UNIVERSITY OF CALIFORNIA,
IRVINE

Design Space Exploration in Cyber-Physical Systems

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Computer Science

by

Maral Amir

Thesis Committee:
Professor Tony Givargis, Chair
Professor Alex Nicolau
Professor Ian G. Harris

2019

DEDICATION

To my parents, Shahin Sadraie and Morteza Amir, for their unconditional love and support, and for always believing in me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

I would like to express my gratitude to my adviser, Professor Tony Givargis. He has helped me become a better and stronger person and has motivated me to develop skills needed to grow as a successful individual in the society.

I would like to thank my committee members Professor Alex Nicolau and Professor Ian G. Harris for their time and support.

I am forever thankful to my parents who have always given me courage, hope and support and to whom I owe all my success. Shahin Sadraie and Morteza Amir, you two are forever my heroes and I am grateful for all your love and kindness that help me reach my dreams. I am deeply grateful to my sisters, Ghazaleh Amir and Maryam Amir for all their love and protection and for always being there for me.

I want to thank UC Irvine graduate division, NSF (Grant Number 1563652), and the school of ICS for providing funding opportunities during my PhD program.

ABSTRACT OF THE THESIS

Design Space Exploration in Cyber-Physical Systems

By

Maral Amir

Master of Science in Computer Science

University of California, Irvine, 2019

Professor Tony Givargis, Chair

Cyber physical systems (CPS) integrate a variety of engineering areas such as control, mechanical and computer engineering in a holistic design effort. While interdependencies between the different disciplines are key attributes of CPS design science, little is known about the impact of design decisions of the cyber part on the overall system qualities. To investigate these interdependencies, this paper proposes a simulation-based Design Space Exploration (DSE) framework that considers detailed cyber system parameters such as cache size, bus width, and voltage levels in addition to physical and control parameters of the CPS. We propose an exploration algorithm that surfs the parameter configurations in the cyber physical sub-systems, in order to approximate the Pareto-optimal design points with regards to the trade-offs among the design objectives, such as energy consumption and control stability. We apply the proposed framework to a network control system for an inverted-pendulum application. The presented holistic evaluation of the identified Pareto-points reveals the presence of non-trivial trade-offs, which are imposed by the control, physical, and detailed cyber parameters. For instance the identified energy and control optimal design points comprise configurations with a wide range of CPU speeds, sample times and cache configuration following non-trivial zig-zag patterns. The proposed framework could identify and manage those trade-offs and, as a result, is an imperative first step to automate the search for superior CSP configurations.

Chapter 1

Introduction

Cyber physical systems (CPSs) integrate various engineering areas such as control, computer, mechanical, and network engineering [5]. The complex and heterogeneous design aspects of CPSs beget methodologies to combine the corresponding disciplines. For example, in automotive industry, it has been investigated that 80% of the innovations in a car are attributed to the computer systems [13].

Sequential and model-based design methodologies [2] are well-established techniques to cope with the complexity of designing CPSs. The idea is, first, to select a promising physical system, then define the controller and finally address design challenges of the embedded computer system. Such sequential separation of decisions reduces the complexity of the design efforts. However, like most greedy approaches, the overall solution is unlikely to be the best possible design due to missed trade-offs between cyber and physical design knobs.

Recent work [17] showed that holistic design approaches result in superior designs compared to sequential design flows. Holistic means that physical, control and cyber attributes of systems are evaluated concurrently. It is evident that properties of the cyber platform are important for overall system qualities, such as energy consumption or control quality [24].

Nevertheless, recent work usually condensed the complexity of the cyber part to derived properties such as sampling rate.

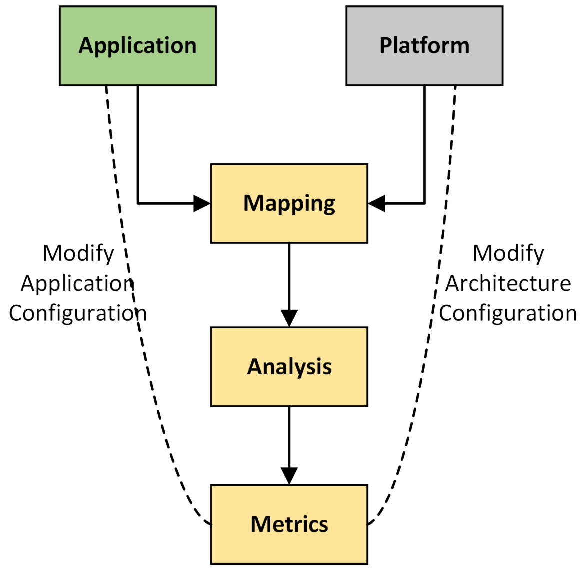

Figure 1.1: Y-chart methodology models the design representation from a top-down view. The behavioral, structural and layout views are each a branch of Y.

In our work, we explicitly investigate if and how specific properties of the cyber system, such as cache size, CPU configuration, memory, peripherals, interfere with overall system performance metrics, such as precision, stability or power consumption. Similar dependencies have been discovered and utilized between physical properties and control parameters, which led to the controller pruning [24]. But does such a connection exists between physical system (PS) and cyber system (CS)? We answer this question in this paper.

Model-based design provides a flexible environment to aid with specifying and analyzing the system requirements from a semantic-oriented perspective rather than an implementation-oriented methodology [5]. If dependable models from diverse disciplines are assembled at the design implementation level with no prior heterogeneous analysis and verification, system failures are likely to emerge and design faults are hard to trace [14]. One complete model of a cyber physical system provides a holistic framework to integrate physical entities, environment and computation platforms . That is, a codependent methodology is required that captures the properties of the system not only for local subsystems but also from a global interoperable point of view.

Our proposed framework implements a design space exploration methodology based on the Y-chart [8] depicted in Figure 1.1 which proposes a clear distinction between the application and architecture in the system design. The Y-chart methodology follows a top-down design approach around the three domains of behavior, structure and layout. As one traverses from the outer edge of the chart toward the center, finer design details emerge and the model abstraction level progressively gets refined. A Y-chart based framework enables the exploration and analysis of the system configurations for further alteration in architecture and application settings under competing mapping strategies.

Our primary tool to develop the networked control application application model that performs simulation and functional verification is Simulink [10]. For platform architecture, the parameterized system on a chip, Platune [9], is utilized to select the appropriate architectural parameter values (e.g. microprocessor, cache, peripherals. etc.) for the respective control application. The tool facilitates:

- Holistic and comprehensive exploration and analysis over the system level design space, and its parameters interaction and correlation.
- Compression of the design space to Pareto-optimal design points with regard to system performance metrics.
- Derivation of a dependency analysis to reduce the search, and support tool-based design space exploration of CPSs.

Cyber physical systems incorporate the interaction and correlation between computing and physical entities. Therefore, the design foundations, methods and tools of CPS engineering should accommodate the interdependency between the physical (mechanical, electrical, etc.) design and computing hardware and system software.

Our contributions in this work are listed as follows:

1. Propose a holistic and interdependent framework to design networked control cyber physical systems that govern the cyber, physical and control subsystems concurrently.

2. Present a simulation based framework that integrates Simulink environment with Platune's SoC and maps the control applications on the computing platform configured in Platune for system analysis, test and verification.

3. Present an efficient and concurrent algorithm to prune and explore the design space for networked control cyber physical systems.

The proposed framework is applied to a real application of control system for automated inverted pendulum to analyze and verify the efficiency and necessity of the suggested work.

The rest of the paper is organized as follows. A review of the state of the art design space exploration frameworks for networked control systems is reported in Section 2. Sections 3 includes the DSE problem statement and a tool-supported holistic methodology is proposed. We demonstrate the workings and effectiveness of our framework for the inverted pendulum example in Section 4. Finally, the conclusions and future work is reported in Section 5.

Chapter 2

Related Work

Holistic and model-based design approaches for CPSs have been an active research topic resulting in a range of compelling related works. [4] presented a top-down design framework which applies reusable blueprints of physical and cyber models to synthesize efficient CPSs. Moreover, [23] have benefited from this top-down system-level design approach of CPSs in automotive domain. They have extracted parameters from the physical systems and modeled their behavior for better optimization of controller in cyber system. They have modeled and estimated the dynamic behavior of the electric vehicle components, e.g. power train, hybrid electrical energy storage, and automotive climate control, in order to improve the performance of the vehicle in terms of driving range and energy consumption. However, we are suggesting that exploration and analysis of the cyber parameters in the controller design of these components does impact the system performance and stability, hence must be holistically optimized.

[19] proposed a framework for CPS Design Space Exploration in which the designers can define physical and computational components and include constraints on system parameters and assembly process. This paper presents the design space as a set of hierarchical AND-

OR elements with Boolean constrains included in design decisions. Feasible configurations are simulated to analyze the changes of variables of interest across different design alternatives. Automatic adaptation of software components in cyber physical systems production is investigated in Other works [20].

[15] focuses on the control/cyber co-design with automatic control selection and parameterization. However, discussed approaches do not consider parameterization of the physical subsystem and do not evaluate the impact of architectural design decisions in the cyber part. [18] presents a co-simulation based framework in Jitterbug to analyze the control quality during Design Space exploration. This paper is aligned with our work in terms of including the control quality in cyber physical systems design decisions. However, they did not consider the mapping process and the computing platform design alternatives for the respective controller application.

[11] introduces a collaborative modeling and simulation tool to design embedded control systems and model the physical plants. The Crescendo tool incorporates a combination of Discrete-Event (DE) controller models with the Continuous-Time (CT) models to allow multidisciplinary system designs. They suggest techniques to reduce the number of simulations for rapid Design Space Exploration applications. While related, our work is orthogonal with this approach and the two methodologies can be combined resulting in cumulative benefits.

The impact of physical design decisions in a holistic CPS Design Space Exploration was outlined in [3], but the cyber system still was represented solely by a sampling rate. Automatic adaptation of cyber components is investigated in [22], in form of an iteration-based exploration for preferable software parameter configuration, under consideration of product and raw material descriptions. In a practical use case, [16] proposes a virtual representation of the robot cell, containing its individual physical and cyber components. In our work we, for the first time, combine physical and cyber parameters in a single framework.

Our work is further related to co-simulation frameworks [7], that facilitate collaborative modeling and simulation of embedded control systems and physical plants, incorporating discrete-event controller models with the continuous-time models to allow multidisciplinary systems design. However, our tool-supported methodology extends this idea to combine the state-of-the art SoC exploration and model-based simulation tools with the design methodology that we describe next.

Chapter 3

The DSE Methodology

The integration of design configuration alternatives and stringent design constraints, produce a complex design space to be explored under tight time-to-market schedules. Design space exploration is performed in early development phases to select between multiple design configurations that meet design requirements. The ability to automatically explore the design space and identify solution candidates can be applied to a range of design and engineering problems, including systems integration and optimization.

Most design space exploration techniques explore the design space by repeatedly performing simulation runs or apply model driven analytical (or numeric) approaches that approximate system behavior, as outlined in [1]. The numeric/analytic approaches apply several restrictions and assumptions such as Markov property [6] to the application model. This class of methodologies which rely on the predictable architectures are appropriate for time critical and safety critical applications. On the other hand, simulation based techniques are generally used when the aforementioned assumptions made by the analytical models are inappropriate. Furthermore, simulation approaches are often used where the numerical analysis of the system model exceeds the time and space complexity of the development computer. Current state of the art arbitrarily applies either simulation and analytical methodologies for system design. Depending on the application, hybrid approaches that integrate approximate models from the analytical methodology with simulation based frameworks can be very helpful to reduce the design space. Our proposed methodology integrates Simulink to model the networked control application with Platune, a system on a chip (SoC) framework to design the computing platform architecture. Further descriptions of the aforementioned tools are mentioned in the next section.

3.0.1 Tools and Environments

In this section, we review the tools and technologies employed in the implementation of the proposed Design Space Exploration framework to design networked control cyber physical systems.

Simulink:

The proposed methodology employs the Simulink environment for model-based simulation and analysis due to the rapid design and algorithm exploration capabilities of the Simulink and code generation commercial tools available for this benchmark (i.e., Simulink Coder). The hierarchical design block representation of this graphical modeling tool simplifies the design complexity and level/language transition [1]. Simulink applies a set of programs called solvers to simulate the system models. A model is represented as a set of ordinary differential equations. According to the nature of the system (e.g., continuous, discrete, time complexity, etc.) a solver is handpicked to apply a numerical method to the system model and compute its states at successive time steps over the simulation time.

Platune:

Platune [9], is a system on a chip (SoC) that collects activity information (e.g. clock frequency) from the component's (e.g., processor, cache, etc.) simulators and feed the component power models to calculate the power consumption metric. Platune, compiles the application written in C program and links the runtime libraries, in order to map the target application to the SoC platform. A platform is a predesigned configurable processing system, that includes a parametrized microprocessor (e.g., CPU speed), parameterized memory (e.g. i$size, i$ line), parameterized interconnect buses (e.g., CPU i$ bus, CPU d$ bus) and parameterized peripherals (e.g. DCT CODEC, UART). Platune integrates a set of simulators for the aforementioned components of the SoC platform to perform functional simulation. The simulators collect activity information to compute power and performance metrics in collaboration with power models for each component. Platune carries a MIPS virtual machine to simulate the application software and generate a report on the power consumption and processing time for the user defined configuration of the SoC platform. The tool is designed for rapid simulation in the high abstraction level and efficient exploration in the exponential configuration space.

3.0.2 The Design Space Exploration Problem

A design space exploration problem is intended to find the optimal combination of values for the system (i.e., the design parameters, e.g., physical parameters, computing platform parameters, control system parameters, etc.). Exhaustive DSE algorithms search all possible combinations of the design variables, the design space, to find the optimal configurations, Pareto optimal design points, with regards to the design objectives (e.g. power consumption) and constraints (e.g., timing constraints). The target design objectives introduce the problem with mono-dimensional design space for systems with one design objective (e.g., power consumption or execution time) and multidimensional design space for systems with multiple design metrics (e.g., power consumption and error). A Pareto design point is considered a global optimum in the mono-dimensional design space with one design objective, while it forms a trade-off curve called Pareto curve in multi objective scenarios.

If the systems parameter configuration is comprised of a vector of n scalar values per parameter, the search problem is formulated as a constrained n-dimensional non-linear optimization problem and the design space is the multiplication of design parameters:

$$S_{\text{System}} = \prod_{n=numofparameters}^{i} P_i \tag{3.1}$$

large complex systems may include billions of design alternatives to be explored as part of the overall design space. A manual inefficient approach to DSE is considered labor intensive, error prone and time/space interactable. That is, an interrelated algorithm is needed not only to reduce the complex design space but also to monitor the design space configuration from a comprehensive and interactive perspective between the subsystem design spaces. The power to automatically surf the design space and explore the solution candidates, promotes DSE tools for many engineering tasks, including systems integration and optimization. This paper suggests a holistic methodology which combines the subsystem search spaces interactively and structure the global system design space in an interactive process rather than a sequential approach.

3.0.3 Design Space Exploration in Networked Control Cyber Physical Systems

In embedded CPS design, the global design space is defined as an integration of local design space configurations, in that the local represents the subsystems involved such as cyber computing platform, physical systems, physical environments and networked control systems.

Potential parameter configuration of the target system is a member of the global design space S_{CPS} in the embedded cyber physical system design. S_{CPS} is defined as the conditional multiplication of local design spaces as:

$$S_{\mathrm{CPS}} = \prod_{n=numofspaces}^{i} S_i | C_{\mathrm{CPS}} \tag{3.2}$$

The C_{CPS} represents a set of conditions and constraints that are imposed by the design objective metrics (functional and non-functional), local space interdependency, etc. Local design space elements S_i, encompass a set of input system level parameters as P1, P2, P3 . The parameter P_i for the respective local space S_i is represented as a tuple:

$$P_i = (S^i_{\mathrm{name}}, S^i_{\mathrm{parameter}}, S^i_{\mathrm{range}}) \tag{3.3}$$

Where S^i_{name} is the corresponding local design space name, $S^i_{\mathrm{parameter}}$ is the input parameter name and S^i_{range} is range for the associated parameter. The proposed methodology provides a holistic design space S_{CPS}, that evaluates multi discipline local design space modules in embedded CPS subsystems interactively. It introduces an abstract design space representation, and provides a comprehensive inter-operable tool-based framework. The methodology is employed to prune the design space and support automated DSE activities exploring the Pareto-optimal design alternative in cyber physical systems.

We consider three local design spaces for embedded cyber physical systems' design in the proposed framework:

- Cyber computing platform
- Physical system
- Networked control system

Accordingly, the global design space configuration for embedded CPS applications is defined as the conditional multiplication of the local spaces:

$$S_{\text{CPS}} = (S_{\text{Cyber}} \times S_{\text{Physical}} \times S_{\text{Control}})|C_{\text{CPS}} \tag{3.4}$$

The modular representation of the proposed DSE architecture is illustrated in Figure 2. The methodology follows the Y-chart philosophy to integrate behavioral, structural and layout abstraction levels as application, platform and mapping views respectively.

The local design spaces S_{Physical}, S_{Control} and S_{CPS} are introduced individually in the following sections.

I. **Physical and Control Design Space, S_{Physical}, S_{Control}:**

The proposed methodology employs Simulink as the model-based framework to surf the physical space S_{Physical} and networked control space S_{Control} to design networked control

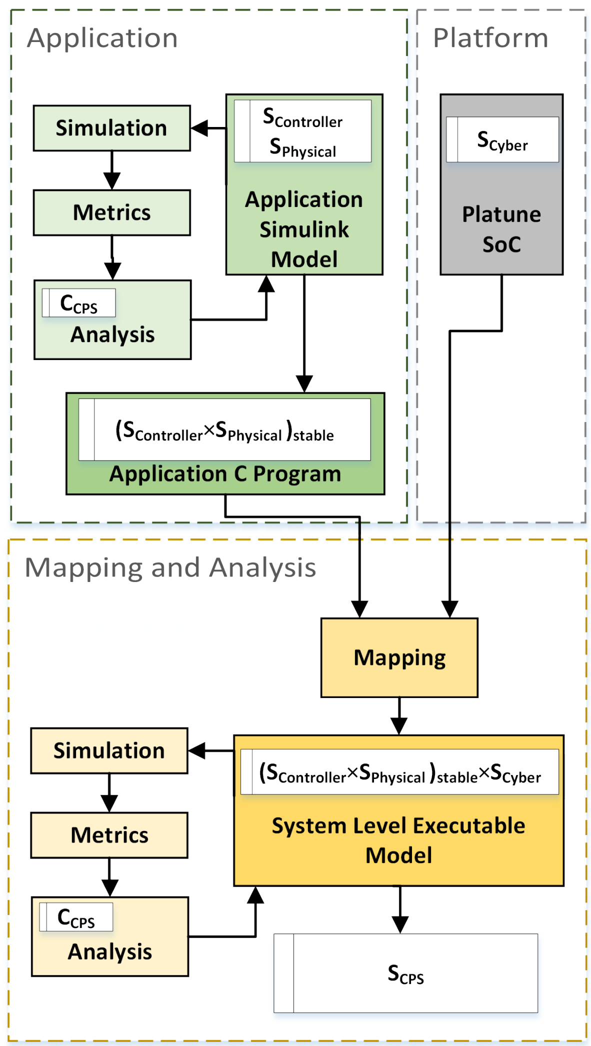

Figure 3.1: The proposed Design Space Exploration architecture integrates $S_{\mathbf{Physical}}$, $S_{\mathbf{Control}}$ and $S_{\mathbf{Cyberl}}$ design spaces interactively to design a networked controlled CPS. The three Y-chart branches are identified in the proposed architecture in dotted squares.

systems and optimize the design to meet predefined performance criteria. Simulink from MATLAB environment is one of the most popular tools among the researchers in various domains. Simulink provides customizable blocks from different areas such as motors, sensors, actuators, controllers to be integrated into physical components for CPS design. This process employs a model as an executable system specification for development. The simulation results for the Simulink models in the $S_{\text{Physical}} \times S_{\text{Control}}$ design space configurations, delivers design objectives such as the controller performance metric (e.g., stability) and the computed physical energy consumption for the specified design configuration in the $S_{\text{Physical}} \times S_{\text{Control}}$

design space.

Control engineers are chasing after a control system design which demonstrates stable behavior while meeting the timing requirements accommodated by the processing platform. Computer Engineers are obliged to design a computing platform to meet the timing requirements of the implemented application (i.e., control algorithm). Time delays or dead time (DTs) are ubiquitous in various system domains. Measurement, analysis, processing and communication lags impose time delays on the control systems [12]. Networked control systems incorporate controllers, sensors, actuator devices to perform several computational tasks and exchange data across the control communication field. The control delay τ_{c}^{k}, includes the computation delay induced by the computation and processing routines in addition to the communication delay which encompasses the sensor-to-controller and controller-to-actuator delays and will be:

$$\tau_k = \tau_{\mathrm{c}}^{k} + \tau_{\mathrm{sc}}^{k} + \tau_{\mathrm{ca}}^{k} \tag{3.5}$$

Where τ_{c}^{k} is the computation delay and τ_{sc}^{k} and τ_{ca}^{k} represent sensor-to-controller and controller-to-actuator delays respectively. Embedded system applications that embed microprocessors with bounded CPU performance, the processing delay can be significant and not to be disregarded in the control system design.

Computational delays can have substantial impact on the performance of a control system. The closed loop feedback of a control system produces unstable and oscillatory behavior if this delay is not compensated. That is, in cyber physical system design, it is crucial to take into account the computational delay of the system as part of the simulation. The proposed framework captures the computational delay of the control system implemented on the processing platform and considers the delay compensation in the cyber physical system design. Accordingly, we need a processing platform to implement the controller application and compute the computation delay of the target control system. We consider a parametrized processing platform in our design framework to implement the controller application as a system on a chip which will be elaborated in the next section.

II. The Cyber Design Space, $S_{\mathbf{Cyber}}$:

As mentioned in the section 3.0.1, we used Platune as the target system on a chip (SoC) to map the controller algorithm on, and calculate the cyber power consumption and computation delay. The platform architecture that represents the cyber design space includes a MIPS R3000 processor, data and instruction cache, buses, on chip memory, UART and CODEC peripherals. Platune supports 26 parameters and each of these parameters can be assigned a value from its range and configuration space $S_{\mathbf{Cyber}}$ could exceed 1014.

Platune is designed to load C applications, just-in-time compile and link the applications with the required runtime libraries, and execute the applications on the SOC with high degree of precision. We convert the controller algorithm from the Simulink model into C code language to be mapped on the Platune SoC platform with user defined specifications and parameters from the local space $S_{\mathbf{Cyber}}$. Platune carries a MIPS virtual machine to simulate the application software and generate a report on the power consumption and execution time for the specified configuration of the SOC platform that is the resulted $S_{\mathbf{Cyber}}$.

III. The CPS Design Space, $S_{\mathbf{CPS}}$:

The integration of the local design space configurations is carried out with regards to the interactions and interdependence between the local design space parameters P_i and objective functions (e.g., time, power consumption, accuracy, etc.). That is, we need to perform a global analysis on the local design space parameters and the output metrics to prune the global design space accordingly and meet the systems requirements and design constrains. Enforcing the system constraints, C_{CPS}, while integrating the local design spaces, $S_{\text{Cyber}} \times S_{\text{Physical}} \times S_{\text{Control}}$, prunes the global design space S_{CPS} for final system design.

With each set of possible configurations for input parameters, there are a set of evaluation metrics associated. We consider two metrics that we believe would vary during different design space configurations in networked control CPS applications.

- **Energy Consumption:** One important requirement that is imposed on embedded systems is low energy consumption. The total energy consumption E_{total} in embedded cyber physical systems takes into account the energy intake of all the subsystems. We accumulate the cyber energy consumption E_{Cyber} in the target computing platform and physical energy consumption E_{Physical} for the physical model during the simulation time. The cyber energy for the designated simulation time is the cyber energy value for one cycle, as calculated by Platune, multiplied by the number of cycles elapsed during the simulation run. Similarly, the physical energy consumption of the system model is calculated by the Simulink environment for the respective simulation time. The number of cycles in each design configuration is calculated by dividing the simulation time by the respective sample time.

$$E_{\text{total}} = E_{\text{Cyber}} + E_{\text{Physical}} \tag{3.6}$$

- **Integral Square Error (ISR):** Integral square error (ISR) is a control quality measure, that illustrates the deviation from the desired output (expected value). This metric is applied in the control system applications that are intended to filter out large error values instantly. It integrates the square of the system error values over the simulation time. The error values are decided by the difference between the desired output (set point) and the actual output. The target output could be the angle of the pendulum in inverted pendulum example.

$$ISR = \sum (ExpectedValue - ActualValue)^2 \tag{3.7}$$

3.0.4 The Proposed DSE Algorithm

The proposed methodology as illustrated in Algorithm 1 constructs the design space for the framework as outlined in this paper. The algorithm integrates the design spaces from various domains in several interactive pruning phases. This approach, 1) reduces the design space drastically in comparison with the sequential design space construction, and 2) takes into account the interoperability between systems parameters and objectives in different domains

Algorithm 1 Design Space Exploration (DSE)

Require: S_{Physical}, S_{Control}, S_{Cyber}, Threshold,min_energy
Ensure: S_{CPS}, S_{Pareto}
for all $s \in S_{\text{Physical}} \times S_{\text{Control}}$ **do**
    $O_{\text{PC}} \leftarrow$ Simulate_Simulink(s)
    if $O_{PC}.O_{\text{Error}} < Threshold$ **then**
        $\left(S_{\text{Physical}} \times S_{\text{Control}}\right)_{\text{Stable}}$.push($s$)
for all $s' \in \left(S_{\text{Physical}} \times S_{\text{Control}}\right)_{\text{Stable}} \times S_{\text{Cyber}}$ **do**
    $O'_{\text{CPS}} \leftarrow$ Simulate_Platune(s')
    if $O'_{CPS}.O_{\text{ExeTime}} < S_{\text{Control}}.P_{\text{Sampletime}}$ **then**
        S_{CPS}.push(s')
for all $s'' \in S_{CPS}$ **do**
    sorted_list $\leftarrow$ Sort(S_{CPS})
    if $O_{CPS}.O_{\text{Energy}} \leq$ min_energy **then**
        min_energy = $O_{CPS}.O_{\text{Energy}}$
        S_{Pareto}.push(s'')

for design of multi-discipline cyber physical systems.

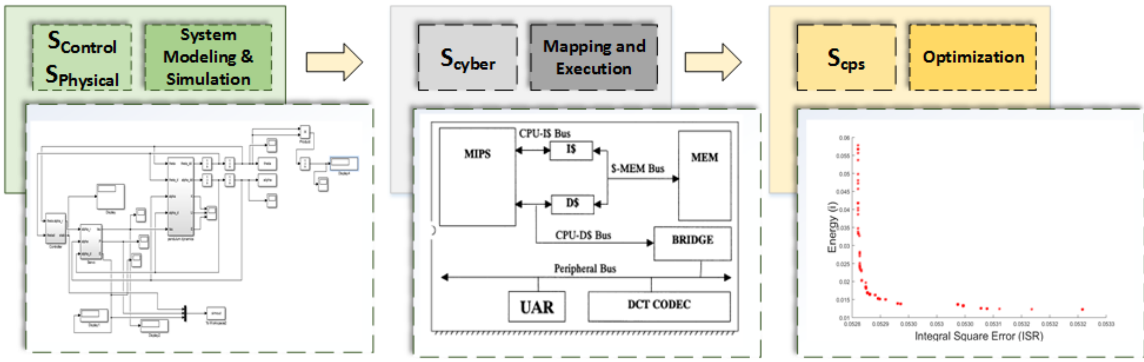

Figure 3.2: The architectural view of the proposed DSE algorithm illustrates the framework in three steps. The first step is represented in the first block that performs system modeling and simulation in the Simulink environment to produce the stable design alternatives. The second block introduces the application mapping on the Platune SoC for simulation, execution and further pruning analysis. The third block is the optimization stage in which a Pareto optimization algorithm is implemented on the final CPS design space to produce the Pareto-optimal design alternatives with regards to the Energy and ISR design objectives.

First, the performance metric (e.g. ISR) in the Simulink model is evaluated for the respective design configurations and stable designs are selected from the $(S_{\text{Physical}} \times S_{\text{Control}})$ design space. Therefore, the first phase of pruning reduces the design space size from $(S_{\text{Physical}} \times S_{\text{Control}})$ to $(S_{\text{Physical}} \times S_{\text{Control}})_{\text{stable}}$ with the respective energy consumptions E_{Physical} and ISR output values which is represented as O_{PC} in Algorithm 1. Next, we integrate the $(S_{\text{Physical}} \times S_{\text{Control}})_{\text{stable}}$ design space with S_{Cyber} for all the possible configurations which, introduces $(S_{\text{Physical}} \times S_{\text{Control}})_{\text{stable}} \times S_{\text{cyber}}$ as the new design space with cyber and physical output metrics (e.g., ISR, cyber energy, physical energy, stability, execution time, etc.). Finally, for each configuration we compare the input parameter, sampling time, with the output metric, execution time, in the entire $(S_{\text{Physical}} \times S_{\text{Control}})_{\text{stable}} \times S_{\text{cyber}}$ design space. This phase of pruning overlaps the control engineers design objectives with software engineers and filters out the design space configurations in which the controller computing delay is larger than the sample time, which results in the final design space as S_{CPS}:

$$S_{\text{CPS}} = \{(S_{\text{Physical}} \times S_{\text{Control}})_{\text{stable}} \times S_{\text{cyber}}\} | C_{\text{CPS}} \tag{3.8}$$

Now the problem should be directed to find a good parameters configuration for the target embedded CPS application in the S_{CPS}design space that optimizes all design objectives, hence multi-objective design optimization problem. Pareto-optimal curves that depict the trade off between the design objectives are solutions to multi-objective optimization problems. The design objective can be a vector of system responses that we are trying to maximize or minimize.

Our intention in this paper is to evaluate and analyze the CPS global design space and to demonstrate that the interaction and interdependency between local design space parameters are not trivial or intuitive and should be accounted for as part of the design decisions. That is, optimal embedded CPS design is in need for a holistic exploration in all the local design space configurations that inhabits the design constraints imposed by the interdependency between the spaces. Therefore, an exhaustive DSE exploration algorithm is afforded to find the Pareto points with regards to the design objective metrics. The pruning steps that integrate the local design spaces S_{Physical}, S_{Control} and S_{Cyber} , into the global design space S_{CPS} is presented in the algorithm. Pareto optimization algorithm is implemented on the global design space S_{CPS} as the third loop, to deliver the Pareto-optimal configurations in the respective embedded cyber physical system design. The architectural view of the proposed algorithm to create the design space for networked control cyber physical systems is illustrated in Figure 3.2.

Chapter 4

Experimental Results

Here, a Design Space Exploration example for the design of a real application of automated networked control system for a state space inverted pendulum model with full state feedback controller has been developed. A comprehensive description of the inverted pendulum control example is outlined in [3]. According to the proposed algorithm that constructs a holistic design space for networked control cyber physical systems, the global design space S_{CPS} for the respective example is as follows:

$$\{(S_{\text{pendulum}} \times S_{\text{controller}})_{\text{stable}} \times S_{\text{platform}}\}|C_{\text{CPS}} \tag{4.1}$$

The local design space parameters that we handpicked to vary for different design configurations are illustrated in Table 1. These configurations are integrated through different filter levels described in section 3. The number of configurations for the integrated design space after each pruning phase is depicted in Table 2. As illustrated in the table the design space for the holistic networked control inverted pendulum example is reduced (compared to the exhaustive search) from 82 million design points to 95316.

Our experiments were performed on a PC with an Intel Core i5 Quad processor running on a 64 bit Windows 7 operating system. First, the networked control system for the inverted pendulum example is modeled in the Simulink from MATLAB R2015a environment employing the Simulink libraries and system blocks (e.g. controller block, function block, etc.). A Dormand-Prince solver with variable step size is selected to perform the simulation of the model per design configuration. Model simulation is iterated for 2252 set of design alternatives in the $(S_{\text{Physical}} \times S_{\text{Control}})$ design space with 30 second simulation time per iteration. The sample time of the controller as a control parameter and the length of the inverted pendulum as a physical parameter are selected to vary between different design points.

Then, the algorithm selects 655 sets of configurations with stable behavior, that is $(S_{\text{Physical}} \times S_{\text{Control}})_{\text{stable}}$ design space, to be integrated with the computing platform design space, S_{Cyber} . The controller C program is loaded in Platune to be mapped and simulated on the configured computing platform. The cyber design space S_{Cyber} for the computing platform provided by the Platune SoC could exceed 10^{14} . In our experiment for the inverted pendulum example we used 32400 sets of design points to represent our S_{Cyber} design space. That is, the controller application is executed on the computing platform from the Platune framework for 113975 design alternatives in the $(S_{\text{Physical}} \times S_{\text{Control}})_{\text{stable}} \times S_{\text{cyber}}$ design space. The computing platform architecture alters for cache size and CPU speed parameter alternatives. Finally, the algorithm applies the final design constraint C_{CPS} to satisfy the control and software engineers requirements in design decisions. That is, the algorithm prunes the configurations in which the sampling time in the Simulink model is larger than computation delay obtained from Platune simulations which introduces 95316 design points for our final design space S_{CPS}. The proposed pruning algorithm has the following advantages:

1. Scales the search space from 83 million design points in the sequential design space, $S_{\text{Cyber}} \times S_{\text{Physical}} \times S_{\text{Control}}$, to 95316 design points in the S_{CPS} design space prior to Pareto optimal configuration selection.

2. The filtered design space, S_{CPS}. , is more appropriate to meet the design decisions of control engineers and software engineers in an interactive manner.

The exploration algorithm to find the Pareto optimal points is implemented for 95316 sets of configurations in the global design space of the networked control inverted pendulum application S_{CPS}. As mentioned before, the objective metrics in our multi-objective optimization problem are the total energy consumption E_{Total} and the control quality ISR. Figure 4.1 depicts the global design space points S_{CPS} with gray dots and Pareto curve as the result of the Pareto optimization algorithm with red dots. The curve represents the trade offs between the energy consumption and ISR values for all the design points in the global design space configurations S_{CPS}.

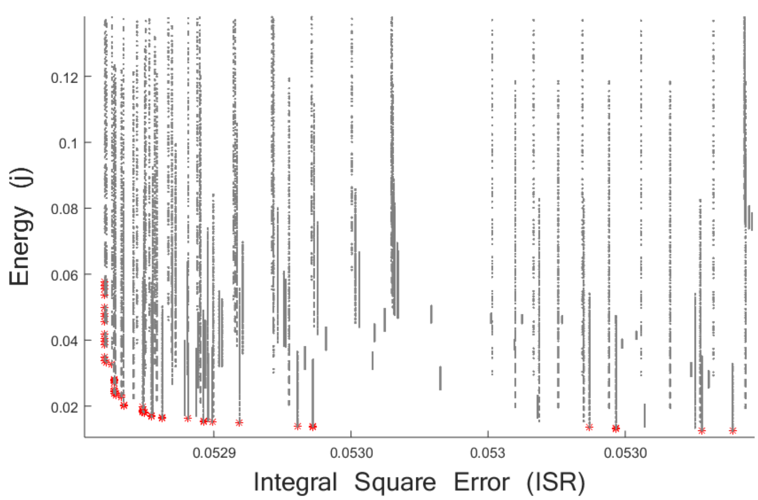

Figure 4.1: Pareto Optimal points, the red stars, residing in the design points in the S_{CPS} design space, the grey points, represents the trade-off between the energy and ISR.

The trade off between the integral square error and total energy consumption is more prominent for low power numbers in the range of 0.015-0.025 joule. Trade-off information is

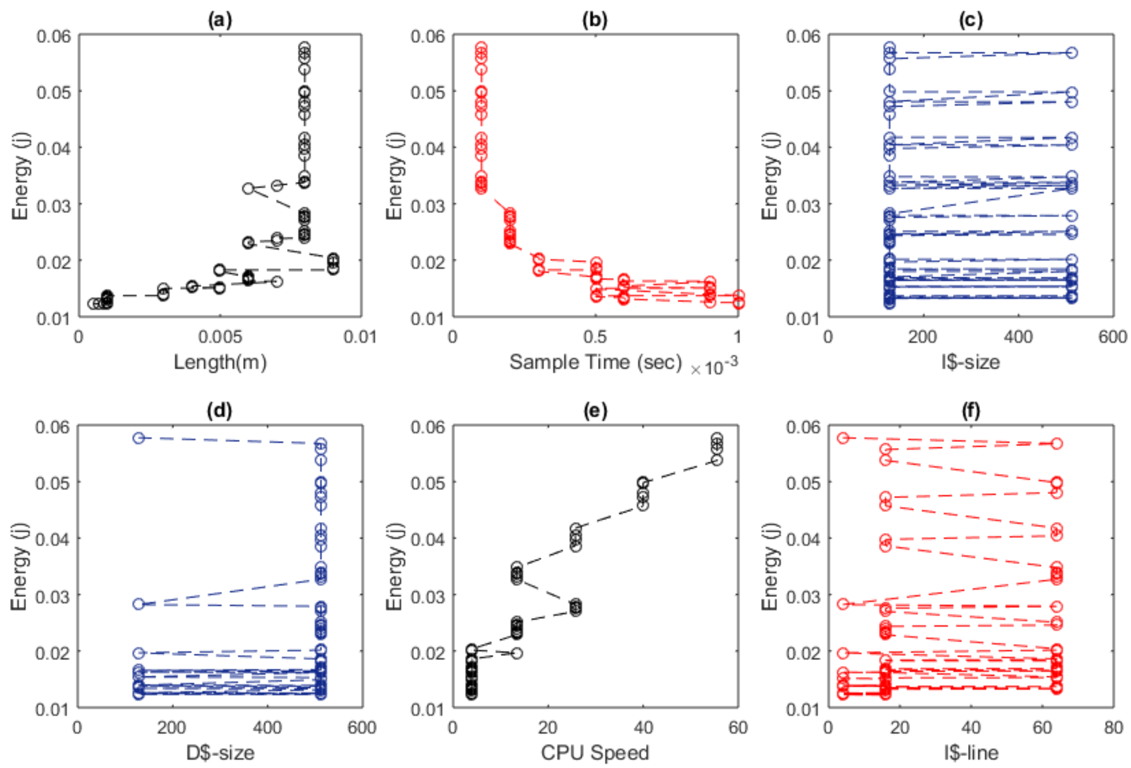

Figure 4.2: The plots illustrate the behavior of design parameters for the cyber physical system example, inverted pendulum with regards to the total Energy metric. These plots include the parameters from physical (a), control (b) and cyber (c, d, e, f) design spaces. The zigzag patterns of the plots represent a non linear behavior for successive design points. For instance, plot (f) depicts the relation between the instruction cache line and total energy values. It can be observed that for two successive Pareto-optimal design points with total Energy between (0.03-0.035), we observe an increase in the corresponding cache line values. On the other hand, in the (0.035-0.04) range for total energy, for two successive Pareto points we observe a decrease in the cache line values.

extremely important to identify the most preferred point along the Pareto-curve. It is evident that the application purpose decides the superior Pareto optimal point which satisfies the design objective accordingly. For example Robotic surgical procedures [21] afford more power consumption in trade off for the best accuracy and system reliability. On the other hand, embedded mobile applications are desperate for low power design decisions.

Figure 4.2 illustrates the variations in cyber, physical and controller parameters for energy consumption in the Pareto-optimal configurations. As it is pronounced by the plots, the respective parameters do not follow a steady pattern in trade off with the energy consumption metric. The swinging behavior of the parameters before the energy consumption values in the Pareto-optimal configurations, accentuates the need for DSE methodologies and frameworks, that perform a holistic and interactive evaluation on design alternatives for embedded cyber physical systems design. For example, the non-intuitive interaction between the local design space parameter, the length of the pendulum, and the design objective metric, total energy consumption, can be interpreted from the Length-Total Energy diagram. Plot (a) in Figure 4.2 discloses that increasing the total energy consumption measurements between the range of 0.018-0.021 joule corresponds with an increase in the respective length values. On the other hand, the increase in the Pareto- optimal energy consumption measurements in the 0.021- 0.024 range observes a decrease in the corresponding length values. This non-trivial correlation between the system level design parameters and metrics plays a significant role in design decisions. That is, we are claiming that embedded CPS designers should take into account not only the trade off between the design objectives to pick the most preferred Pareto points, but also the interdependency between the design space parameters (cyber parameters, physical parameters, control parameters, etc.) and the design metrics (ISR, Energy, Power, etc.). Accordingly, noticing the interplay between the design parameters and design metrics includes the hardware, inventory and technology constraints in the design decisions. For example, maybe one Pareto point is preferred due to the inventory availability for the corresponding physical parameter. Consequently, the engineering of cyber physical systems inherently demands collaboration between diverse domains and a holistic computer-aided design framework is imperative to automate the costly design task, the design space exploration of alternative solutions.

Table 4.1: Design Parameters in S_{Control}, S_{Physical} and S_{Cyber} Design Spaces horizon.

Space	S_{Control}	S_{Physical}	S_{Cyber}
Parameters	$P_{\text{Sample Time}}$	P_{Length}	$\{P_{\text{i\$-size}}$, $P_{\text{i\$-line}}$, $P_{\text{i\$-associate}}$, $P_{\text{d\$-size}}$, $P_{\text{d\$-line}}$, $P_{\text{d\$-associte}}$, $P_{\text{CPU Speed}}\}$
Tuples	$P_{\text{Sample Time}}$ =(S_{Control}, Sample Time, [1e-4:28])	P_{Length} =($S_{\text{inverted pendulum}}$, Length, [1e-4:20])	$P_{\text{i\$-size}}$=($S_{\text{platform}}$, i$-size, [128-32k]) $P_{\text{i\$-line}}$=($S_{\text{platform}}$, i$-line, [4:64]) $P_{\text{i\$-associate}}$=($S_{\text{platform}}$, i$-associate, [1:16) $P_{\text{d\$-size}}$=($S_{\text{platform}}$, d$-size, [128:32k]) $P_{\text{d\$-line}}$=($S_{\text{platform}}$, d$-line, [4:64]) $P_{\text{d\$-associte}}$=($S_{\text{platform}}$, d$-associate, [1:16]) $P_{\text{CPU Speed}}$=(S_{platform}, CPU Speed, [4:256])

Table 4.2: Number of Configurations in the Design Space

Design Space	# of Configurations
$S_{\text{Physical}} \times S_{\text{Control}}$	2552
S_{Cyber}	32400
$S_{\text{Physical}} \times S_{\text{Control}} \times S_{\text{Cyber}}$	82,648,800
$\left(S_{\text{Physical}} \times S_{\text{Control}}\right)_{\text{Stable}}$	655
$\left(S_{\text{Physical}} \times S_{\text{Control}}\right)_{\text{Stable}} \times S_{\text{Cyber}}$	113,975
S_{CPS}	95316

Chapter 5

Conclusion and Future Work

In this paper a holistic and interactive Design Space Exploration framework is presented for use in the design of cyber physical systems. We first discussed a simulation and exploration tool that combines state-of-the-art exploration of SoC parameters with a model-based simulation tools such as Simulink. As a result, we are able to analyze the impact of cyber design decisions, such as voltage, processor configurations, or cache sizes on the overall CPS performance.

We applied the proposed framework in the design of a real networked control application to verify the efficiency and usefulness of the proposed work. Our experimental results confirm that sophisticated frameworks are needed for the design of cyber physical systems due to the non-linear behavior of the Pareto optimal design points. In our future work, we plan to apply methodologies to automatically mine the interdependencies between the parameters from different design spaces in the cyber-physical systems and employ the correlations to present heuristics for efficient design space exploration.

Bibliography

[1] D. Andrews, A. Kana, J. Hopman, and J. Romanoff. State of the art report on design methodology. In *Marine Design XIII, Volume 1*, pages 3–16. CRC Press, 2018.

[2] E. Bartocci, J. Deshmukh, A. Donzé, G. Fainekos, O. Maler, D. Ničković, and S. Sankaranarayanan. Specification-based monitoring of cyber-physical systems: a survey on theory, tools and applications. In *Lectures on Runtime Verification*, pages 135–175. Springer, 2018.

[3] H. M. Buini, S. Peter, and T. Givargis. Including variability of physical models into the design automation of cyber-physical systems. In *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*, pages 1–6. IEEE, 2015.

[4] A. Canedo, E. Schwarzenbach, A. Faruque, and M. Abdullah. Context-sensitive synthesis of executable functional models of cyber-physical systems. In *Cyber-Physical Systems (ICCPS), 2013 ACM/IEEE International Conference on*, pages 99–108. IEEE, 2013.

[5] R. Chamberlain. *Cyber Physical Systems. Design, Modeling, and Evaluation*. Springer, 2019.

[6] M. H. Davis. *Markov models & optimization*. Routledge, 2018.

[7] C. Gamble and K. Pierce. Design space exploration for embedded systems using co-simulation. In *Collaborative Design for Embedded Systems*, pages 199–222. Springer, 2014.

[8] A. Gerstlauer, C. Haubelt, A. D. Pimentel, T. P. Stefanov, D. D. Gajski, and J. Teich. Electronic system-level synthesis methodologies. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 28(10):1517–1530, 2009.

[9] T. Givargis and F. Vahid. Platune: a tuning framework for system-on-a-chip platforms. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 21(11):1317–1327, 2002.

[10] A. Knight. *Basics of MATLAB and Beyond*. Chapman and Hall/CRC, 2019.

[11] P. G. Larsen, C. Gamble, K. Pierce, A. Ribeiro, and K. Lausdahl. Support for co-modelling and co-simulation: The crescendo tool. In *Collaborative Design for Embedded Systems*, pages 97–114. Springer, 2014.

[12] W. S. Levine. *Control system fundamentals*. CRC press, 1999.

[13] T. LITMAN. Autonomous vehicle implementation predictions: Implications for transport planning.[sl], 01 2014. 36-42 p.

[14] C. Lv, X. Hu, A. Sangiovanni-Vincentelli, Y. Li, C. M. Martinez, and D. Cao. Driving-style-based codesign optimization of an automated electric vehicle: a cyber-physical system approach. *IEEE Transactions on Industrial Electronics*, 66(4):2965–2975, 2018.

[15] M. Maasoumy, Q. Zhu, C. Li, F. Meggers, and A. Vincentelli. Co-design of control algorithm and embedded platform for building hvac systems. In *Cyber-Physical Systems (ICCPS), 2013 ACM/IEEE International Conference on*, pages 61–70. IEEE, 2013.

[16] J. Michniewicz and G. Reinhart. Cyber-physical robotics–automated analysis, programming and configuration of robot cells based on cyber-physical-systems. *Procedia Technology*, 15:566–575, 2014.

[17] T. Mück, A. A. Fröhlich, G. Gracioli, A. M. Rahmani, J. G. Reis, and N. Dutt. Chips-ahoy: a predictable holistic cyber-physical hypervisor for mpsocs. In *Proceedings of the 18th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation*, pages 73–80. ACM, 2018.

[18] N. Mühleis, M. Glaß, L. Zhang, and J. Teich. A co-simulation approach for control performance analysis during design space exploration of cyber-physical systems. *ACM SIGBED Review*, 8(2):23–26, 2011.

[19] H. Neema, Z. Lattmann, P. Meijer, J. Klingler, S. Neema, T. Bapty, J. Sztipanovits, and G. Karsai. Design space exploration and manipulation for cyber physical systems. In *IFIP First International Workshop on Design Space Exploration of Cyber-Physical Systems (IDEAL'2014), Springer-Verlag Berlin Heidelberg*, 2014.

[20] M. O'Kelly, V. Sukhil, H. Abbas, J. Harkins, C. Kao, Y. V. Pant, R. Mangharam, D. Agarwal, M. Behl, P. Burgio, et al. F1/10: An open-source autonomous cyber-physical platform. *arXiv preprint arXiv:1901.08567*, 2019.

[21] T. Ortmaier, M. Gröger, D. H. Boehm, V. Falk, and G. Hirzinger. Motion estimation in beating heart surgery. *Biomedical Engineering, IEEE Transactions on*, 52(10):1729–1740, 2005.

[22] J. Otto and O. Niggemann. Automatic parameterization of automation software for plug-and-produce. In *The AAAI-15 Workshop on Algorithm Configuration (AlgoConf 2015), Austin, Texas*, 2015.

[23] K. Vatanparvar and M. A. Al Faruque. Otem: Optimized thermal and energy management for hybrid electrical energy storage in electric vehicles.

[24] K. Vatanparvar and M. A. Al Faruque. Control-as-a-service in cyber-physical energy systems over fog computing. In *Fog Computing in the Internet of Things*, pages 123–144. Springer, 2018.