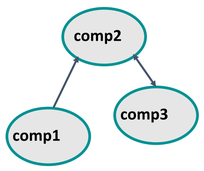
Thrust 1: Iterative Modeling of Physics and Platform

Approach:

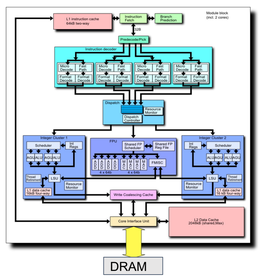
- We design modeling and simulation infrastructure to enable performance, power, and utilization analysis.
- We enable iterative development of controller, hardware platform in a given physical environment
- We analyze microarchitectural, controller-specific hardware improvements



- Learning-based Reachability Analysis of Unified Hardware-Software Models



- Unified Hardware-Software Models

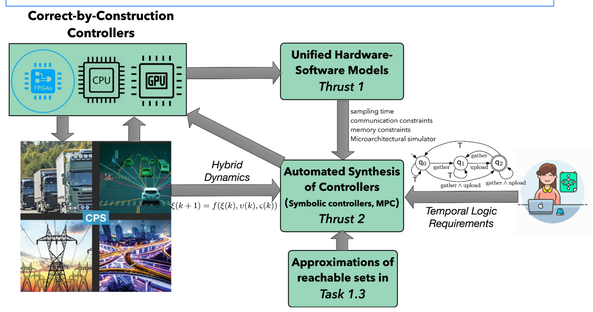


- Microarchitectural CPS Simulator

Thrust 2: Automated Synthesis of Controllers

Approach:

- We take into consideration the hardware-software characteristics investigated in Thrust 1 and synthesize embedded control software rendering complex logic specifications over concrete systems
- We propose specialized hardware for the synthesized embedded controllers to gain significant computational performance and energy efficiency
- We propose analysis techniques and optimization strategies to achieve timing predictability of the execution of synthesized control software and desirable closed-loop control performance, in turn, feeding information back to the customized hardware to design

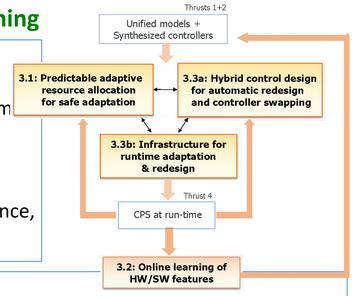


Proposed platform- and resource constrained formal synthesis

Thrust 3: Real-time Optimization and Adaptive Control through Learning

Approach:

Leverage experimental data from a running system to update the models, controllers, and resource allocation on the fly, to optimize safety, performance, and resource efficiency.



Key research questions:

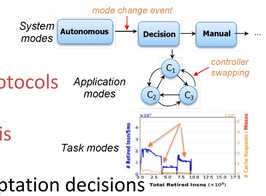
- When? → Online monitoring & detection methods
- What? → Control/scheduling co-design decisions
- How? → Runtime mechanisms and protocols
- Adaptation must be done promptly, safely, and efficiently

Predictable adaptive resource allocation:

- Holistically captures resource interactions & WCET impacts
- Exploit dynamism with multi-mode allocation

Infrastructure for safe runtime adaptation and redesign:

- OS, hypervisor and network protocols for safe adaptation
- Overhead-aware formal analysis of mode transition protocols
- Data-driven exploration of adaptation decisions
- Efficient monitoring and detection techniques

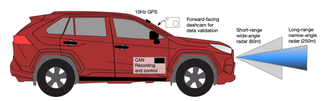


Design of supervisory algorithms for learning hardware operation:

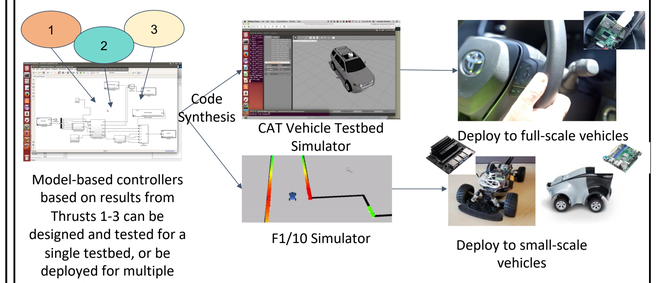
- Online learning of h/w feature joint stochastic processes
- Hardware exploration-exploitation trade-off by control s/w

Thrust 4: Multi-platform Computational and CPS Validation

Approach: leverage code generation and open testbeds to deploy across platforms, and share results.

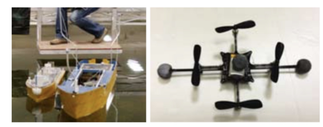


Naturalistic Driving Data Platform provides large-scale data



Model-based controllers based on results from Thrusts 1-3 can be designed and tested for a single testbed, or be deployed for multiple testbeds

Approach: through model-based software approaches, testbeds and data can be more broadly used, and more quickly deployed.



A 1:30 scale marine vessel testbed (partnership with NTNU)

Crazyflie quadrotor (testbed at UC Santa Cruz)

Challenge problems in ship docking (marine), will push reach-avoid work. Computational limitations and uncertainty of the crazyflie platform exercise aspects of robustness.

This project is funded by the National Science Foundation under Grant no. 2111688

Intellectual merit

- New CPSs hardware and control software that adapt to the specifications and the environment they are deployed on.
- New tools to codesign hardware and control software reducing development cost and time.

Broader impacts

- Improving the safety and energy-efficiency of autonomous systems.
- Providing open source models, algorithms, software, and platform designs for deployment in industrial systems.
- Train the workforce of the future in CPS.

Thrust 5:

- Expanding underrepresented minorities (URM) in CPS across campuses: UCSC, UC Berkeley, U. Penn, Vanderbilt Univ., CU Boulder
- Cross-institutional co-advising of senior projects in computation-aware CPS
- Computation-aware Algorithmic Design of CPS (CAADCPS) workshops in CPS-IoT weeks