

## 13 Model-based Software Synthesis of Cyber-Physical Systems

Peng Deng, Advisor: Dr. Qi Zhu, University of California, Riverside



### Introduction

#### Challenges on CPS software design:

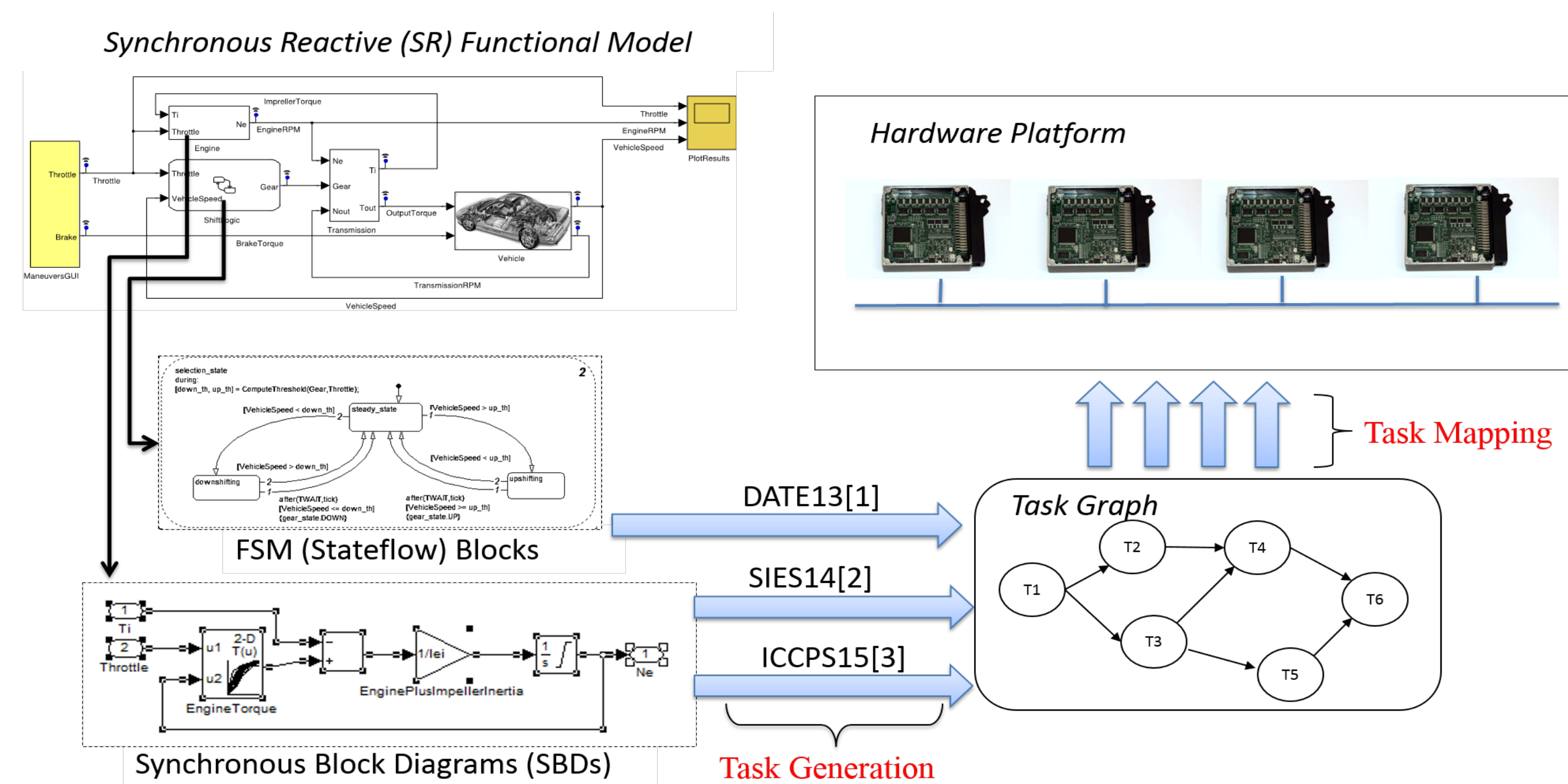
- Complexity (e.g. 100 million lines of code in a premium car).
- Stringent requirements on timing and other design metrics (performance, reliability, extensibility, etc.).
- Usage of multicore and distributed platforms.

#### Current practice:

- Lack of timing consideration in task generation.
- Gap between task generation and task mapping.

#### Our approach:

- Model-based software synthesis (MS2) framework that integrates task generation and task mapping with consideration of timing and other related design metrics.

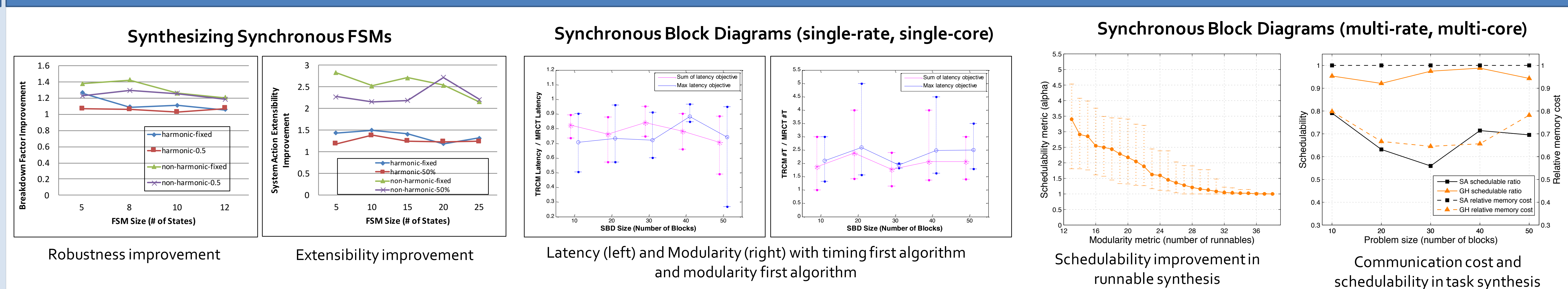


Model-based software synthesis (MS2) framework overview

### MS2 Framework

Functional Model	Design Objectives	Approach
<p><b>Synchronous Reactive FSM</b></p>	<p>Constraints:</p> <ul style="list-style-type: none"> <li>• Task execution order.</li> </ul> <p>Design metrics:</p> <ul style="list-style-type: none"> <li>• Extensibility: capability to accommodate modifications and upgrades.</li> <li>• Breakdown factor: timing robustness of the system.</li> </ul>	<ul style="list-style-type: none"> <li>• Metrics to measure extensibility and breakdown factor</li> <li>• Heuristic for multitask generation and scheduling.</li> </ul>
<p><b>Synchronous Block Diagram (Single rate)</b></p>	<p>Constraints:</p> <ul style="list-style-type: none"> <li>• Reusability: no false input-output dependency.</li> <li>• Code size: no code duplication among tasks.</li> </ul> <p>Design metrics:</p> <ul style="list-style-type: none"> <li>• Modularity: number of generated tasks.</li> <li>• Timing: output latencies.</li> </ul>	<p>Alg. 1. Optimal <i>timing</i>, best possible modularity, while guarantee reusability and code size</p> <p>Alg. 2. Optimal <i>modularity</i>, best possible timing, while guarantee reusability and code size</p>
<p><b>Synchronous Block Diagram (Multi-rate)</b></p>	<p><b>Runnable Synthesis:</b></p> <p>Constraints:</p> <ul style="list-style-type: none"> <li>• Reusability</li> <li>• Code size</li> <li>• Execution order</li> </ul> <p>Design metrics:</p> <ul style="list-style-type: none"> <li>• Modularity</li> <li>• Schedulability</li> </ul> <p><b>Task Synthesis:</b></p> <p>Constraints:</p> <ul style="list-style-type: none"> <li>• Execution order</li> <li>• Schedulability</li> <li>• End to end latency</li> </ul> <p>Design metrics:</p> <ul style="list-style-type: none"> <li>• Communication cost</li> </ul>	<ul style="list-style-type: none"> <li>• Defined Firing Execution Timed Automata (FETA) to capture timing behavior throughout the flow.</li> <li>• Runnable synthesis: algorithms to optimize <i>abstract</i> schedulability.</li> <li>• Task synthesis: algorithms to evaluate schedulability and optimize communication cost.</li> </ul>

### Results



[1] P. Deng, F. Cremona, Q. Zhu, M. Di Natale and H. Zeng. "A Model-based Synthesis Flow for Automotive CPS", ICCPS 2015  
 [2] P. Deng, Q. Zhu, M. Di Natale and H. Zeng. "Task Synthesis for Latency-sensitive Synchronous Block Diagram", SIES 2014  
 [3] Q. Zhu, P. Deng, M. Di Natale and H. Zeng. "Robust and Extensible Task Implementation of Synchronous Finite State Machines", DATE 2013  
 [4] B. Zheng, W. Li, P. Deng, L. Gerard, Q. Zhu and N. Shankar. "Design and Verification for Transportation System Security", DAC 2015

[5] M. Belviranli, P. Deng, L. Bhuyan, R. Gupta and Q. Zhu. "PeerWave: Exploiting Wavefront Parallelism on GPUs with Peer-SM Synchronization", ICS 2015  
 [6] P. Deng, Q. Zhu, A. Davare, A. Mourikis, X. Liu and M. Di Natale. "WiP Abstract: An Efficient Control-driven Period Optimization Algorithm for Distributed Real-time Systems", ICCPS 2014  
 [7] C. Zhang, P. Deng, H. Geng, J. Liu, Q. Zhu and Y. Shi. "MSim: A General Cycle Accurate Simulation Platform for Memcomputing Studies", DATE 2014  
 [8] Q. Zhu and P. Deng. "Design Synthesis and Optimization for Automotive Embedded Systems", ISPD 2014