Compositional Design of Autonomous Systems: From Hardware Selection to Decision Making

Gioele Zardini [*,**]

[*] *Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305 USA*
[**] *Laboratory for Information & Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, USA (e-mail: gzardini@mit.edu).*

Abstract: When designing autonomous systems, we need to consider multiple trade-offs at various abstraction levels, and choices of single (hardware and software) components need to be studied jointly. For instance, the design of future mobility solutions (e.g., autonomous vehicles) and the design of the mobility systems they enable are closely coupled. Indeed, knowledge about the intended service of novel mobility solutions would impact their design and deployment process, whilst insights about their technological development could significantly affect transportation policies.
Co-designing autonomous systems is a complex task for at least two reasons. First, the co-design of interconnected systems (e.g., networks of cyber-physical systems) involves the simultaneous choice of components arising from heterogeneous fields, while satisfying systemic constraints and accounting for multiple objectives. Second, components are connected via interactions between different stakeholders. I will present a framework to co-design such systems, leveraging a monotone theory of co-design. The framework will be instantiated in applications in mobility and autonomy. Through various case studies, I will show how the proposed approaches allow one to efficiently answer heterogeneous questions, unifying different modeling techniques and promoting interdisciplinarity, modularity, and compositionality. I will then discuss open challenges for compositional systems design optimization.

Keywords: Robotics; Large Scale Complex Systems; Transportation Systems; Intelligent Autonomous Vehicles; Systems and Control for Societal Impact; Networked Systems

1. INTRODUCTION

The proper study of manking is the science of design.
— Herbert A. Simon.

The design and operation of *complex systems* stands out as one of the paramount challenges of this century. Such systems are labeled as complex not only due to the intricacies of their individual components, but also because their functioning hinges on complex *interactions* among these components. To give a sense of the kind of systems we are interested in, think about the complex circuit governing a sensor employed in autonomous driving contexts, an autonomous vehicle which leverages the sensor, as well as a number of other complex hardware and software components within the autonomy stack, a fleet of autonomous vehicles of this kind, deployed following certain principles, and interacting via complex patterns, and a complex mobility system leveraging autonomous mobility-on-demand (i.e., the fleet) systems as well as standard transit options. Each of these systems is complex per se, and is influenced and influences other ones at different scales.

How do we assess the impact of local design decisions at the system level? How can we formulate, and automatically solve co-design problems involving such complex systems? Traditionally, the design optimization of selected components is treated in a compartmentalized manner, blocking the collaboration of multiple designers, and modularity.

In the context of autonomous systems, existing techniques do not allow one to simultaneously consider the specificity and formality of technical results for selected disciplines (e.g., decision making and perception), and more practical trade-offs related to energy consumption, computational efforts, performance, and monetary costs.

This matter is discussed in my dissertation (Zardini (2023)), and references therein.

Desiderata and challenges

To deal with the above, one needs a comprehensive task-driven co-design automation theory, allowing multiple domains to interact, clearly specifying components, and their interactions at the system level. In particular, a successful framework should achieve the following desiderata.

Formal Complex systems consist of diverse components, and the abstraction one chooses for the design exercise must transcend particular domains. At the same time, to be tangible, the abstraction must be mathematically precise, avoiding vague statements about the problem at hand. Furthermore, we typically want to characterize all the objectives of the design problem, without sacrifices.

Compositional First, we have to account for *horizontal* composition. This refers to the interconnections and interactions of different components and their configuration. Indeed, in most cases, choices that are made at the level of components without looking at the entire system are doomed to be suboptimal. Second, we have to consider *vertical* (i.e., *hierarchical*) composition. This refers to the principle "your system is just a component in someone else's system".

Collaborative There are two types of collaboration. First, there is a collaboration between human and machine, in the definition and solution of co-design problems. Second, and most importantly, is the collaboration among different "experts" or teams in the design process.

Computationally tractable One needs to be able to compute solutions of the design problem efficiently. Therefore, we strive to create not only a qualitative modeling framework for co-design, but also a formal and quantitative description that will be suitable for setting up an optimization problem which can be solved to obtain an optimal design.

Continuous Rather than viewing designs as a single decision made at one point in time, one must see them as continuously evolving entities. The designer should be able to smoothly characterize this evolution within a framework of co-design.

Manipulable Not only we want the designer to be able to specify models for design problems, and to do that over time, but we also want the whole problem manipulation process to be smooth. For instance, we might need to ignore certain objectives, ask different questions given the same co-design architecture, etc.

Intellectually tractable The design process should not be limited to system architects and specialists. Rather, it should be collaborative. Sometimes, when developing design optimization tools, one confuses the *developer's* and the *user's* viewpoints. While we want the chosen formalism to possess the above properties, we also want stakeholders to take an active role. This sets the need for a simple, cross-domain user interface.

2. A MONOTONE THEORY OF CO-DESIGN

This section is a quick summary of the main concepts related to the monotone theory of co-design, presented in great detail in Censi (2015); Censi et al. (2024); Zardini (2023), with insights into the developer viewpoint. The reader is assumed to be familiar with basic concepts of order theory (a good source is Davey and Priestley (2002)).

2.1 Formulating co-design problems

The monotone theory of co-design is based on the atomic notion of a monotone design problem with implementation (MDPI).

Definition 1. Given partially ordered sets (posets) $\mathcal{F}, \mathcal{R}$, (functionalities and resources), we define a *MDPI* as a tuple $\langle \mathcal{I}_d, \mathsf{prov}, \mathsf{reqs} \rangle$, where $\mathcal{I}_d$ is the set of implementations, and prov, reqs are maps from $\mathcal{I}_d$ to $\mathcal{F}$ and $\mathcal{R}$, respectively:

$$\mathcal{F} \xleftarrow{\mathsf{prov}} \mathcal{I}_d \xrightarrow{\mathsf{reqs}} \mathcal{R}.$$

We compactly denote the MDPI as $d\colon \mathcal{F} \nrightarrow \mathcal{R}$. Furthermore, to each MDPI we associate a monotone map $\bar{d}$, given by:

$$\begin{aligned} \bar{d}\colon \mathcal{F}^{\mathrm{op}} \times \mathcal{R} &\to \langle \mathcal{P}(\mathcal{I}_d), \subseteq \rangle \\ \langle f^*, r \rangle &\mapsto \{ i \in \mathcal{I}_d \colon (\mathsf{prov}(i) \succeq_{\mathcal{F}} f) \wedge (\mathsf{reqs}(i) \preceq_{\mathcal{R}} r) \}, \end{aligned}$$

where $(\cdot)^{\mathrm{op}}$ reverses the order of a poset. The expression $\bar{d}(f^*, r)$ returns the set of implementations (design choices) $S \subseteq \mathcal{I}_d$ for which functionalities f are feasible with resources r. We represent a MDPI in diagrammatic form as a block with green wires on the left for functionalities, and dashed red ones on the right for resources.

Remark 2. (Monotonicity). Consider a MDPI for which we know $\bar{d}(f^*, r) = S$.

- $f' \preceq_{\mathcal{F}} f \Rightarrow \bar{d}(f'^*, r) = S' \supseteq S$. Intuitively, decreasing the desired functionalities will not increase the required resources;
- $r' \succeq_{\mathcal{R}} r \Rightarrow \bar{d}(f^*, r') = S'' \supseteq S$. Intuitively, increasing the available resources cannot decrease the provided functionalities.

For related examples and detailed explanations we refer to our draft book and my dissertation (Censi et al. (2024); Zardini (2023)).

Remark 3. (Populating models). In practical cases, one can populate the feasibility relations of MDPIs with analytic relations (e.g. cost functions, precise relationships), numerical analysis of closed-form relations (e.g., optimal control problems), and in a data-driven fashion (e.g., via POMDPs, simulations, or by solving instances of optimization problems). For detailed examples refer to Zardini (2023); Zardini et al. (2021a,b, 2022).

Individual MDPIs can be composed in several ways to form a co-design problem (a multigraph of MDPIs), allowing one to decompose a large problem into smaller subproblems, and to interconnect them. An exhaustive list of compositions is provided in Censi et al. (2024); Zardini (2023). Series composition happens when a functionality of a MDPI is required by another MDPI (e.g., the power provided by a battery is needed by an electric motor to produce torque). The symbol "$\preceq$" is the posetal relation, which represents a co-design constraint: the resource one problem requires, cannot exceed the functionality another problem provides. Parallel composition formalizes decoupled processes happening together, and loop composition describes feedback.[1] Notably, MDPIs are closed under compositions (i.e., a composition of MDPIs is an MDPI).

2.2 Solving co-design problems

Definition 4. Given a MDPI d, we define monotone maps

- $h_d\colon \mathcal{F} \to \mathsf{A}\mathcal{R}$, mapping a functionality to the *minimum* antichain of resources providing it;

[1] The formalization of feedback makes the category of MDPIs a traced monoidal category Zardini (2023).

- $h'_d \colon \mathcal{R} \to \mathsf{A}\mathcal{F}$, mapping a resource to the *maximum* antichain of functionalities provided by it.

Solving MDPIs requires finding such maps. If such maps are Scott continuous, and posets involved are complete, one can rely on Kleene's fixed point theorem to design an algorithm solving the queries "fix a functionality and find minimum resources to achieve it" and "fix a resource and find maximum functionalities that can be achieved" (and the related design choices).

The resulting algorithm is guaranteed to converge to the set of optimal solutions, or to provide a certificate of infeasibility. Furthermore, the complexity of solving such problems is only linear in the number of options available for each component (as opposed to combinatorial). For more details, see Zardini (2023).

2.3 Compositional perspective

The modeling and algorithmic results presented in the previous subsections can be elegantly summarized by defining locally posetal, traced monoidal categories of design problems, of solutions (i.e., maps of the form of Definition 4). The solution algorithm emerges from a functor between the two categories (i.e., the "solution of composition of MDPIs is the composition of solutions of single MDPIs").

2.4 From autonomy to future mobility

The presented framework has been applied to various problems in automotive, autonomy, and transportation. Here, we report an exemplary application all the way from autonomy to future mobility (Figure 1). Starting from a model of an intermodal mobility system, one builds the interconnected co-design diagram, populates it via models of diverse nature (e.g., catalogue-based, via first principles, and data-driven) and efficiently solves it finding insightful trade-offs for policy makers (Zardini et al. (2023)). For instance, we are able to assess the trade-offs of average travel time in a city, and total investment costs for the municipality, characterizing each solution by the chosen technologies and fleet sizes for the different mobility options. In a similar way, one can model the task-driven co-design of a single autonomous vehicle, and characterize trade-offs of task complexity and monetary and power requirements. Thanks to the compositionality properties of the approach, one can now model the task-driven co-design of a single autonomous vehicle in a fleet and solve it, obtaining insights all the way from the platform level (e.g., controllers, sensors, algorithms) to the system level (e.g., policies of a municipality).

3. OUTLOOK

In this extended abstract, we compactly presented a new class of tools to model and solve complex system design optimization problems. This class of tools is new, and sets the stage for several directions for future research.

Co-design Games The presented framework is naturally *collaborative* and *decentralized*, and allows one to consider multiple (often conflicting) objectives. However, it is essential to acknowledge a fundamental assumption that underpinned our previous applications: the presumption that while multiple designers may model different components within a complex architecture, they all share a common interest in minimizing a predefined set of objectives. This is suitable for a specific subset of engineering design problem (e.g., designing an autonomous vehicle). However, it necessitates further developments to explicitly account for *strategic design interactions* (e.g., when designing a multi-stakeholder mobility system). In short, we should develop a theory of co-design games. A starting point was provided via posetal games in Zanardi et al. (2021).

Spatio-temporal resources So far, we have not explicitly attached a time-related aspect to resources and functionalities. Interestingly, this would extend the notion of optimization query as well. For instance, one could be interested in the minimal sequence of resources, such that a certain functionality is achieved by a certain time instant.

Computation-aware Solution Schemes Up to this point, the approach involved creating a specific co-design diagram, determining how to populate each individual design problem within it, and then addressing a particular query. In the future, we would be interested in conducting this process with a computation-aware approach, designing algorithms which autonomously determine which model to sample based on the existing partial solution.

REFERENCES

Censi, A. (2015). A mathematical theory of co-design. *arXiv preprint arXiv:1512.08055*.

Censi, A., Lorand, J., and Zardini, G. (2024). *Applied Compositional Thinking for Engineering*. URL `https://bit.ly/3qQNrdR`. Work in progress book.

Davey, B.A. and Priestley, H.A. (2002). *Introduction to lattices and order*. Cambridge university press.

Zanardi, A., Zardini, G., Srinivasan, S., Bolognani, S., Censi, A., Dörfler, F., and Frazzoli, E. (2021). Posetal games: Efficiency, existence, and refinement of equilibria in games with prioritized metrics. *IEEE Robotics and Automation Letters*, 7(2), 1292–1299.

Zardini, G. (2023). *Co-design of complex systems: From autonomy to future mobility systems*. Ph.D. thesis, ETH Zurich.

Zardini, G., Censi, A., and Frazzoli, E. (2021a). Co-design of autonomous systems: From hardware selection to control synthesis. In *2021 European Control Conference (ECC)*, 682–689. doi: 10.23919/ECC54610.2021.9654960.

Zardini, G., Lanzetti, N., Censi, A., Frazzoli, E., and Pavone, M. (2023). Co-design to enable user-friendly tools to assess the impact of future mobility solutions. *IEEE Transactions on Network Science and Engineering*, 10(2), 827–844.

Zardini, G., Milojevic, D., Censi, A., and Frazzoli, E. (2021b). Co-design of embodied intelligence: A structured approach. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7536–7543. doi:10.1109/IROS51168.2021.9636513.

Zardini, G., Suter, Z., Censi, A., and Frazzoli, E. (2022). Task-driven modular co-design of vehicle control systems. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2196–2203. IEEE.

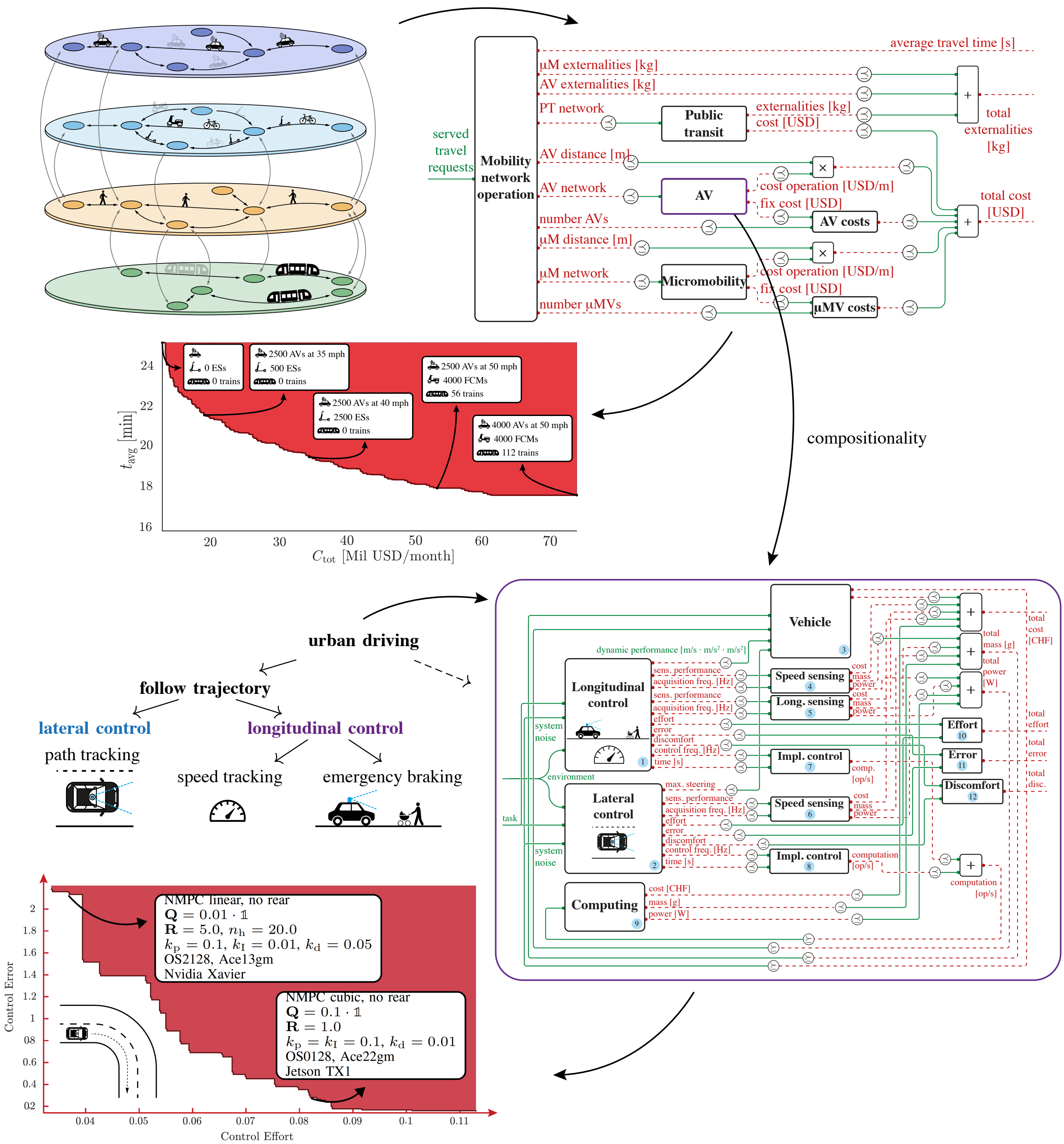

Fig. 1. Exemplary application of the co-design toolbox, from the co-design of an autonomous vehicle, to the co-design of an urban mobility system.