



**HAL**  
open science

## Co-simulation-based optimization: state of the art

Diego Alejandro Vega, Vincent Chevrier

► **To cite this version:**

Diego Alejandro Vega, Vincent Chevrier. Co-simulation-based optimization: state of the art. 2024.  
hal-04668193

**HAL Id: hal-04668193**

**<https://inria.hal.science/hal-04668193v1>**

Preprint submitted on 7 Aug 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0  
International License

# CO-SIMULATION-BASED OPTIMIZATION: STATE OF THE ART

DIEGO ALEJANDRO VEGA AND VINCENT CHEVRIER

ABSTRACT. A complex big-picture problem can be studied by combining the knowledge of various disciplines' experts through interdisciplinary collaboration. This collaboration carries a considerable challenge, as it needs to handle the interaction of domains without losing their capabilities. One solution for this problem is co-simulation, which creates a virtual representation that can be tested, validated, and improved. Co-simulation provides the chance to get high-level insights into the complex system; optimization takes advantage of this opportunity by exploring the various system configurations to determine the best performance according to a predefined objective.

Numerous fields have already employed this combination of co-simulation and optimization in the literature to study and enhance the performance of complex systems. This survey compiles, assesses, and links existing papers on co-simulation-based optimization to give foundations and guidelines for future research on this combined approach.

An overview of co-simulation-based optimization from the conceptual, multidisciplinary, and optimization perspectives is presented. To provide general descriptions of the integration process of co-simulation-based optimization, this overview discusses the trends in the literature, which primarily involves a wide range of domains, numerous optimization methods, diverse strategies used to handle interoperability between co-simulation and optimization, and the benefits of generalization tools and frameworks.

## CONTENTS

1. Introduction	2
1.1. Motivation	2
1.2. Need for the survey	3
2. Co-simulation and optimization	5
2.1. Main concepts: Co-simulation	5
2.2. Main concepts: Optimization	9
2.3. Co-simulation-based optimization: black box approach	9
2.4. Challenges and conclusions	11
3. Multidisciplinary Analysis	11
3.1. Simulation domains	12
3.2. Simulations interoperability	13
3.3. Challenges and conclusions	15
4. Optimization analysis	17
4.1. Optimization methods in the literature	17
4.2. Optimization Software and frameworks	20
4.3. Challenges and conclusion	22

---

*Date:* August 5, 2024.

*Key words and phrases.* Co-simulation, Coupled-simulation, Optimization, Black-box approach.

5. Discussion and conclusions	23
References	24
Appendix A. Classification table	34
Appendix B. Abbreviations	37
Appendix C. Software compilation	38

## 1. INTRODUCTION

**1.1. Motivation.** The growing presence of product complexity in the industry is expanding the need to understand and predict the global behavior of a complex system in a reliable way. Co-simulation is presented as one way of modeling this complexity throughout the virtual representation of heterogeneous entities interacting with each other at a local level, where as a consequence, the global behavior emerges [29].

The emerging behavior achieved with the co-simulation can be tested using different setup configurations to understand or improve certain performances at a low cost compared to real setup experiments. Introducing optimization as the next step of co-simulation comes naturally, as in the field of simulation, one of the ultimate goals is to find a combination of parameters that optimizes a key output performance [114]. Given the wide amount of optimization methodologies available in the literature, it is necessary to analyze the methods often used to understand how to choose a particular optimization methodology correctly, this means taking advantage of the optimization toolbox of algorithms.

In this article, we present a survey on the combination of co-simulation and optimization with the purpose of understanding how this combined approach is being implemented, as well as highlighting the recurrent strategies, structures, and patterns that are found in the literature's implementations of co-simulation-based optimization. The sample of publications used in this survey is obtained using the Equation 1 on three different Scientific databases (SCOPUS, Web of Science, and DBLP), which resulted in a total of 126 publications between journal articles, book chapters, and conference proceedings. This sample is studied and reduced to 96 after excluding publications that are not part of the co-simulation-based optimization field. The goal of this publication sample database is to provide a representative notion of the work being done in research with co-simulation-based optimization until the end of 2023 and use this representative sample to analyze how this process is being done and, what challenges remain to be fixed in the field.

(1) TITLE (("co-simulation" OR "cosimulation" OR "coupled simulation" ) AND "optimization" )

We perform a preliminary analysis of the sample database to observe the growing relevance over time. Figure 1 shows the publishing timeline. It illustrates the initial studies conducted on the subject between 2007 and 2009, followed by an increase in publications over time. The problem has been more relevant during the last ten years, particularly in the last three years when the number has remained around 13. This trend demonstrates the value of the survey's findings for further research on the subject and the need to combine all of the individual efforts, particularly concerning methodology and standard operating procedures.

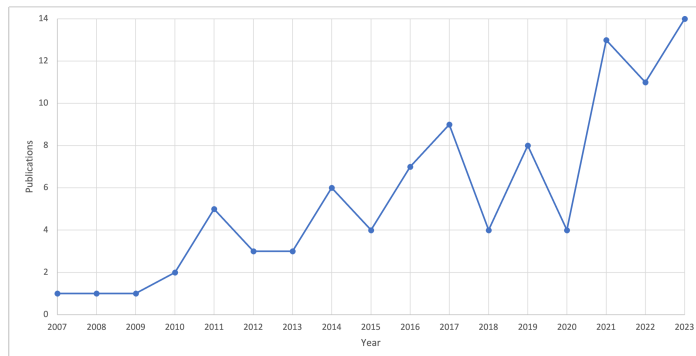


FIGURE 1. Publications about co-simulation-based optimization over the years until 2023. Combined results from SCOPUS, Web of Science, and DBLP using Equation 1 on February 2024.

The domain distribution of the publications is another analysis that can provide general insights into the literature. We take Table 1 from the SCOPUS analysis tool, which shows the number of publications that involve each of the general domains as defined in the SCOPUS indexing (a publication may involve more than one domain). The table shows that the primary fields involved are engineering and computer science, and the applications are diverse, spanning from physics and mathematics to the natural and social sciences. Because of the wide range of applications throughout the research spectrum, co-simulation-based optimization methodology has the potential to benefit a wide range of areas. Given the multidisciplinary nature of the methodology examined in this survey, a more thorough examination of the domains and methods by which this technique is currently being applied is necessary.

**1.2. Need for the survey.** The co-simulation survey done by Gomes, et al. [58] showed the growing interest and the advantages of co-simulation by bridging individual contributions, which improves future applications and leads to a deeper understanding of co-simulation. The same situation is happening currently on co-simulation-based optimization as a technique, there is a growing interest represented in the many individual contributions that are not linked yet in a survey. Despite the growing interest, to the best of our knowledge, there is no survey focused on co-simulation-based optimization, which is causing several researchers to propose already existing approaches unknowingly. One example of this is found in the reviewed articles sample, where papers such as [213, 17, 70, 187, 141] talk about co-simulation, other papers such as [180, 203, 73, 205, 212] talk about coupled simulations, and finally the paper [5] talks of cosimulation, when the three concepts are referring to the same technique.

To address the problem of multiple terminologies for one technique, the term “co-simulation” was proposed as a unifying term in the Gomes et al. survey [58]. An example of the advantages of consistent nomenclature in the context of this paper (co-simulation-based optimization) is the COSMO methodology, which was first introduced in the publication [207] and then used in [208, 209, 210, 206]. This type of uniformity supports the development and validation of a methodology’s

TABLE 1. Number of publications per research field, using the field categories as defined in SCOPUS. Results from SCOPUS after manually removing papers out of the scope of this survey from equation 1 on February 2024, with a total of 85 publications.

Field	Articles
Engineering	53
Computer Science	34
Mathematics	13
Physics and Astronomy	12
Environmental Science	11
Materials Science	10
Energy	10
Agricultural and Biological Sciences	3
Chemical Engineering	3
Social Sciences	3
Earth and Planetary Sciences	2
Chemistry	2
Decision Sciences	1
Biochemistry, Genetics, and Molecular Biology	1

performance by making its results easier to compile and compare. Therefore, we suggest using *co-simulation-based optimization* for the technique described in this paper in future applications.

The main contribution of this survey, which presents a state-of-the-art review on co-simulation-based optimization, is to provide an analysis of how co-simulation and optimization are implemented. To do this, we examine the literature for emerging patterns in proposed strategies to identify a general abstraction of the methodologies. This abstraction showed that extensibility, standardization, and frameworks are the main approaches for handling the various disciplines involved in co-simulation-base optimization, as well as the various optimization techniques available.

In this document, we first present in Section 2 a general description of the main concepts of co-simulation and optimization, and the way they interact with each other. Throughout the confrontation of the concepts and applications found in the literature, we propose a new classification based on how a co-simulation is constructed. This classification provides an abstract representation of a co-simulation process and helps future researchers understand which problems can be solved with co-simulation.

Second, as discussed in Section 1.1, the involvement of several domains in a co-simulation-based optimization problem is a relevant and increasing situation in the field, which motivated us to perform a multidisciplinary analysis of the literature. Section 3 presents a study of the works done on co-simulation-based optimization, focusing on which domains are interacting with each other, how are the heterogeneous domains interacting with each other, and which simulation software is supporting this interdisciplinary initiative.

Third, we extend the additional complexity that brings the integration of optimization with co-simulation. In Section 4 we will study the literature from the optimization point of view, which means, analyzing which optimization methods are being used, which additional challenges are emerging from the optimization implementation, and how is the interoperability between co-simulation and optimization being handled.

Finally, Section 5 presents a discussion of the findings and contributions of the article, along with the concluding remarks. Furthermore, we present Table 5 in Appendix A, which contains the classification of the entirety of the literature sample database in terms of the construction classification approach (described in Sections 2.1.1 and 2.1.2), the simulation domains involved, the simulation software, the optimization method, and the optimization software. In Appendix C, we also provide a list of all the software mentioned in the articles together with a brief description of its purpose in Table 8. The list of acronyms used in the paper can be found in Appendix B.

## 2. CO-SIMULATION AND OPTIMIZATION

In this section, we provide a general description of co-simulation, optimization, and the black-box approach, as well as how we can conceptually combine these techniques. This establishes the basis of the core concepts of the document (co-simulation and optimization), as well as the synergy between them using the black-box approach, culminating in a general abstraction of co-simulation-based optimization. To accomplish these definitions, we use the seminal and implementation articles; in these papers, we find the concepts to determine the structure and evolution of the methodologies for co-simulation and optimization. Additionally, we will propose a construction classification for understanding existing co-simulation problems and identify the general process followed, offering an abstraction of the co-simulation structure to help understand the optimization implementation.

**2.1. Main concepts: Co-simulation.** The co-simulation methodology always starts with a target system that needs to be computationally represented. To do this, concepts such as modeling, simulation, and coupling are used to achieve a co-simulation that can effectively represent the target system. A model is defined as: “For an observer B, an object  $A^*$  is a model of an object A to the extent that B can use  $A^*$  to answer questions that interest him about A” [127], we can use a model of the target system to create a simulation (virtual implementation of the model  $A^*$ ) that behaves similarly to the entity A. The modeling and simulation process is illustrated in Figure 2a. However, in the case of some complex target systems, it is often found that a simulation is not enough to represent the interaction between heterogeneous domains in a system (multidisciplinary) [189]. In these cases, co-simulation is presented as a solution to handle the complexity and is defined as: “The theory and techniques to enable global simulation of a coupled system via the composition of simulators” [58]. This definition presents co-simulation as an extension of the simulation methodology that divides the target system into coupled sub-systems.

To illustrate the co-simulation concept, we propose the scheme seen in Figure 2b as a general abstraction of the co-simulation methodology. The co-simulation process starts with a complex target system that is conceptually divided into sub-systems, each sub-system is simulated using the same modeling and simulation

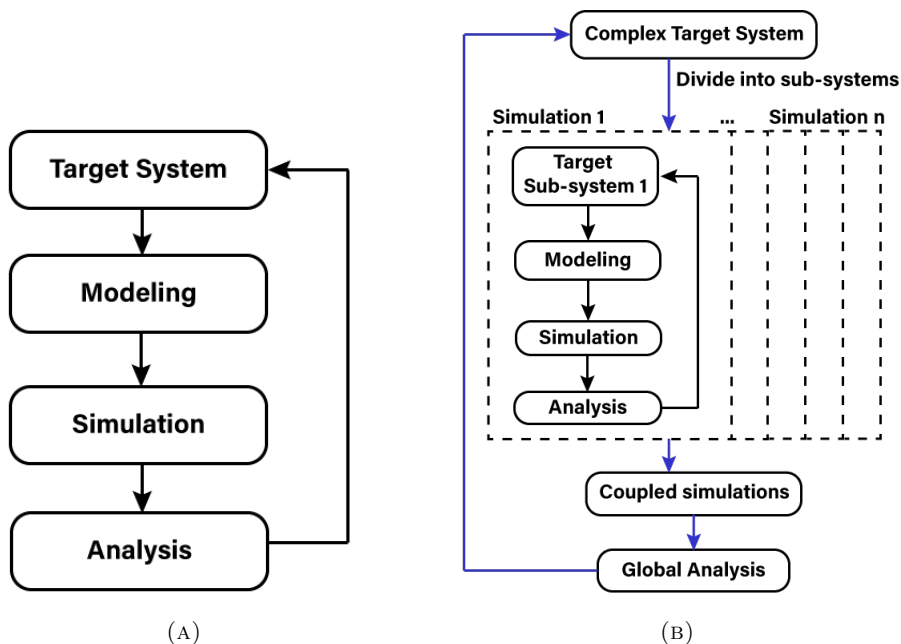


FIGURE 2. (a) Simulation process scheme (Adapted from [200]) including an analysis step after the simulation to compare with the target system, and a feedback loop connection to re-model and simulate until the validation of the simulation is achieved. (b) Proposed Co-simulation process scheme as an extension of (a) including the division of the complex system into sub-models that will be later coupled and co-simulated.

process described before (Figure 2a), then a coupling process is required, where the interactions between the different models are then performed using dedicated software in charge of exchanging and synchronizing data between models [58, 144], and finally, a global analysis to determine if the reliable representation is achieved, otherwise the process can restart for adjustments.

Co-simulation is becoming a growing trend for several reasons: first, as more and more projects involve developing teams from various domains of expertise, different formalisms inevitably have to collaborate; second, because simulation is a widely used technique, many model implementations still exist, making it cost- and time-effective to reuse existing models; and third, when multiple enterprises are involved in the same simulation project, there may be concerns about the intellectual property over the models, which means working with black-box approach (discussed in Section 2.3), which considers models that can interact with others while considering the internal processes as unknown. A method that addresses all the aforementioned challenges is co-simulation.

We have found in the literature sample reviewed for this article (described in Section 1.1) that there are two main approaches concerning how a co-simulation is constructed starting from the identification of the study problem (in this case we talk about the target system) to the final result (in this case the final product

is the functioning co-simulation). The classification depends on the context of the research; this means details like the different simulation software availability and the compatibility between the domains involved in terms of simulation coupling and type of information exchange. This discovery leads to a better understanding of the co-simulation pertinence and methodology.

The proposed classification helps understand the general design of the co-simulations that are already available in the literature and helps identify situations in which co-simulation may be the most appropriate methodology to apply to a new complex problem. We classified the 96 documents reviewed (classification available in Table 5) using the classical top-down/bottom-up approaches.

*2.1.1. Construction classification: Bottom-up approach.* In general, the bottom-up approach uses the inductive reasoning method, which starts with specific observations that can build a more complex or general theory. This means we can start with the definition of specific systems to be simulated, then a big-picture complex system can be built through the coupling of the different simulations. This scenario can arise when multiple pre-existing simulations are available but do not provide reliable enough results. In these scenarios, the simulations can boost each other's performance with coordinated interactions. Figure 3 provides a general illustration of the co-simulation bottom-up approach. It shows how previously available independent simulations that represent different components of a complex system are coupled to create a co-simulation. This process involves assessing whether the simulation input and output are compatible and how they interact (coupling). Using the individual capabilities of each simulation and combining them to enhance the global behavior with the coupling mechanisms allows the user to present a virtual representation of the complex system.

The methodology used in the research documents that we classify as the Bottom-up approach describes how independent simulations are designed in two or more different simulation software, to later be coupled and used as a collective coordinated complex simulation (co-simulation). The articles that present the bottom-up approach are identified in the literature sample (see Table 5).

*2.1.2. Construction classification: Top-down approach.* In general, the top-down approach uses the deductive reasoning method, which starts from a general theory or methodology to formulate a hypothesis over a system, then collects the information, and finally validates or rejects the hypothesis. For the co-simulation context, this means that there is a co-simulation methodology used to formulate and test a hypothesis over a complex system that fits the methodology domain. This situation occurs when the target system cannot be handled with only one simulation, but a co-simulation software or framework allows it to divide into smaller, interconnected sub-systems. Figure 4 illustrates the general structure of this approach, which breaks down a complex target system into smaller sub-systems that are subsequently simulated and coupled using dedicated software. The resulting co-simulation is then evaluated and contrasted with the target system to ensure validity.

In the case of the research works reviewed with the top-down approach, documents such as [2, 180, 196, 16, 189] present general frameworks for co-simulation of a complex system that allows it to model a system in a modular manner, where most often each module belongs to a different domain (full classification available



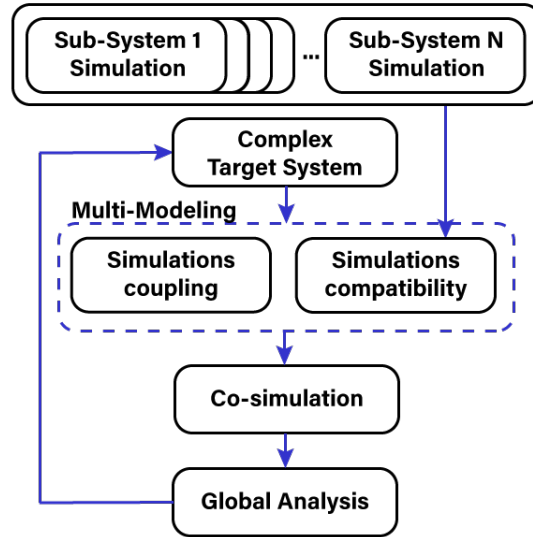


FIGURE 3. Proposed classification: Bottom-up approach for co-simulation. A target system is abstracted as a co-simulation, using pre-defined simulations coupled with each other and compared with the target system in a global analysis stage.

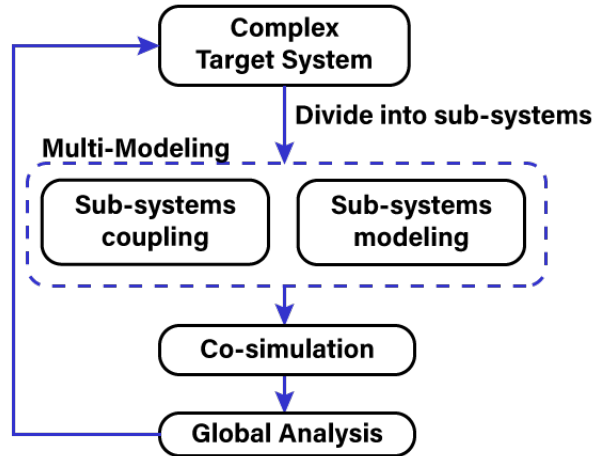


FIGURE 4. Proposed classification: Top-down approach for co-simulation. A target system is multi-modeled, i.e., it is divided into subsystems that are coupled and simulated. To provide feedback, this co-simulation is then compared to the target system using global analysis.

in Table 5). For example, the co-simulation of a complex mechanical part from a helicopter is co-simulated using the ANSYS environment in [180], where 3 different

simulations are designed and coupled, each simulation describes the system under different domains (fluid, physical, and thermal mechanics).

The use of either approach to achieve a co-simulation is just the beginning of the study of a complex system. The co-simulation allows us to test configurations by tuning parameters or coupling patterns; these possibilities raise the interest in finding a combination of parameters that optimizes a key output performance. For this, the combined simulation-based optimization methodology has been an important technology tool since 1998 [114], and for this reason, it is a natural continuation for co-simulation to collaborate with the optimization field.

**2.2. Main concepts: Optimization.** Optimization consists of finding a value that minimizes or maximizes some specific function among all possible values (in a defined domain) that satisfy some conditions or constraints given [27], the general optimization process is shown in Figure 5. To begin, a target system is modeled, which means defining constraints, objective function, and decision variables [163]. The model's output will then be analyzed using an optimization method to determine the ideal model configuration. Lastly, an analysis is done on the data to see if the outcomes match the model's expectations, and if not, the process is restarted.

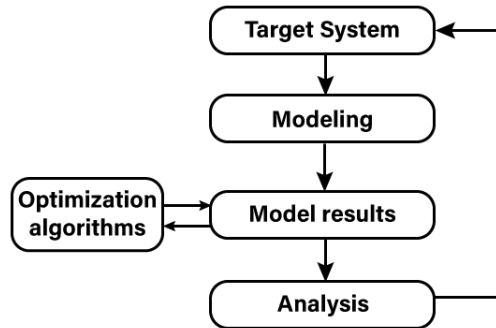


FIGURE 5. Optimization process scheme. Adapted from [163]

The schemes presented in Figures 5, 2a, and 2b present a general methodology similarity since they study a target system by proposing a model to understand it or predict it, and then analyze and compare the results of the model with the target system to provide feedback and adjustments. This shows the fundamental compatibility between simulation, co-simulation, and optimization; including the fact that large-scale, stochastic, and high-complexity problems constitute typical areas in which simulation-based optimization is applied [59], we can affirm that optimization can leverage co-simulations to explore the process design space [54]. However, in order to present a general abstraction of the combination of co-simulation and optimization, it is necessary to have a representation capable of encapsulating these concepts in a way that they can interact and boost each other.

**2.3. Co-simulation-based optimization: black box approach.** The concept of the black box was already discussed in computer science around 1988; it was defined as hardware and software components that show an interface so that programming can be conducted based on knowledge of the operation of the interface but not the functioning of the components as such [190]. The idea is to have a

module or “box” with internal unknown mechanisms and states, and the only way to interact with this box is by providing information (input), asking the module to process that information, and getting the results (output). The black-box approach is effortlessly applied for co-simulation representations, as the modeling and simulation processes are already encapsulating a system representation into a box with inputs, outputs, and internal processes. Therefore, we can use the black-box approach to abstract the general structure of a co-simulation.

We described two distinct approaches for building a co-simulation; both methods address the co-simulation design and validation procedure. However, the outcome of these techniques is the same for both methods: a multi-model integrated into a co-simulation. This multi-model can be illustrated with boxes and arrows representing the models and couplings as an abstraction of the structure. We present a general co-simulation black-box representation in Figure 6a that shows how several coupled sub-models create a bigger box known as a multi-model, which has inputs and outputs. In Figure 6b, we present a black-box representation of the co-simulation study case developed in [189], which depicts the impact of weather conditions on the house internal temperature and the operation of an air conditioning system that adjusts the temperature based on the occupancy of the house and the current temperature.

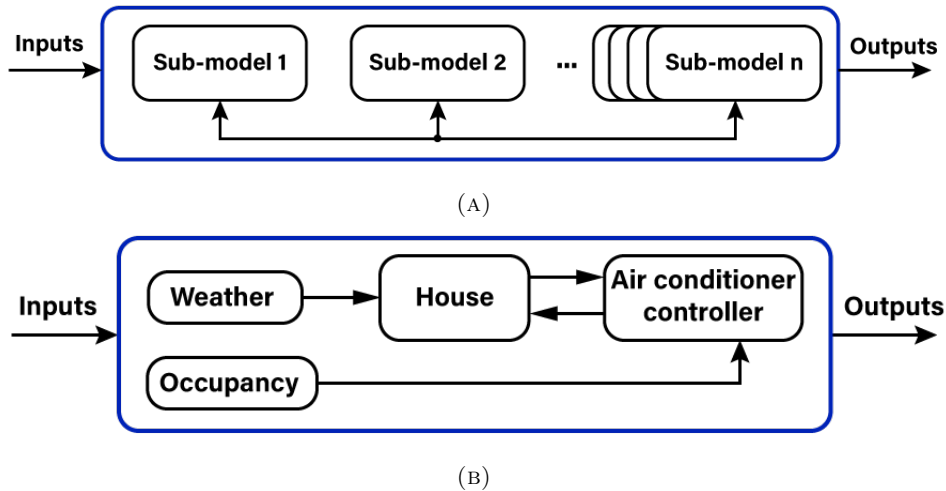


FIGURE 6. (a) Co-simulation black-box general representation. (b) House internal temperature’s co-simulation black-box representation (adapted from [189]).

We may begin to consider a generic representation of the cooperation between co-simulation and optimization thanks to the general black-box representation proposed in this Section. Drawing from Figure 5, the multi-model generalization suggested in Figure 6a, and the architectures employed in the reviewed documents (e.g., [2, 180, 196, 16, 189]), it is possible to identify the need for an interface between the co-simulation and the optimization algorithm as shown in Figure 7. This

interface is responsible for translating the inputs and outputs of the co-simulation software towards the algorithm, depending upon the algorithm’s requirements.

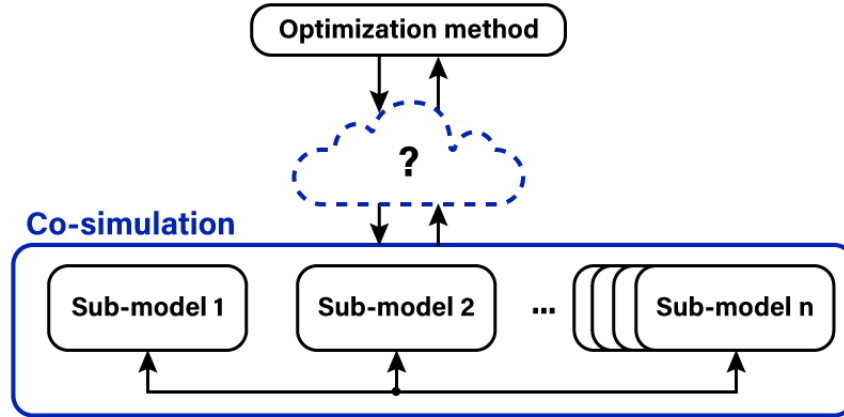


FIGURE 7. Co-simulation and optimization black-box representation.

**2.4. Challenges and conclusions.** Co-simulation is a helpful technique for comprehending and forecasting the behavior of intricate mechanisms. Optimization can leverage this representation to provide a detailed analysis and suggest adjustments or solutions. These general concepts have a clear potential; yet, because the idea of co-simulation-based optimization is still in exploratory stages, new applications may need further research to reconcile the methodology of existing applications with novel ones. A generalization of the co-simulation and optimization interaction will define the specific nature and requirements of the interaction between the components. This interaction is not clear enough for a working generalization (Figure 7), as there is a wide variety of strategies to manipulate and communicate the co-simulation with the optimization.

Co-simulation is a methodology beneficial for multidisciplinary work, as it aims to connect heterogeneous systems without taking them out of their domain. For this reason, many of the applications found in the literature combine domain-specific simulation software, causing a convergence of the applications into this methodology. Nevertheless, the many individual efforts to perform co-simulation-based optimization are not working under a general method, causing redundant frameworks to be proposed in different domains.

### 3. MULTIDISCIPLINARY ANALYSIS

In this section, we present a review of the recurrent domains and software used for co-simulation-based optimization applications. We also highlight the recurrent interdisciplinary collaborations and the advantages of using co-simulation for optimizing complex multi-domain problems. An overview of the literature focused on the multidisciplinary aspects of a problem, these applications usually propose a holistic representation of a system that can provide a better performance than other models. The emerging patterns found through gathering and studying the

literature can lead to a conceptual architecture of the process, which is a comprehensive contribution to the field that will benefit future applications in the same area. The goal is to understand how to generalize the co-simulation-based optimization methodology, why doing so is beneficial, and how the availability of domain-specific software impacts the approaches employed in the implementations.

The heterogeneous approach of co-simulation encourages multidisciplinary works, for example, the design of a vehicle involves the interaction of its major sub-systems (engine, control, hydraulics suspension, and pressure), which involves different domain-specific dynamics like mechanics, logic, fluid, and thermal. In [154] a virtual prototype project is proposed between companies to combine the sub-systems to evaluate the performance of the vehicle using a shared co-simulation environment to find the best configuration of the sub-systems that will produce the desired vehicle performance (global optimization). To understand the multidisciplinary influence on the topic we will use the following methodology: first, we will analyze the different domains interacting in the co-simulation-based optimization works in the literature, this domain-approach review will highlight the synergies already present between domains and encourage new collaborations. Second, we will perform a co-simulation software review to extend the analysis from this section to the software availability factor, which can influence the research path followed by each domain expert. Finally, we will discuss the challenges in the multidisciplinary field of co-simulation-based optimization.

**3.1. Simulation domains.** In contrast to the SCOPUS classification of domains, where the primary goal is to categorize the research fields (see Table 1), there are many simulation domains with different levels of detail ranging from specialized domain simulations (i.e. fluid dynamics simulation with FLUENT [203] or with COMSOL [194], numerical simulation with MATLAB [156]) to big systems domain simulations (i.e. building performance simulation with EnergyPlus [203, 191, 126, 115, 36, 25] or with TRNSYS [41, 42, 46, 47], vehicle simulation with CarSim [150]). As part of this survey, we provide a full classification in Table 5 of the simulation domains in the literature (third column) that compile a total of 49 different domains. This classification was done manually, taking into account the general purpose of the software used, as well as the document definition of the model created with the software. From these domains, we can identify a particularly interesting one, the Computational Fluid Dynamics (CFD) [75], which is used on 21% of the articles reviewed in connection with other domains. In Table 2 we can see all articles from the sample database that use CFD simulations, the table shows the great variety of combinations present in terms of domains and software. Additionally, we include the interoperability methodology used to integrate the CFD simulation with the other domain software.

Table 2 shows that CFD simulations are widely combined with other domains such as solid (Computed-Aided Design for creating 2D or 3D virtual representations of real-world products), energy, numerical, building, and more. The diversity of combinations and integration methodologies shows the complexity of the field of co-simulation. We can see the presence of software to handle the integration of the unitary simulations from different domains, manual implementations, and the use of import/export tools to combine the simulations. The different methods highlight the methodological significance of the interoperability of the simulations, which is correlated with the software compatibility and the domains in question.

TABLE 2. Co-simulation collaborations with CFD by domain, simulation software, and integration method.

Article	Simulation domain	Software	Integration method
[180]	Fluid Solid	FLUENT Solidworks	Import CAD file into ANSYS
[196]	Fluid Electric	Spectre LT-Spice	Manual (coded in Verilog)
[203]	Fluid Building	FLUENT EnergyPlus	Manual
[62]	Fluid Control	FloMaster Simulink	FMU imported into Simulink
[73]	Fluid Solid	FLUENT Solidworks	Imported into ANSYS plugin
[17]	Fluid and electric	COMSOL	COMSOL framework
[70]	Fluid Mathematical	FLUENT Simulink	Imported using MATLAB plugin
[194]	Fluid Electrical Solid	COMSOL COMSOL COMSOL	COMSOL Framework
[179]	Fluid and solid	COMSOL	COMSOL framework
[145]	Fluid Mechanical	Gleeble Gleeble	Gleeble framework
[116]	Fluid Numerical	FLUENT Particle Flow Code	Manual
[74]	Solid Fluid Dynamic	ANSYS FLUENT EDEM	UDF library imported into ANSYS
[192]	Fluid Solid	FLUENT EDEM	UDF library imported into ANSYS
[211]	Solid Fluid	Pro ENGINEER FLUENT	Imported using ANSYS plugin
[64]	Fluid Solid	FLUENT EDEM	UDF library imported into ANSYS
[195]	Fluid Solid	FLUENT MATLAB	Imported using MpCCI toolset
[41]	Fluid Energy	FLUENT TRNSYS	Manual
[71]	Fluid Engine	FLUENT GT-Power	UDF library imported into ANSYS
[132]	Fluid Numerical	FrontFlow/Blue VSI Fortran	Manual
[44]	Fluid Electrochemical	AMPERES IPS	AMPERES-IPS Framework
[215]	Fluid Chemical	FLUENT Aspen Plus	APECS Framework

**3.2. Simulations interoperability.** Interoperability, as described in [184], exists in 6 levels that go from low integrability to full conceptual interoperability (no compatibility, technical, syntactic, semantic, pragmatic, dynamic, and conceptual).

This concept has increased its relevance in the simulation fields [106], discussing the importance of offering interoperability across software from different domains. This means that a model created in a particular software should be able to interact with other models created with other software without losing capabilities. To evaluate the current literature environment regarding software interoperability we focus on co-simulation interoperability (sub-systems interaction), in which we identified 3 approaches within the literature reviewed, i.e. manual implementations, collaborating standards, and co-simulation framework. In Section 4, we will expand the analysis to the interoperability between optimization and co-simulation.

*3.2.1. Manual implementations.* This methodology covers applications in which the user manages the information transfer between the various simulation software programs, handling all aspects of the interoperability of the simulations [203, 201, 116, 159, 41, 132]. This implementation strategy solves the software interoperability issue for the specific study instance but does not offer a generic solution that can be applied to other comparable projects in the future, making it an ad hoc solution. Specific examples of manual implementations coupling using high-level programming languages to set up the co-simulation such as ActiveX [63], Verilog [196], C++ [204], and communication protocols like Memory Mapped Files [125], or TCP/IP and DNP3 [158, 2, 121, 122]. The presence of manual implementations in the literature highlights the benefits of a fully manual interface between simulations, including control over the coupling mechanism in terms of input/output formats and event coordination. However, it also highlights the drawbacks of an ad hoc solution to the interoperability problem, which includes providing a scheduling algorithm and solving the information coherence, as well as the method’s lack of reproducibility, which will be of greater concern for the next two categories.

*3.2.2. Collaborating Standards.* This strategy is used by software companies to offer compatibility with other software that could improve the performance of a simulation. We identified two categories of collaboration standards: those that allow simulations to be imported in a standard format and those that offer software extensions such as toolboxes, plugins, and APIs. The first type of collaboration involves simulation software that offers services such as plugins/toolbox [70, 193, 66, 156, 47, 128, 131, 211, 42, 195, 46, 214, 37, 157, 205, 73, 15] or APIs [171] to enable compatibility with other software. These tools are an extension of the main software capabilities to provide a link with external software that handles the interoperability of the simulations. The second type of collaboration consists of simulation software that provides the capability of importing models created in external software using a standard format. In the literature, we find examples of domain-specific standards formats such as CAD files for 3D and 2D physical models [180, 73, 5], Materials Property Data exchange (.matml) [172], circuit simulation import using a Touchstone file standard [129] or the s-parameter format [111, 110]; we also find examples of non domain-specific formats like UDF libraries (.dll) [74, 192, 64, 210, 67, 26, 71], XML format [202], ASCII Files [28], and the Functional Mock-up Interface (FMI) standard [146] that generates “.fmu” files from any domain [62, 126, 36, 25, 106, 150].

The existence of multiple exportation standard applications and software compatibility tools is proof of the usefulness of co-simulation software, which is different from traditional simulation software in that it can couple two or more simulations to

produce a global behavior. Examples that we found include the Model.CONNECT [68, 19] software, which provides an easy-to-use interface for simulation model coupling and focuses on the configuration of the interaction of the models, while the definition of the individual models is imported from external software; other examples include OpenModelica [142], Simulink [103] and Adams [13] software, which combines native simulation capabilities with importation methods to combine simulations in a shared environment; this shared environment concept is especially exploited in the next category.

**3.2.3. Frameworks.** This methodology is a general approach to co-simulation, where we find architectures that encapsulate several domain-specific simulation software to coexist in one big environment. This approach is not opposed to the previous exporting standards approach; on the contrary, it can use standards to expand its reach. The objective of these frameworks is to offer a platform to develop multidisciplinary simulations that can be coupled easily. This approach establishes a standard for the coupling mechanisms, which means that the interoperability is handled internally by the framework, and the user only focuses on the simulation's setup. The advantage of co-simulation frameworks (also referred to as platform, environment, or architecture) is that they encourage reusability as they offer a set of tools to continue developing co-simulations for future researchers which at the same time will contribute to the validation of the framework. The examples found in the literature are presented in Table 3, where we classify each framework based on the domain covered, the articles that use the framework, and the source proposing the framework, which comes from published articles describing the characteristics of the framework or a software product offered in the market.

Table 3 shows that there are already 25 co-simulation frameworks available in the literature, each with a different target domain. The benefit of a framework over an ad hoc application is that its results serve as an incentive for the domain to continue working on the issue with the same framework in the future, thereby facilitating and encouraging progress. The table presents two special cases of frameworks that have already been used in multiple applications in the literature sample database: ANSYS [82], a multi-domain co-simulation tool, with six applications, and COSMO [207] for the mobility field, with four. This reproducibility demonstrates the value of a framework as each iteration takes advantage of the co-simulation features and validates the suggested framework. The framework category is the most robust of the three categories this section defines since it provides a common environment for user-friendly coupling mechanisms, opens the door for further applications following the same structure, and works towards an abstract method to perform co-simulation.

**3.3. Challenges and conclusions.** This section presented an analysis of the literature sample on integrating multiple domains into a co-simulation. This analysis shows the increasing presence of collaboration and frameworks in co-simulation software (Tables 2 and 3), these strategies use general patterns to couple two or more simulations that can belong to different domains. The co-simulation state-of-the-art [58] described extensibility as a remaining challenge for co-simulation, that is, the ability to include new features; we find frameworks such as MECYSCO [189], APECS [215], Helics[155] and GUSMA [154] as examples of the extensibility



TABLE 3. Co-simulation frameworks in the literature.

Framework name	Domain covered	Used in	Proposed by
EXPPO	Cyber-Physical Systems	[22]	Article [22]
HELICS	Multi-domain	[155]	Article [65]
MATLAB	Multi-domain	[68]	Software product [97]
EV-EcoSim	Electric vehicle	[21]	Article [21]
ANSYS	Multi-domain	[124, 170, 119, 72, 40, 198]	Software product [82]
MECSYCO	Multi-domain	[189]	Software product [39]
MODUS	Hardware/Software	[117, 23]	Article [23]
COMSOL	Multiphysics	[17, 194, 179]	Software product [32]
Cadence	Thermo-electric	[69]	Software product [91]
Amperes-IPS	3D Magnetic [164]	[44]	Article [44]
GUSMA	Vehicle design	[154]	Article [154]
COSMO	Mobility	[30, 209, 206, 208]	Article [207]
EPPY	Building design	[115]	Article [115]
LabView	Aerial vehicle design	[107]	Software product [34]
Rhino	Building design	[191]	Software product [101]
ADS	Circuit design	[199, 112]	Software product [181]
Gleeble	Thermomechanical	[145]	Software product [94]
COSCO	Fog computing	[187]	Article [187]
APECS	Fossil energy	[215]	Software product [130]
Not-named	Bio-fabrication	[54]	Software product [38]
Not named	Photonic and electronic	[12]	Software product [9]
Not named	Multi-domain	[197]	Article [197]
Not named	ROS-based [153]	[16]	Article [188]
Not named	OPAL-RT-based [99]	[24]	Article [24]
Not named	Robotics	[6]	Article [6]

advantages. These frameworks present co-simulation software that handles simulation tools, capabilities of importing external models, and also the possibility of developing new compatibility modules to other software using high-level code languages.

Another challenge discussed in [58] is the combined approach of the discrete event and continuous time for co-simulation software, the idea being to leverage the combined features of the two formalisms. We can see that the number of different methods used to perform the co-simulation in the literature is a sign of the complexity that presents the combination of heterogeneous simulations in terms of formalisms, domains, environment, and communication. The level of integration achieved between the simulations can be assessed with the interoperability levels mentioned in Section 3.2. These levels evaluate the integration of components according to integrability, interoperability, and composability [184]. The black-box approach claims its relevancy in this topic, by encompassing the extensibility and framework approaches, these approaches aim to describe a system in terms of abstract components that interact and that can support further composition with new components in a nested manner, which is compatible with the black-box approach.

At the same time that the number of co-simulation frameworks increases and abstract structures are proposed, the opportunity to include optimization as an

extension of the capabilities of co-simulation grows. This survey takes this opportunity to leverage existing applications of co-simulation-based optimization and confront them with the cumulative complexity carried by the complex systems studied, the co-simulation particularities, and the optimization field requirements (discussed in the next section).

There is remarkable progress in the field, which we can prove by comparing three of the conclusions of the state-of-the-art article on co-simulation from 2018 [58]. First, by 2018 the large majority of applications used ad-hoc couplings between simulators, which now (at least for the co-simulation-based optimization share) is mostly dominated by import standards and co-simulation frameworks. Second, the challenges for co-simulations were many, but two specific mentions are already discussed in this document, modular coupling (in Section 2) and finding standards for hybrid co-simulation (in this Section). Third, Gomes, et al. assert that the main conclusion of the review is that there is a lack of research into the modular, stable, and accurate coupling of simulators, which is now an important concern of software developer companies, as shown by the amount of co-simulation software available. The next step for the co-simulation technique that provides complex system abstractions is to offer complex insights into the system. This is the objective of implementing optimization as a general combined technique; this combination offers to explore, study, and improve complex systems.

#### 4. OPTIMIZATION ANALYSIS

In this section, we present a review of the optimization methods used in co-simulation-based optimization (Section 4.1) and analyze the interaction strategies between co-simulation and optimization (Section 4.2) to understand the nature of the interaction illustrated in Figure 7. An analysis of the methodologies and interactions in this field will help identify promising approaches for working with co-simulation-based optimization, which will benefit both software companies and users.

As described in Section 2.3, the interface between the co-simulation and the optimization method can be understood using the black-box approach as shown in Figure 7. The goal of this analysis is to understand two main aspects of the figure, namely the internal structure of the optimization method box and the interface characteristics. To achieve this, we reviewed the database sample regarding the optimization methods used, and the interaction strategies used to combine optimization and co-simulation (Table 5) to be later confronted with the black-box approach.

**4.1. Optimization methods in the literature.** The optimization methodology used in simulations is a recurrent combination, as it is a natural use of system virtualization (simulation) to test and find improvements in the system by tuning the structure or parameters which can lead to an optimal or pseudo-optimal performance configuration. This makes the process less expensive than testing on the real system. To understand how the optimization methodologies are being implemented in the particular case of interest of this article (i.e., co-simulation), we will analyze the optimization types present in the literature sample, which follows the tree structure abstracted in Figure 8.

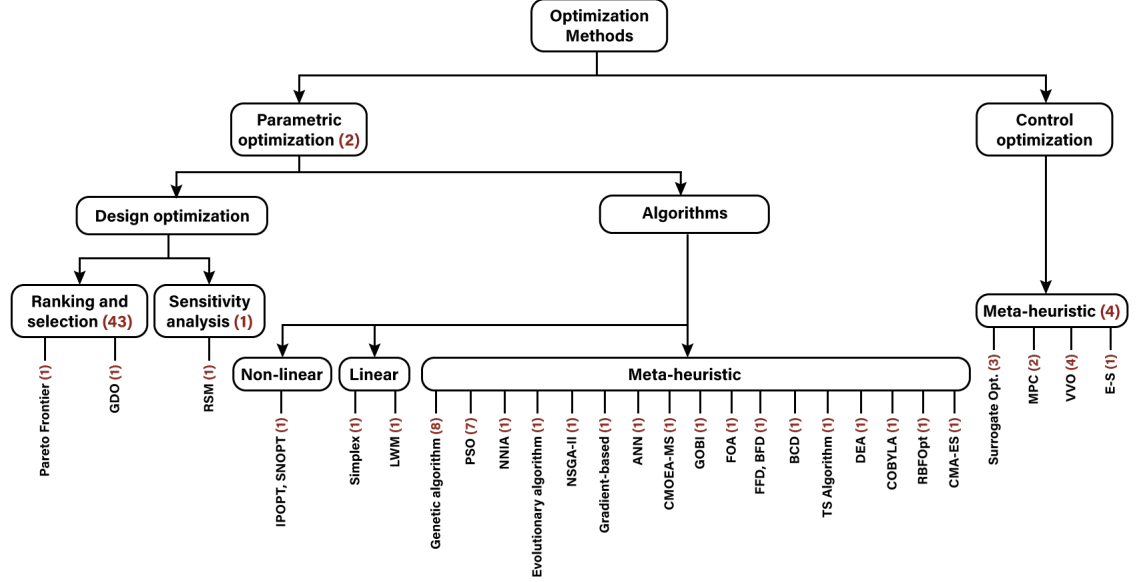


FIGURE 8. Optimization categories and specific methodologies used in the sample database (the amount of articles using each method is included in red in parenthesis, see table 5).

$$\begin{aligned}
 & \min \text{ or } \max \quad f(\vec{x}) \\
 (2) \quad & \text{subject to:} \\
 & g_i(\vec{x}) \leq C_i, \quad i = 1, \dots, n
 \end{aligned}$$

There are two types of optimization problems: parametric and control [59]. Parametric optimization refers to problems that can be represented with the general Equation 2, where we are minimizing or maximizing an objective function  $f(\vec{x})$ , the vector  $\vec{x}$  represents all the parameters or decision variables of the problem, the problem also has  $n$  constraints represented by  $g_i(\vec{x})$  compared to  $n$  constants  $C_i$  [59]. In general, parametric optimization problems are known as static systems [44, 215], whereas control optimization problems handle dynamic systems as they change in some way from time to time, the objective is to choose a set of actions to move the system along a desirable path. The general Equation 3 is defined in [59] where  $\mathcal{S}$  denotes the states of the system and  $|\mathcal{S}|$  the number of states in the system,  $\mu(i)$  refers to the action taken in state  $i$ , and finally the function  $f$  denotes the objective function to be optimized.

$$(3) \quad \min \text{ or } \max \quad f(\mu(1), \mu(2), \dots, \mu(|\mathcal{S}|))$$

Each of these categories mentioned has many techniques proposed which can be again classified into sub-categories, the classification depends on the problem's nature as well as the approach taken to solve it [59]. In Figure 8 we see the category division of optimization methods, starting from the basic categories (parametric

and control) taken from [59], and then the subdivision depending on the characteristics of the methods found in the literature<sup>1</sup>. Hereafter, we briefly describe each category and method of optimization in Figure 8 including the articles from the sample database that use each method (see Table 7 for abbreviations).

Design optimization encompasses the approaches that aim to reduce the solution search space to a relatively small size (most of the time under 20 scenarios [59]) using intuitive or mathematical tools, to be later confronted against each other with an objective function. An example of this approach is *Ranking and selection*, which is the methodology most used in the literature (with 43 of the 96 articles in this category, see Table 5) with two additional cases of particular ranking and selection post-processing techniques, namely, Pareto Frontier [195] and GDO [211]. Sensitivity analysis is another approach that reduces the search space by performing a mathematical treatment of the available data like in [40] and a more specific method of sensitivity analysis like RSM in [116].

The algorithmic approach to optimization follows a specific set of steps to find an optimal or near-optimal result while avoiding exploring all the search space. The linear (e.g. IPOPT, SNOPT [17]) and non-linear (e.g. simplex [24] and LWM [67]) algorithms depend on the assumption that we know the tendency of the solution.

In contrast, for other algorithms that present unknown tendencies, the meta-heuristic approach can offer interesting algorithms that employ distinct features to advance from one system's performance to a better one until a stopping criterion is satisfied. These methods are shown in Figure 8 and we link, in the next paragraph, each algorithm with the articles from the database.

We find in the literature sample database meta-heuristic methods using adaptive algorithms such as Particle Swarm Optimization (PSO) [124, 66, 47, 131, 107, 46, 37], also evolutionary algorithms [16] such as Fruit-Fly optimization algorithm (FOA) [170] (both part of the swarm intelligence approach), constrained multi-objective evolutionary algorithm with multiple stages (CMOEA-MS) [179], Covariance matrix adaptation evolution strategy (CMA-ES) [5], Differential Evolution Algorithm (DEA) [12], genetic algorithms [54, 21, 171, 128, 25, 42, 106, 63] including the variation NSGA-II [70]. We also have gradient-based algorithms [189] like GOBI [187] and Block Coordinate Descent (BCD) [155], methods that approximate behavior to mathematical functions like Constrained Optimization By Linear Approximation (COBYLA) [197] and Radial Basis Function Optimization (RBFOpt) [191], methods that use Neighborhood search strategies like Non-Dominated Neighborhood Immune Algorithm (NNIA) [180] and Tabu Search (TS) [204], and finally other approaches like Artificial Neural Networks (ANN)[199], and FFD/BFD [22].

Compared to parametric optimization, the number of control optimization problems using co-simulation in the literature is quite small (14 of the 96 documents). In control optimization, due to the dynamic nature of the system, the solutions are mostly focused on dynamic or mathematical programming [59], this explains the dominance of meta-heuristics found in the literature. We found studies that implement meta-heuristics [206, 209, 208, 158] to optimize dynamic systems with

---

<sup>1</sup>the Figure 8 classification is not exhaustive regarding existing optimization methodologies, but it is exhaustive regarding the 96 articles reviewed in the sample literature database.

techniques such as surrogate optimization [111, 112, 110], Model Predictive Control (MPC) [115, 126], Volt-var optimization (VVO) [121, 122, 26, 2], and EXTREMUM-SEEKING (E-S) [150].

The literature contains a variety of algorithms that demonstrate the wide range of optimization techniques that may be applied in a co-simulation scenario. These techniques involve modifying parameters or control strategies to explore the search space for optimal performance. Regardless of the approach, the interaction between co-simulation and optimization is crucial for comprehending the conceptual relationship between these elements. To investigate this further, we conduct an interoperability analysis similar to the one conducted for co-simulation interoperability, in order to identify the features of this collaboration.

**4.2. Optimization Software and frameworks.** As discussed in Section 3.3, an analysis of the software used to develop the implementations in the literature gives the opportunity to study the characteristics of the conceptual interface between concepts. In the previous section (Section 3.2), we discussed the three approaches used to handle the interoperability between the simulations; now we will do the same analysis focused on the interoperability between co-simulation and optimization. To better understand the requirements and challenges of this collaboration, we reviewed the sample database of articles and extracted three categories that represent the interoperability management strategy between co-simulation and optimization.

*4.2.1. Manual optimization.* The first category covers the cases where no optimization software was involved in the process, this means that the optimization process was done by the researcher. There are 38 articles in the database of this review that used the Ranking and Selection optimization methodology without using any optimization software (see table 5), which, as described in this Section, consists of reducing the search space as much as possible to have a reasonable amount of scenarios to co-simulate, compare, and select the optimal. Apart from these cases, there is only one more case that fits this category [67] where co-simulation was used to define the constraints of the optimization problem, and the optimization process (Linear Weighted Method) was performed manually by the researcher.

*4.2.2. Software collaboration.* The second category refers to the cases where two or more different software environments were combined to achieve the co-simulation-based optimization process. All cases are classified in Table 4 where we can see the co-simulation software, the optimization software, and the method used to integrate these components. In general, we found three types of integration, manual, using extension tools, and optimization frameworks. The first type is the manual interface, which refers to the cases where the researcher takes the outputs of the co-simulation software and provides the data as input to the optimization software, either by manually assigning the input/output files [111, 110, 201, 122, 121, 131] or by creating code that automatizes this part of the process with high-level languages such as C++ [204, 26], ActiveX [63], or Python [197, 115]. The second type is the use of existing tools to handle interoperability such as libraries [42, 46, 54, 199], plugins [70, 124, 2, 24, 37, 66, 15, 214], export standards [150, 128, 5], and APIs [179, 171, 12, 126, 25, 155]; these types of tools are always used in the presence of high-level and high-demand software languages such as Python and MATLAB

(see Table 4). Finally, the third type refers to optimization frameworks, which are optimization-dedicated software that are specially designed to serve as a modular tool that can collaborate with third-party software to achieve optimization using different optimization algorithms with a black-box approach (as described in Section 2.3), the optimization framework software found in the literature are GenOpt [47], OptiY [158], and modeFRONTIER [195, 106].

TABLE 4. Methods of interoperability between co-simulation and optimization software found in the literature (Software references in Table 8).

Co-simulation software	Optimization software	Interface method	Ref
AnnAGNPS	C++	Manual	[204]
EMTP-RV	C++	Manual	[26]
ADS	FEKO	Manual	[111, 110]
Matlab	GenOpt	Optimization framework	[47]
SWAT	Lingo	Manual	[201]
ANSYS	MATLAB	Plugin	[70, 124]
CarSim	MATLAB	Exporting standards	[150]
COMSOL	MATLAB	API	[179, 171]
EnergyPlus	MATLAB	Manual	[115]
OPAL-RT	MATLAB	Plugin	[2, 24]
OpenDSS	MATLAB	Exporting standards	[128]
PSIM	MATLAB	Plugin	[37]
RSCAD	MATLAB	Manual	[121, 122]
Solcore (python)	MATLAB	API	[12]
Solidworks, ADAMS	MATLAB	Exporting standards	[5]
SPEED	MATLAB	Manual	[63]
SWAT	MATLAB	Manual	[131]
TRNSYS	MATLAB	Library	[42, 46]
Xilinx	MATLAB	Plugin	[66]
Python	MicroGP library	Library	[54]
ANSYS	modeFRONTIER	Optimization Framework	[195]
OpenModelica	modeFRONTIER	Optimization framework	[106]
SimulationX, LS-DYNA	OptiY	Optimization framework	[158]
ADS	Python	Library	[199]
Aspen plus	Python	Manual	[197]
EnergyPlus	Python	API	[126, 25]
Gridlab	Python	API	[155]
GT-Power	Simulink	Plugin	[15]
Solidworks, ADAMS	Simulink	Plugin	[214]

4.2.3. *Co-simulation-based optimization frameworks.* The third and last category is the most robust of the approaches, as it aims to incorporate all aspects of the process into a single software framework. A software environment that fully supports the co-simulation-based optimization process is a great tool for researchers to study

and understand complex systems insightfully, with a reliable complex virtual representation and several optimization methods available to be implemented into the model effortlessly, which is a big opportunity for upcoming studies. The big-picture approach of the framework methodology provides an environment that encourages multidisciplinary collaboration and reusability. In the literature, we found a total of 11 co-simulation-based optimization frameworks that are specially designed for a domain-specific research problem, there are the electronic framework offered by ADS [181] used in [112], the process engineering dedicated framework APECS [130] used in [215], the multiphysics framework COMSOL [32] used in [17], the fog computing framework COSCO proposed in [187], the mobility framework COSMO proposed in [207] and used in [206, 209, 208], the physics framework DAKOTA-IPS [183, 147] proposed in [44], the electric vehicle power framework EV-ECOSim proposed in [21], the mathematical-based framework MATLAB [97] used in [16], the hardware and software (H/S) framework MODUS used in [23, 117], the Cyber-physical Systems framework EXPPO proposed in [22], and the Computer-aided Design framework Rhino 3D [101] used in [191].

The frameworks presented propose a generalization of the co-simulation-based optimization approach to propose an environment for a specific domain, even more general frameworks are found in the literature that aim to use the same approach to cover multidisciplinary domains in general, the frameworks found present software environment with the purpose of total simulation integration for design, manufacturing, and engineering problems. It offers a wide variety of domain-specific modules to create simulations that share the same environment for maximum interoperability, such as ANSYS [82] (used in [180, 116, 211, 40]), Isight [176] (used in [170]), LabView [34] (used in [107]), and MECSYCO [39] (used in [189]).

**4.3. Challenges and conclusion.** The many optimization categories and techniques found in the literature showed the wide variety of optimization possibilities available. This shows the complexity of deciding which optimization methodology to utilize. In the literature, almost half of the cases (47 out of 96) choose to diminish the scenario amount to perform the analysis over a controlled number of co-simulations to find the optimal (design optimization) and to use meta-heuristics almost the other half of the cases (44 out of 96) to find a near-optimal solution that is computationally viable.

As each optimization method employs different approaches, each with highly varied setups and interaction features between the optimization strategy and the co-simulation, there is a challenge to propose a generalized structure for co-simulation-based optimization that ensures interoperability between the co-simulation and optimization software.

We can find in the literature some proposals that contribute to the field with ideas like cross-company collaboration [97, 32], domain-specific frameworks combining specialized software with optimization tools [187, 130, 207, 181, 21, 44, 117, 101, 22], multi-disciplinary sub-systems simulations in a common environment [82, 176, 34], and conceptual frameworks using a modular architecture with black-box optimization [39, 189, 113, 47, 167, 195, 106, 55, 158].

We highlight the importance of extensibility and the black-box approach as an adaptive strategy in co-simulation software, as it offers flexibility in the integration of external simulations while also allowing optimization methods to be integrated into the process. Extensibility was already mentioned in [58] as a key challenge

to be explored in co-simulation. This article’s work shows that the combination of the many applications in the literature with the black-box approach and the extensibility has benefited not only co-simulation (discussed in Section 3) but also co-simulation-based optimization by producing emergent patterns. This trend has to be consolidated using a retrospective view to propose methodologies and frameworks using a black-box approach and extensibility.

## 5. DISCUSSION AND CONCLUSIONS

The path followed to review the topic of co-simulation-based optimization started with the consolidation of a sample database of 96 articles using the equation 1 to find articles in the literature that combine co-simulation with optimization, then we performed a study endeavor consisting of analyzing the articles from three different points of view.

First, we began with the definition of the concepts of co-simulation and optimization, and then we analyzed the collaboration of the concepts in an abstract manner. This analysis showed the co-simulation conceptual construction process (top-down and bottom-up approaches) and the subsequent conceptual integration with optimization using the black-box approach. This approach offered an abstract representation of the components of a system, which is compatible with the abstract schemes of co-simulation and optimization (Figures 6a and 5, respectively).

Second, we performed a multidisciplinary analysis of the database that showed the wide variety of domains cooperating in co-simulations, many domain-specific software are used for each sub-system which then needs to interact with the others. This led us to study the interoperability of co-simulations as it is a key factor in the performance of a complex system representation. The presence of many manual implementations has provoked the proposal of collaborative standards and frameworks as generalized methods that consolidate the understanding of co-simulation.

Finally, we performed an analysis of the database from the point of view of optimization that showed characteristics similar to the multidisciplinary analysis, that is, big variety and interoperability relevancy. Additionally, some robust frameworks found in the literature are starting to grasp a generalization of the co-simulation-based optimization process to be used in the study of complex systems.

In the process of developing this state-of-the-art research, we achieved several contributions relevant to co-simulation-based optimization, starting with the construction classification for co-simulations in Section 2, which showed how the reasoning approaches *bottom-up* and *top-down* can lead a complex system to be co-simulated. Next, the black-box approach was proposed as an intuitive abstraction tool to represent co-simulation and optimization, which provided a shared semantic environment to analyze the interoperability strategies found in the literature database.

The number of heterogeneous domains participating in this topic showed the relevance and applicability of the co-simulation-based optimization method for understanding and predicting complex systems, while also bringing attention to the way in which the components interact. This means that the collaboration of diverse domains implies an effort of interoperability, and based on the analysis done



over the literature, we proposed the classification into three types of interoperability strategies: manual implementations, software collaboration standards, and frameworks that offer a multi-domain shared environment for co-simulation.

From an optimization perspective, we found and classified the optimization methods used in the literature database (Figure 8). This classification allowed us to examine the characteristics of the interaction between the co-simulation and optimization components to determine the three types of interoperability strategies that are utilized in the literature to integrate co-simulation and optimization, that is, manually, by software collaboration, and using co-simulation-based optimization frameworks.

The main conclusion found in the reviewing process of this article is linked to the interest in capitalizing on the many applications by proposing frameworks that offer extensibility capabilities that make possible the integration of co-simulation and optimization tools. A step further could be to better formalize the black-box representation of figure 7 by the development of standards for both co-simulation and optimization interoperability on which frameworks can rely.

#### REFERENCES

1. Emmanuelle Abisset-Chavanne and Francisco Chinesta, *Toward an optimisation of the reactive resin transfer molding process: thermo-chemico-mechanical coupled simulations*, International journal of material forming **7** (2014), 249–258.
2. Mohd Asim Aftab, Astha Chawla, Pedro P. Vergara, Shehab Ahmed, and Charalambos Konstantinou, *Volt/var optimization in the presence of attacks: A real-time co-simulation study*, 2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), IEEE, 2023, pp. 1–7.
3. EOStTrack Engineering & Agencies, *Lingo software*, <https://www.eostrack.com/lingo#:~:text=OVERVIEW,faster%2C%20easier%20and%20more%20efficient.>, 2023, Accessed: 31.05.2024.
4. U.S. Environmental Protection Agency, *Moves software*, <https://www.epa.gov/moves/latest-version-motor-vehicle-emission-simulator-moves>, 2024, Accessed: 31.05.2024.
5. M. Ahmed, C. Oekermann, and F. Kirchner, *Cosimulation environment for mechanical design optimization with evolutionary algorithms*, World Symp. Comput. Appl. Res., WSCAR, Institute of Electrical and Electronics Engineers Inc., 2014, Journal Abbreviation: World Symp. Comput. Appl. Res., WSCAR, pp. 1–6.
6. Mohammed Ahmed, Yong-Ho Yoo, and Frank Kirchner, *A co-simulation framework for design, test and parameter optimization of robotic systems*, ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics), VDE, 2010, pp. 1–6.
7. Aiguasol, *Trnflow software*, <https://aiguasol.coop/energy-software/trnflow-ventilation-module-for-trnsys-18/?privacy=updated>, 2024, Accessed: 31.05.2024.
8. Eclipse Foundation AISBL., *Eclipse*, <https://www.eclipse.org/downloads/>, 2024, Accessed: 31.05.2024.
9. D Alonso-Alvarez, T Wilson, P Pearce, M Fuhrer, D Farrell, and N Ekins-Daukes, *Solcore software*, <https://www.solcore.solar/>, 2018, Accessed: 31.05.2024.
10. RTDS Technologies Inc AMETEK, *Rscad software*, <https://www.rtds.com/technology/graphical-user-interface>, 2024, Accessed: 31.05.2024.
11. ———, *Rtds software*, <https://www.rtds.com/>, 2024, Accessed: 31.05.2024.
12. Tom H Anderson, Benjamin J Civiletti, Peter B Monk, and Akhlesh Lakhtakia, *Coupled optoelectronic simulation and optimization of thin-film photovoltaic solar cells*, Journal of Computational Physics **407** (2020), 109242.
13. Hexagon AB and/or its subsidiaries, *Adams software*, <https://hexagon.com/products/product-groups/computer-aided-engineering-software/adams>, 2024, Accessed: 28.05.2024.

14. LLC Aquaveo, *Mt3dms software*, <https://www.aquaveo.com/software/gms-mt3dms>, 2021, Accessed: 31.05.2024.
15. Ivan Arsie, Ivan Criscuolo, Luigi De Simio, and Sabato Iannaccone, *Optimization of control parameters for a heavy-duty cng engine via co-simulation analysis*, Tech. report, SAE Technical Paper, 2011.
16. Erdem Arslan, Memduh Suveren, and Seyed Tofigh Hosseini Moghaddam, *Ros gazebo and matlab/simulink co-simulation for cart-pole system: A framework for design optimization*, 2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS), IEEE, 2023, pp. 1–7.
17. Seyed Amir Assadi, Omri Tayyara, Joshua Palumbo, Andrew Chen, Mohammad Shawkat Zaman, Carlos Da Silva, Sanjeev Chandra, Cristina H Amon, and Olivier Trescases, *Electro-thermal co-design of a high-density power-stage for a reconfigurable-battery-assisted electric-vehicle fast-charger using multi-physics co-simulation and topology optimization*, 2023 IEEE Applied Power Electronics Conference and Exposition (APEC), IEEE, 2023, pp. 1808–1815.
18. Rockwell Automation, *Arena software*, <https://www.rockwellautomation.com/en-us/products/software/arena-simulation.html>, 2024, Accessed: 28.05.2024.
19. AVL, *Model.connect software*, <https://www.avl.com/en/simulation-solutions/software-offering/simulation-tools-a-z/modelconnect>, 2024, Accessed: 05.06.2024.
20. Ghassane Ayad, J Song, Thierry Barriere, B Liu, and JC Gelin, *A fully coupled simulation and optimization scheme for the design of 3d powder injection molding processes*, AIP Conference Proceedings, vol. 908 (1), American Institute of Physics, 2007, pp. 507–512.
21. Emmanuel Balogun, Elizabeth Buechler, Siddharth Bhela, Simona Onori, and Ram Rajagopal, *Ev-ecosim: A grid-aware co-simulation platform for the design and optimization of electric vehicle charging infrastructure*, IEEE Transactions on Smart Grid **15** (2023), no. 3, 3114–3125.
22. Y.D. Barve, H. Neema, Z. Kang, H. Vardhan, H. Sun, and A. Gokhale, *Exppo: Execution performance profiling and optimization for cps co-simulation-as-a-service*, Journal of Systems Architecture **118** (2021), cited By 0.
23. Michael Stübert Berger, José Soler, Hao Yu, M Tsagkaropoulos, Y Leclerc, and C Olma, *Methodology and toolset for model verification, hardware/software co-simulation, performance optimisation and customisable source-code generation*, WSEAS Transactions on Information Science and Applications **6** (2013), no. 10, 169–178.
24. Guna R Bharati, Sudipta Chakraborty, and Julien Darrah, *A co-simulation-based hardware-in-the-loop architecture for validating power systems optimization with large-scale grid models*, 2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), IEEE, 2021, pp. 1–5.
25. Esther Borkowski, Mattia Donato, Giovanni Zemella, Dimitrios Rovas, and Rokia Raslan, *Optimisation of controller parameters for adaptive building envelopes through a co-simulation interface: A case study*, Proceedings of Building Simulation 2019: 16th Conference of IBPSA, International Building Performance Association (IBPSA), 2019, pp. 1372–1379.
26. Riccardo Bottura, Alberto Borghetti, Fabio Napolitano, and Carlo Alberto Nucci, *Ict-power co-simulation platform for the analysis of communication-based volt/var optimization in distribution feeders*, ISGT 2014, IEEE, 2014, pp. 1–5.
27. Stephen Boyd, Stephen P Boyd, and Lieven Vandenbergh, *Convex optimization*, Cambridge university press, 2004.
28. Christian Brecher, W Klein, and Marco Tannert, *Optimization of multi-stage closed-die forging processes by coupled simulation of the machine and the forging processes*, Production Engineering **4** (2010), 279–286.
29. David Chavalarias, Paul Bourguine, Edith Perrier, Frederic Amblard, Francois Arlabosse, Pierre Auger, Jean-Bernard Baillon, Olivier Barreteau, Pierre Baudot, Elisabeth Bouchaud, et al., *French roadmap for complex systems 2008-2009*, arXiv preprint arXiv:0907.2221 (2009).
30. Pengfei Chen, Petros Ioannou, and Maged Dessouky, *Mixed freight dynamic routing using a co-simulation optimization approach*, IEEE Transactions on Intelligent Transportation Systems **23** (2021), no. 8, 12833–12845.
31. Cloud Computing and Distributed Systems (CLOUDS) Laboratory, *ifogsim software*, <http://www.cloudbus.org/cloudsim>, 2017, Accessed: 31.05.2024.

32. COMSOL, *Comsol software*, <https://www.comsol.com/>, 2024, Accessed: 31.05.2024.
33. Itasca Consultants, *Particle flow code software*, <https://www.itasca.fr/en/software/pfc>, 2024, Accessed: 31.05.2024.
34. NATIONAL INSTRUMENTS CORP, *Labview software*, <https://www.ni.com/en/support/downloads/software-products/download/labview-control-design-and-simulation-module.html#460270>, 2024, Accessed: 31.05.2024.
35. Mechanical Simulation Corporation, *Carsim software*, <https://www.carsim.com/>, 2005, Accessed: 31.05.2024.
36. Giorgio Cucca and Anton Ianakiev, *Assessment and optimisation of energy consumption in building communities using an innovative co-simulation tool*, *Journal of Building Engineering* **32** (2020), 101681.
37. John Darvill, Alin Tisan, and Marcian Cirstea, *A novel psim and matlab co-simulation approach to particle swarm optimization tuning of pid controllers*, 2014 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), IEEE, 2014, pp. 784–789.
38. Irmen de Jong, *Pyro - python remote objects - 4.82*, <https://pyro4.readthedocs.io/en/stable/index.html>, 2024, Accessed: 23.07.2024.
39. Université de Lorraine and Inria, *Mecsyco software*, <http://www.mecsyco.com/>, 2017, Accessed: 03.07.2024.
40. W Deng, BQ Mao, BW Liang, and P Song, *Co-simulation of stabilization accuracy optimization of overhead weapon station*, 2015 International conference on Applied Science and Engineering Innovation, Atlantis Press, 2015, pp. 972–977.
41. Zhimin Du, Peifan Xu, Xinqiao Jin, and Qiaoling Liu, *Temperature sensor placement optimization for vav control using cfd-bes co-simulation strategy*, *Building and Environment* **85** (2015), 104–113.
42. Hagar Elarga, Andrea Dal Monte, Rune Korsholm Andersen, and Ernesto Benini, *Pv-pcm integration in glazed building. co-simulation and genetic optimization study*, *Building and Environment* **126** (2017), 161–175.
43. MACCON Bespoke electric motors and drive electronics, *Speed software*, <https://www.macon.com/cae-software/simcenter-speed.html>, 2024, Accessed: 31.05.2024.
44. Wael R Elwasif, David E Bernholdt, Sreekanth Pannala, Srikanth Allu, and Samantha S Foley, *Parameter sweep and optimization of loosely coupled simulations using the dakota toolkit*, 2012 IEEE 15th International Conference on Computational Science and Engineering, IEEE, 2012, pp. 102–110.
45. McNeel Europe., *Optimization solver with surrogate models software*, <https://www.food4rhino.com/en/app/opossum-optimization-solver-surrogate-models>, 2024, Accessed: 31.05.2024.
46. Maria Pia Fanti, Agostino Marcello Mangini, Michele Roccotelli, Francesco Iannone, and Alessandro Rinaldi, *A natural ventilation control in buildings based on co-simulation architecture and particle swarm optimization*, 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2016, pp. 002621–002626.
47. Maria Ferrara, Jean Christophe Vallee, Louena Shtrepi, Arianna Astolfi, and Enrico Fabrizio, *A thermal and acoustic co-simulation method for the multi-domain optimization of nearly zero energy buildings*, *Journal of Building Engineering* **40** (2021), 102699.
48. Collaborative Center for Research on Innovative Simulation Software, *Frontflow/blue software*, [http://www.ciss.iis.u-tokyo.ac.jp/project/rss/software/07\\_info.html](http://www.ciss.iis.u-tokyo.ac.jp/project/rss/software/07_info.html), 2024, Accessed: 03.06.2024.
49. C++ Foundation, *Standard c++*, <https://isocpp.org/>, 2024, Accessed: 28.06.2024.
50. Python Software Foundation, *python software*, <https://www.python.org/>, 2001, Accessed: 31.05.2024.
51. ELECTRICITE DE France, HYDRO-QUEBEC, and RTE RESEAU DE TRANSPORT D'ELECTRICITE, *Emtp-rv software*, <https://www.emtp.com/>, 2024, Now called EMTP, Accessed: 31.05.2024.
52. FreyrSCADA, *Dnp3 client master simulator*, <https://www.freyrscada.com/dnp3-ieee-1815-Client-Simulator.php>, 2024, Accessed: 31.05.2024.
53. LLC Gamma Technologies, *Gt-power software*, <https://www.gtisoft.com/gt-power/>, 2024, Accessed: 03.06.2024.

54. L. Giannantoni, R. Bardini, and S. Di Carlo, *A methodology for co-simulation-based optimization of biofabrication protocols*, Bioinformatics and Biomedical Engineering (Rojas I., Valenzuela O., Rojas F., Herrera L.J., and Ortuño F., eds.), vol. 13347 LNBI, Springer Science and Business Media Deutschland GmbH, 2022, Journal Abbreviation: Lect. Notes Comput. Sci., pp. 179–192 (English).
55. OptiY GmbH, *Optiy software*, <https://www.optiy.eu/default.htm>, 2005, Accessed: 31.05.2024.
56. PTV Planung Transport Verkehr GmbH, *Vissim software*, <https://www.ptvgroup.com/en/products/ptv-vissim>, 2024, Accessed: 31.05.2024.
57. ———, *Visum software*, <https://www.ptvgroup.com/en/products/ptv-visum>, 2024, Accessed: 31.05.2024.
58. Cláudio Gomes, Casper Thule, David Broman, Peter Gorm Larsen, and Hans Vangheluwe, *Co-simulation: A survey*, ACM Comput. Surv. **51** (2018), no. 3.
59. Abhijit Gosavi, *Simulation-based optimization*, Operations Research/Computer Science Interfaces Series, vol. 25, Springer US, 2003.
60. ESI Group, *Simulationx software*, <https://www.esi-group.com/products/simulationx>, 2024, Accessed: 31.05.2024.
61. Tooling Tech Group, *Unigraphics (ug) nx software*, <https://toolingtechgroup.com/stamping-design-division/capabilities/software/unigraphics-ug-nx/>, 2024, Accessed: 03.06.2024.
62. Xiangyu Gui and Xiaojing Fu, *Optimization of heat dissipation and control scheme for wind turbine gearbox based on flomaster-simulink co-simulation*, Journal of Physics: Conference Series, vol. 2584, IOP Publishing, 2023, p. 012049.
63. Tahar Hamiti, Chris Gerada, and Michael Rottach, *Weight optimisation of a surface mount permanent magnet synchronous motor using genetic algorithms and a combined electromagnetic-thermal co-simulation environment*, 2011 IEEE energy conversion congress and exposition, IEEE, 2011, pp. 1536–1540.
64. Dandan Han, Dongxing Zhang, Huirong Jing, Li Yang, Tao Cui, Youqiang Ding, Zhendong Wang, Yunxia Wang, and Tianliang Zhang, *Dem-cfd coupling simulation and optimization of an inside-filling air-blowing maize precision seed-metering device*, Computers and Electronics in Agriculture **150** (2018), 426–438.
65. Trevor D. Hardy, Bryan Palmintier, Philip L. Top, Dheepak Krishnamurthy, and Jason C. Fuller, *Helics: A co-simulation framework for scalable multi-domain modeling and analysis*, IEEE Access **12** (2024), 24325–24347.
66. Raaed Faleh Hassan, Ahmed Rashid Ajel, Saad Jabbar Abbas, and Amjad Jaleel Humaidi, *Fpga based hil co-simulation of 2dof-pid controller tuned by pso optimization algorithm*, ICIC Express Letters **16** (2022), no. 12, 1269–1278.
67. Haibin He, Xiaodong Ruan, Jie Wu, Fujian Zhao, Feng Wu, Lei Wang, and Jichong Yin, *Structure optimization of natural gas engine's intake pipe based on one-dimensional and three-dimensional coupled simulation*, IOP Conference Series: Earth and Environmental Science, vol. 781, IOP Publishing, 2021, p. 042029.
68. Franz Rudolf Holzinger and Martin Benedikt, *Hierarchical coupling approach utilizing multi-objective optimization for non-iterative co-simulation*, Proceedings of the 13th International Modelica Conference, Regensburg, Germany, March 4–6, 2019, no. 157, Linköping University Electronic Press, 2019.
69. Fengze Hou, Yunyan Zhou, Fengman Liu, Meiyong Su, Cheng Chen, Jun Li, Tingyu Lin, and Liqiang Cao, *Optimization design of 2.5 d tsv package using thermo-electrical co-simulation method*, 2016 IEEE 66th Electronic Components and Technology Conference (ECTC), IEEE, 2016, pp. 1964–1969.
70. Jianjun Hou, Shuxun Li, Weiliang Pan, and Lingxia Yang, *Co-simulation modeling and multi-objective optimization of dynamic characteristics of flow balancing valve*, Machines **11** (2023), no. 3, 337.
71. Xian Jun Hou, Shu Chen, and Zhi'en Liu, *Egr system performance optimization of diesel engine based on gt-power and fluent co-simulation*, Advanced Materials Research **860** (2014), 1703–1709.
72. Yushan Hou, Zongqian Shi, Sheng Li, Yingkui Zhang, Jun Bai, Shenli Jia, and Lijun Wang, *Co-simulation on the optimization design of high-speed electromagnetic repulsion mechanism*

- of vacuum circuit breaker*, 2016 27th International Symposium on Discharges and Electrical Insulation in Vacuum (ISDEIV), vol. 2, IEEE, 2016, pp. 1–4.
73. Chong Hu, Yuan Zhao, Zhe Zhang, Haiyang Zhang, and Daifen Chen, *Optimization of flow field structure for proton exchange membrane fuel cell stack by multi-physics coupling simulation*, International Journal of Electrochemical Science **18** (2023), no. 7, 100195.
  74. Hongji Hu, Zhonglin Zhou, Wencheng Wu, Wenhao Yang, Tao Li, Cheng Chang, Wanjun Ren, and Xiaolong Lei, *Distribution characteristics and parameter optimisation of an air-assisted centralised seed-metering device for rapeseed using a cfd-dem coupled simulation*, Biosystems Engineering **208** (2021), 246–259.
  75. Howard H. Hu, *Chapter 10 - computational fluid dynamics*, Fluid Mechanics (Fifth Edition) (Pijush K. Kundu, Ira M. Cohen, and David R. Dowling, eds.), Academic Press, Boston, fifth edition ed., 2012, pp. 421–472.
  76. Advanced Micro Devices Inc., *Xilinx software*, <https://www.xilinx.com/support/download.html>, 2024, Accessed: 31.05.2024.
  77. Altair Engineering Inc., *Altair edem software*, <https://altair.com/edem/features>, 2024, Accessed: 31.05.2024.
  78. ———, *Dshplus software*, <https://fluidon.com/en/tools/dshplus>, 2024, Accessed: 31.05.2024.
  79. ———, *Feko software*, <https://altair.com/feko>, 2024, Accessed: 31.05.2024.
  80. Altair Engineering Inc, *Psim software*, <https://altair.com/psim>, 2024, Accessed: 31.05.2024.
  81. Analog Devices Inc., *Ltspice software*, <https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html>, 2024, Accessed: 03.06.2024.
  82. ANSYS Inc, *Ansys software*, <https://www.ansys.com/>, 2024, Accessed: 28.05.2024.
  83. ———, *Aqua software*, <https://www.ansys.com/webinars/ansys-simulation-of-hydrodynamic-wave-loading-on-offshore-structures>, 2024, Accessed: 28.05.2024.
  84. ———, *Fluent software*, <https://www.ansys.com/products/fluids/ansys-fluent>, 2024, Accessed: 31.05.2024.
  85. ———, *Hfss software*, <https://www.ansys.com/products/electronics/ansys-hfss>, 2024, Accessed: 31.05.2024.
  86. ———, *Maxwell software*, <https://www.ansys.com/products/electronics/ansys-maxwell#:~:text=What%20is%20Ansys%20Maxwell%3F,varying%20magnetic%20and%20electric%20fields.>, 2024, Accessed: 31.05.2024.
  87. ———, *Modelcenter software*, <https://www.ansys.com/products/connect/ansys-modelcenter>, 2024, Accessed: 31.05.2024.
  88. ———, *Motor-cad software*, <https://www.ansys.com/products/electronics/ansys-motor-cad>, 2024, Accessed: 31.05.2024.
  89. ———, *Simplorer software*, <https://www.ansys.com/training-center/course-catalog/electronics/introduction-to-ansys-simplorer>, 2024, Accessed: 31.05.2024.
  90. Aspen Technology Inc, *Aspen plus software*, <https://www.aspentech.com/en/products/engineering/aspen-plus>, 2024, Accessed: 28.05.2024.
  91. Cadence Design Systems Inc, *Sigrity powerdc software*, [https://www.cadence.com/zh\\_CN/home/tools/ic-package-design-and-analysis/si-pi-analysis-point-tools/sigrity-powerdc.html](https://www.cadence.com/zh_CN/home/tools/ic-package-design-and-analysis/si-pi-analysis-point-tools/sigrity-powerdc.html), 2024, Accessed: 03.06.2024.
  92. ———, *Spectre software*, [https://www.cadence.com/en\\_US/home/tools/custom-ic-analog-rf-design/circuit-simulation/spectre-x-simulator.html](https://www.cadence.com/en_US/home/tools/custom-ic-analog-rf-design/circuit-simulation/spectre-x-simulator.html), 2024, Accessed: 31.05.2024.
  93. Docker Inc., *Docker*, <https://www.docker.com/>, 2024, Accessed: 31.05.2024.
  94. Dynamic Systems Inc, *Gleeble software*, <https://gleeble.com/>, 2024, Accessed: 31.05.2024.
  95. Electric Power Research Institute Inc, *OpenDSS software*, <https://www.epri.com/pages/sa/opendss>, 2001, Accessed: 31.05.2024.
  96. FunctionBay Inc, *Recurdyn software*, <https://functionbay.com/en/page/single/2/recurdyn-overview>, 2024, Accessed: 31.05.2024.
  97. MathWorks Inc., *Matlab software*, <https://www.mathworks.com/products/matlab.html>, 1994, Accessed: 31.05.2024.
  98. OPAL-RT Technologies Inc., *ephasorsim software*, <https://www.opal-rt.com/systems-ephasorsim/>, 2024, Accessed: 31.05.2024.

99. OPAL-RT TECHNOLOGIES Inc, *Opal-rt software*, <https://www.opal-rt.com/>, 2024, Accessed: 31.05.2024.
100. Robert McNeel & Associates (TLM Inc.), *Grasshopper*, <https://www.rhino3d.com/en/features/#grasshopper>, 1993, Accessed: 03.06.2024.
101. Robert McNeel & Associates TLM Inc, *Rhino 3d*, <https://www.rhino3d.com/>, 1993, Accessed: 03.07.2024.
102. Synopsys Inc, *Hspice software*, <https://www.synopsys.com/implementation-and-signoff/ams-simulation/primesim-hspice.html>, 2024, Accessed: 31.05.2024.
103. The MathWorks Inc, *Simulink software*, <https://www.mathworks.com/products/simulink-online.html>, 1994, Accessed: 31.05.2024.
104. VMS Software Inc., *Vsi fortran software*, <https://vmssoftware.com/products/fortran/>, 2014, Accessed: 31.05.2024.
105. Accellera Systems Initiative, *Systemc software*, <https://systemc.org/overview/systemc/>, 2024, Accessed: 31.05.2024.
106. Marco Inzillo and Carlos Kavka, *Multi-disciplinary optimization with standard co-simulation interfaces.*, ICSoft, 2017, pp. 453–458.
107. Khaled Ben Khoud, Soufiene Bouallègue, and Mounir Ayadi, *Design and co-simulation of a fuzzy gain-scheduled pid controller based on particle swarm optimization algorithms for a quad tilt wing unmanned aerial vehicle*, Transactions of the Institute of Measurement and Control **40** (2018), no. 14, 3933–3952.
108. Hannes Klarner, *Pyboolnet software*, <https://pyboolnet.readthedocs.io/en/master/installation.html>, 2024, Accessed: 31.05.2024.
109. Joshua M Kovitz, Vignesh Manohar, and Yahya Rahmat-Samii, *Feed horn optimization using feed+ reflector co-simulation for advanced reflector antennas*, 2017 International Applied Computational Electromagnetics Society Symposium-Italy (ACES), IEEE, 2017, pp. 1–2.
110. Slawomir Kozie and John W Bandler, *Fast design optimization of microwave structures using co-simulation-based tuning space mapping*, Applied Computational Electromagnetics Society Journal **26** (2011), no. 8, 631.
111. Slawomir Koziel, *Robust optimization of microwave structures using co-simulation-based surrogate models*, 2011 IEEE International Symposium on Antennas and Propagation (AP-SURSI), IEEE, 2011, pp. 2924–2927.
112. Piotr Kurgan, Slawomir Koziel, Leifur Leifsson, and Xiaosong Du, *Expedite design of variable-topology broadband hybrid couplers for size reduction using surrogate-based optimization and co-simulation coarse models*, Procedia Computer Science **108** (2017), 1483–1492.
113. Lawrence Berkeley National Laboratory, *Genopt software*, <https://simulationresearch.lbl.gov/G0/>, 2021, Accessed: 03.06.2024.
114. Averill M Law and W David Kelton, *Simulation modeling and analysis*, vol. 3, McGraw-hill New York, 2000.
115. Daniel Lee, Ming Chong Lim, Lebsework Negash, and Han-Lim Choi, *Eppy based building co-simulation for model predictive control of hvac optimization*, 2018 18th International Conference on Control, Automation and Systems (ICCAS), IEEE, 2018, pp. 1051–1055.
116. Yifeng Li, Xunpeng Qin, Zhenyuan Zhang, and Huanyu Dong, *Operation parameters optimization of a separating system for non-ferrous metal scraps from end-of-life vehicles based on coupled simulation*, Waste Management **120** (2021), 667–674.
117. Michael Loupis, *A toolset for the design of embedded systems, enabling hw/sw co-simulation, performance optimisation and source code generation*, 2017 Panhellenic Conference on Electronics and Telecommunications (PACET), IEEE, 2017, pp. 1–4.
118. LS-DYNA, *Ls-dyna software*, <https://lsdyna.ansys.com/>, 2023, Accessed: 31.05.2024.
119. Hongyu Lu, Huapeng Zhao, Anfeng Huang, Dongwook Kim, Jingdong Sun, Jun Hu, and Hongseok Kim, *High power wireless power transfer efficiency and emi co-optimization based on fast field-circuit co-simulation*, 2019 IEEE International Conference on Computational Electromagnetics (ICCEM), IEEE, 2019, pp. 1–3.
120. Yuanbo Ma, Bin Li, Zhaohui Wu, Haigang Wu, and Zhijian Chen, *A co-simulation method of power amplifier for reliability optimization*, 2018 IEEE International Conference on Electron Devices and Solid State Circuits (EDSSC), IEEE, 2018, pp. 1–2.

121. M. Manbachi, A. Sadu, H. Farhangi, A. Monti, A. Palizban, F. Ponci, and S. Arzanpour, *Real-time co-simulation platform for smart grid volt-var optimization using iec 61850*, IEEE Transactions on Industrial Informatics **12** (2016), no. 4, 1392–1402, cited By 56.
122. M Manbachi, A Sadu, H Farhangi, Antonello Monti, Ali Palizban, Ferdinanda Ponci, and Siamak Arzanpour, *Impact of ev penetration on volt-var optimization of distribution networks using real-time co-simulation monitoring platform*, Applied Energy **169** (2016), 28–39.
123. G Manzi and U Muehlmann, *Passive uhf rfid sensor/transponder antenna optimization for backscatter operation by electromagnetic-circuitual co-simulation*, Proceedings of the 11th International Conference on Telecommunications, IEEE, 2011, pp. 17–22.
124. Hongyong Mao, Manjun Jing, Zhuo Fan, Jingjing Lu, Xin Xiao, Zhen Fang, Zihan Liu, Minghao Chen, and Yong Jiang, *A multi-body structure optimization and fully coupled hydrodynamic simulation method for marine semi-submersible equipment*, 11 2023.
125. Jean-Pierre Massat, Christophe Laurent, Jean-Philippe Bianchi, and Etienne Balmès, *Pantograph catenary dynamic optimisation based on advanced multibody and finite element co-simulation tools*, Vehicle System Dynamics **52** (2014), no. sup1, 338–354.
126. Patrick J McCurdy, Kaleb Pattawi, Chenli Wang, Thomas Roth, Cuong Nguyen, Yuhong Liu, and Hohyun Lee, *Validation approach for energy optimization models of grid-interactive buildings using co-simulation*, ASME International Mechanical Engineering Congress and Exposition, vol. 85642, American Society of Mechanical Engineers, 2021, p. V08BT08A041.
127. Marvin Minsky, *Matter, mind and models*, (1965).
128. Seyed Masoud Mohseni-Bonab, Ali Hajebrahimi, Ali Moeini, and Innocent Kamwa, *Optimization application in integrated transmission and distribution operation: Co-simulation approach*, 2020 IEEE Power & Energy Society General Meeting (PESGM), IEEE, 2020, pp. 1–5.
129. Jinchao Mou and Zhongxiang Shen, *Co-simulation and co-optimization strategy for active absorber with periodic structure*, 2017 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS), IEEE, 2017, pp. 1–4.
130. CAPE OPEN LABORATORIES NETWORK, *Advanced process engineering co-simulator (apecs)*, <https://www.colan.org/experiences-projects/advanced-process-engineering-co-simulator-apecs/>, 2001, Accessed: 28.05.2024.
131. Khadije Norouzi Khatiri, Mohammad Hossein Niksokhan, Amin Sarang, and Asghar Kamali, *Coupled simulation-optimization model for the management of groundwater resources by considering uncertainty and conflict resolution*, Water Resources Management **34** (2020), no. 11, 3585–3608.
132. Kazunori Nozaki, Hiroyuki Ohsaki, Masao Tanaka, Annemie Vanhirtum, and Eisaku Sakane, *Strategic optimization of computation nodes allocation in a coupled simulation: Computational fluid dynamics and aero acoustic simulations*, 2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet, IEEE, 2012, pp. 211–214.
133. nsnam, *ns-3 software*, <https://www.nsnam.org/>, 2011, Accessed: 31.05.2024.
134. College of Agriculture, Life Sciences, Texas A&M AgriLife Extension Service, and Texas A&M AgriLife Research, *Swat software*, <https://swat.tamu.edu/software/>, 2024, Accessed: 31.05.2024.
135. United States Department of Agriculture, *Annagnps software*, [https://www.appolutelydigital.com/ModelPrimer/chapter2\\_section1.html](https://www.appolutelydigital.com/ModelPrimer/chapter2_section1.html), 2010, Accessed: 28.05.2024.
136. Laboratory of Computational Mechanics, *Universal mechanism software*, <https://universalmechanism.com/en/pages/index.php?id=1>, 2024, Accessed: 31.05.2024.
137. U.S. Department of Energy, *Gridlab-d software*, <https://www.gridlabd.org/>, 2023, Accessed: 31.05.2024.
138. U.S. Department of Energy’s and Building Technologies Office, *Energyplus software*, <https://energyplus.net/>, 2021, Accessed: 31.05.2024.
139. National Institute of Standards and Technology, *Ucef: Universal cps environment for federation software*, <https://www.nist.gov/ctl/smart-connected-systems-division/iot-devices-and-infrastructures-group/ucef-universal-cps>, 2021, Accessed: 31.05.2024.
140. U.S. Department of the Interior, *Modflow*, <https://www.usgs.gov/mission-areas/water-resources/science/modflow-and-related-programs>, 2022, Accessed: 31.05.2024.

141. U. Oh, N. Nonaka, and J. Ishimoto, *Valve optimization with system-fluid-magnetic co-simulation and design of experiments*, **23**, no. 3, 683–692, Publisher: Korean Society of Automotive Engineers.
142. OpenModelica, *Openmodelica software*, <https://openmodelica.org/>, 2024, Accessed: 31.05.2024.
143. Oracle, *Java software*, <https://www.java.com/en/>, 2024, Accessed: 31.05.2024.
144. Peter Palensky, Arjen van der Meer, Claudio Lopez, Arun Joseph, and Kaikai Pan, *Applied cosimulation of intelligent power systems: Implementing hybrid simulators for complex power systems*, IEEE Industrial Electronics Magazine **11** (2017), no. 2, 6–21.
145. Jong-Soo Park, Seung-Woo Kim, Hyung-Cheol Lim, and Jong-Hun Kang, *Flow stress optimization of inconel 718 based on a coupled simulation of material-forming analysis and joule heating analysis*, Metals **12** (2022), no. 12, 2024.
146. Modelica Association Project, *Fmi standard*, <https://fmi-standard.org/>, 2024, Accessed: 31.05.2024.
147. SWIM Project, *Integrated plasma simulator*, <https://ipsframework.sourceforge.net/doc/html/>, 2011, Accessed: 31.05.2024.
148. OPNET SIMULATOR PROJECTS, *Opnet software*, <https://opnetprojects.com/opnet-network-simulator/>, 2005, Accessed: 31.05.2024.
149. PTC, *Proengineer software*, <https://www.ptc.com/en/products/creo/pro-engineer>, 2024, Accessed: 31.05.2024.
150. Arash Rahnema, Meng Xia, Shige Wang, and Panos J Antsaklis, *Passivation and performance optimization using an extremum seeking co-simulation framework with application to adaptive cruise control systems*, 2016 American Control Conference (ACC), IEEE, 2016, pp. 6109–6114.
151. Remcom, *Mom software*, <https://www.remcom.com/resources/articles-and-papers/3d-electromagnetic-simulation-vs-planar-mom>, 2024, Accessed: 31.05.2024.
152. Open Robotics, *Gazebo software*, <https://gazebo.org/home>, 2002, Accessed: 31.05.2024.
153. ———, *Ros: Robot operating system*, <https://www.ros.org/>, 2021, Accessed: 22.07.2024.
154. Andreas Rüdener, Song Han, Marcus Geimer, et al., *Optimization of the development process of mobile machines using a standardized co-simulation.*, Landtechnik **67** (2012), no. 2, 122–126.
155. Rabayet Sadnan, Nathan Gray, Anamika Dubey, and Anjan Bose, *Distributed optimization for power distribution systems with cyber-physical co-simulation*, 2021 IEEE Power & Energy Society General Meeting (PESGM), IEEE, 2021, pp. 1–5.
156. Reinhard Schemmel, Viktor Krieger, Tobias Hensel, and Walter Sestro, *Co-simulation of matlab and ansys for ultrasonic wire bonding process optimization*, Microelectronics Reliability **119** (2021), 114077.
157. A. Schirrer and M. Kozek, *Co-simulation as effective method for flexible structure vibration control design validation and optimization*, 2008, pp. 481–486.
158. Tobias Schulze, Jürgen Weber, Knut Großmann, Lars Penter, and Christer Schenke, *Hydraulic die cushions in deep drawing presses: Analysis and optimization using coupled simulation*, Fluid Power Systems Technology, vol. 57236, American Society of Mechanical Engineers, 2015, p. V001T01A066.
159. Antonio Ciccomancini Scogna and LianKheng Teoh, *Sipi co-extraction and spice co-simulation for package on-die decap optimization*, 2016 IEEE 20th Workshop on Signal and Power Integrity (SPI), IEEE, 2016, pp. 1–4.
160. SDS, *Sdm software*, <https://systemdynamics.org/resources-old/sdm-doc/>, 2024, Accessed: 31.05.2024.
161. SDTools, *Oscar software*, <https://www.sdtools.com/solutions/oscar/>, 2002, Accessed: 03.06.2024.
162. Siemens, *Simcenter flomaster software*, <https://plm.sw.siemens.com/es-ES/simcenter/systems-simulation/flomaster/>, 2024, Accessed: 31.05.2024.
163. Jan A Snyman, *Practical mathematical optimization*, Springer, 2005.
164. INTEGRATED Engineering Software, *Amperes software*, <https://www.integratedsoft.com/products/Amperes>, 2021, Accessed: 28.05.2024.
165. Siemens Digital Industries Software, *Amesim software*, <https://plm.sw.siemens.com/en-US/simcenter/systems-simulation/amesim/>, 2024, Accessed: 28.05.2024.



166. Anshu Soni and Virender Ranga, *Api features individualizing of web services: Rest and soap*, International Journal of Innovative Technology and Exploring Engineering **8** (2019), no. 9, 664–671.
167. ESTECO SpA, *modefrontier software*, <https://engineering.esteco.com/modefrontier/>, 2024, Accessed: 31.05.2024.
168. Dirk Spenneberg, Martin Albrecht, and Till Backhaus, *Monster: A new behavior-based microkernel for mobile robots*, Proceedings of the ECMR **2005** (2005).
169. Giovanni Squillero, *Microgp3 software*, <https://ugp3.sourceforge.net/>, 2006, Accessed: 31.05.2024.
170. Hongbiao Sun, Wenqiang Li, Lanjiang Zheng, Sitong Ling, and Wanchang Fu, *Adaptive co-simulation method and platform application of drive mechanism based on fruit fly optimization algorithm*, Progress in Nuclear Energy **153** (2022), 104397.
171. Lu Sun, RuXin Zhang, ChaoLing Du, WangXu Rong, Xiang Li, YangXi Chen, TianYi Fu, ShuiYan Cao, and DaNing Shi, *Optical optimization of ultra-thin crystalline silicon solar cells by a co-simulation approach of fem and ga*, Applied Physics A **127** (2021), no. 7, 1–9.
172. Wei Sun and Shi Sha Zhu, *The mechanical excavator-hydraulic system of the establishment of co-simulation and optimization*, Advanced Materials Research, vol. 706, Trans Tech Publ, 2013, pp. 1651–1656.
173. Dassault Systèmes, *Abaqus software*, <https://www.3ds.com/products/simulia/abaqus>, 2002, Accessed: 28.05.2024.
174. ———, *Cst studio suite*, <https://www.3ds.com/products/simulia/cst-studio-suite>, 2002, Accessed: 31.05.2024.
175. ———, *Dymola software*, <https://www.3ds.com/products/catia/dymola>, 2002, Accessed: 31.05.2024.
176. ———, *Isight software*, <https://www.3ds.com/products/simulia/isight>, 2002, Accessed: 31.05.2024.
177. ———, *Simpack software*, <https://www.3ds.com/products/simulia/simpack>, 2002, Accessed: 31.05.2024.
178. ———, *Solidworks software*, <https://www.solidworks.com/>, 2002, Accessed: 03.06.2024.
179. Jiatong Tan, Quan Li, Bo Zhao, Chao Ma, Yingwei Wu, Yuanming Li, Jing Zhang, Yanan He, Tianzhou Ye, Cheng Zhang, et al., *Multi-objective optimization design of plate-type fuel via co-simulation method*, Annals of Nuclear Energy **169** (2022), 108914.
180. Wuzhong Tan, Zhi Chen, Zhizuo Li, and Hongzhi Yan, *Thermal-fluid-solid coupling simulation and oil groove structure optimization of wet friction clutch for high-speed helicopter*, Machines **11** (2023), no. 2, 296.
181. Keysight Technologies, *Ads software*, <https://www.keysight.com/us/en/products/software/pathwave-design-software/pathwave-advanced-design-system.html>, 2000, Accessed: 28.05.2024.
182. ———, *Exata cps software*, <https://www.keysight.com/us/en/product/SN050ECPA/exata-network-modeling-critical-infrastructure.html>, 2000, Accessed: 31.05.2024.
183. National Technology and LLC. Engineering Solutions of Sandia, *Dakota software*, <https://dakota.sandia.gov/>, 2024, Accessed: 31.05.2024.
184. Andreas Tolk, Saikou Y Diallo, and Charles D Turnitsa, *Applying the levels of conceptual interoperability model in support of integratability, interoperability, and composability for system-of-systems engineering*, Journal of Systems, Cybernetics, and Informatics **5** (2007), no. 5.
185. TRANSVALOR, *Forge software*, <https://www.transvalor.com/en/forge>, 2024, Accessed: 31.05.2024.
186. TRNSYS, *Transient system simulation tool software*, <https://www.trnsys.com/>, 2019, Accessed: 31.05.2024.
187. Shreshth Tuli, Shivananda R Poojara, Satish N Srirama, Giuliano Casale, and Nicholas R Jennings, *Cosco: Container orchestration using co-simulation and gradient based optimization for fog computing environments*, IEEE Transactions on Parallel and Distributed Systems **33** (2022), no. 1, 101–116.
188. Bavantha Udugama, *Mini bot 3d: A ros based gazebo simulation*, arXiv preprint arXiv:2302.06368 (2023).

189. Diego Vega and Vincent Chevrier, *A framework for co-simulation based optimization*, Proceedings of the 35th European Modeling & Simulation Symposium (EMSS 2023) Article 024, CALTEK, 2023, p. 10.
190. Philipp Von Hilgers, *The History of the Black Box: The Clash of a Thing and Its Concept*, Cultural Politics **7** (2011), no. 1, 41–58 (en).
191. Christoph Waibel, Ralph Evins, and Jan Carmeliet, *Co-simulation and optimization of building geometry and multi-energy systems: Interdependencies in energy supply, energy demand and solar potentials*, Applied Energy **242** (2019), 1661–1682.
192. Yingbo Wang, Hongwen Li, Hongnan Hu, Jin He, Qingjie Wang, Caiyun Lu, Peng Liu, Dong He, and Xin Lin, *Dem-cfd coupling simulation and optimization of a self-suction wheat shooting device*, Powder Technology **393** (2021), 494–509.
193. Yong Wang, Linlin Wang, Chenchen Li, Ziao Xue, Yuan Sun, Ruiyong Ma, Dagang Wang, Muchun Cui, Xuefeng Wei, Liang Tang, et al., *Optimization of excavator bucket structure by a coupled simulation method*, Applied Sciences **13** (2023), no. 20, 11336.
194. Yuanjie Wang, Xueliang Fan, and Qilin Cai, *Optimization for ev fuse based on thermal-electrical coupled simulation*, 2022 7th International Conference on Power and Renewable Energy (ICPRE), IEEE, 2022, pp. 786–790.
195. Nadja Wirth, Pascal Bayrasy, Bettina Landvogt, Klaus Wolf, Francesco Cecutti, and Tomasz Lewandowski, *Analysis and optimization of flow around flexible wings and blades using the standard co-simulation interface mpcci*, Recent Progress in Flow Control for Practical Flows: Results of the STADYWICO and IMESCON Projects (2017), 283–321.
196. Hao Xu, Jiuzhou Wang, Zai-Fa Zhou, Zhenxiang Yi, Ming Qin, and Qing-An Huang, *System-level optimization of mems thermal wind sensor based on the co-simulation of macromodel and board-level interface circuits*, Journal of Microelectromechanical Systems (2023).
197. Mikael Yamanee-Nolin, Anton Löfgren, Niklas Andersson, Bernt Nilsson, Mark Max-Hansen, and Oleg Pajalic, *Single-shooting optimization of an industrial process through co-simulation of a modularized aspen plus dynamics model*, Computer Aided Chemical Engineering, vol. 46, Elsevier, 2019, pp. 721–726.
198. Xinqiang Yan, ChuangXin Ma, Lei Shi, Yan Zhuo, Xiaohong Joe Zhou, Long Wei, and Rong Xue, *Optimization of an 8-channel loop-array coil for a 7 t mri system with the guidance of a co-simulation approach*, Applied Magnetic Resonance **45** (2014), 437–449.
199. Shuai Yang, Ahmad Khusro, Weiwei Li, Mohammad Vaseem, Mohammad Hashmi, and Atif Shamim, *Optimization of ann-based models and its em co-simulation for printed rf devices*, International Journal of RF and Microwave Computer-Aided Engineering **32** (2022), no. 3, e23012.
200. Bernard P Zeigler, Herbert Praehofer, and Tag Gon Kim, *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*, Academic press, 2000.
201. Baichuan Zhang, Huaicheng Guo, Yang Zhang, Zheng Li, Yanxiao Liu, Shuhang Wang, and Zhenghui Fu, *A coupling simulation and optimization method developed for environmental-economic management of lake watershed*, Journal of Environmental Management **318** (2022), 115546.
202. Lei Zhang, *System generator model-based fpga design optimization and hardware co-simulation for lorenz chaotic generator*, 2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), IEEE, 2017, pp. 170–174.
203. Qihang Zhang, Qinli Deng, Xiaofang Shan, Xin Kang, and Zhigang Ren, *Optimization of the thermal environment of large-scale open space with subzone-based temperature setting using bem and cfd coupling simulation*, Energies **16** (2023), no. 7, 3214.
204. Shuifeng Zhang, Jinchi Zhang, Miaoqing Meng, Peixian Chen, Xin Liu, Guoliang Liu, and Zheyang Gu, *A multi-objective decision making system (mdms) for a small agricultural watershed based on meta-heuristic optimization coupling simulation*, Water **13** (2021), no. 10, 1338.
205. J. Zhao, Y. Lu, X. Wang, J. Zhuang, and Z. Han, *A bionic profiling-energy storage device based on MBD-DEM coupled simulation optimization reducing the energy consumption of deep loosening*, **234**, Publisher: Elsevier B.V.
206. Y. Zhao and P. Ioannou, *A co-simulation, optimization, control approach for traffic light control with truck priority*, Annual Reviews in Control **48** (2019), 283–291, cited By 3.

207. Yanbo Zhao, Petros Ioannou, and MM Maged, *Routing of multimodal freight transportation using a co-simulation optimization approach*, Transportation Research Board 96th Annual Meeting, 2017.
208. Yanbo Zhao, Petros A Ioannou, and Maged M Dessouky, *A hierarchical co-simulation optimization control system for multimodal freight routing*, 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2017, pp. 1–6.
209. ———, *Dynamic multimodal freight routing using a co-simulation optimization approach*, IEEE Transactions on Intelligent Transportation Systems **20** (2019), no. 7, 2657–2667.
210. Zeming Zhao, Kai Wei, Wenhao Ding, Wei Du, and Huailong Li, *Um-simulink co-simulation for the vibration reduction optimization of a magnetorheological damping steel-spring floating slab track*, Construction and Building Materials **307** (2021), 124923.
211. Qin Zhaoju, Li Yingsong, Yang Zhenzhong, Duan Junfa, and Wang Lijun, *Diesel engine piston thermo-mechanical coupling simulation and multidisciplinary design optimization*, Case Studies in Thermal Engineering **15** (2019), 100527.
212. B Zheng, SY Fu, and JL Lei, *Titanium alloy piston thermo-mechanical coupling simulation and multi-objective optimization design*, **14**, no. 4.
213. Dawei Zhou, Li Lu, and Changqing Yang, *Optimization of high speed permanent magnet synchronous motor based on co-simulation method*, 2023 5th Asia Energy and Electrical Engineering Symposium (AEEES), IEEE, 2023, pp. 460–465.
214. Xiangyang Zhou, Beilei Zhao, and Guohao Gong, *Control parameters optimization based on co-simulation of a mechatronic system for an ua-based two-axis inertially stabilized platform*, Sensors **15** (2015), no. 8, 20169–20192.
215. Stephen E Zitney, *Advanced co-simulation for computer-aided process design and optimization of fossil energy systems with carbon capture*, Design for Energy and the Environment, CRC Press, 2009, pp. 211–228.

## APPENDIX A. CLASSIFICATION TABLE

TABLE 5. Literature sample classification (part 1 of 2).

Article	Approach	Simulation domains	Simulation software	Optimization method	Opt. Software
[2]	Top-down	Power, control, communication	OPAL-RT, DNP3, EXATACPS	VVO	MATLAB
[180]	Top-down	Fluid, solid	FLUENT, Solidworks	NNIA	FLUENT
[196]	Top-down	Fluid, electric	Spectre, LTspice	Ranking and selection	Non
[16]	Top-down	Robotic, dynamics	Gazebo, Simulink	Evolutionary Algorithms	MATLAB
[203]	Bottom-up	Fluid, building	EnergyPlus, FLUENT	Ranking and selection	Non
[62]	Bottom-up	Fluid, control	FloMaster, Simulink	Ranking and selection	Non
[73]	Bottom-up	Fluid, solid	Solidworks, FLUENT	Ranking and selection	Non
[193]	Bottom-up	Solid, dynamic	EDEM, Adams	Ranking and selection	Non
[21]	Top-down	Power, control	GridLAB-D, python	Genetic algorithm	Python
[17]	Top-down	Electric, fluid	COMSOL, FLUENT	IPOPT, SNOPT	COMSOL
[70]	Top-down	Mathematical, fluid	Simulink, FLUENT	NSGA-II	MATLAB
[124]	Top-down	Hydraulic, electrical, control	AQWA, ANSYS	PSO	MATLAB
[189]	Top-down	Control, electrical, numerical	OpenModelica, Java	Gradient-based	Java
[205]	Top-down	Soil, solid	EDEM, RecurDyn	Ranking and selection	Non
[141]	Bottom-up	System, fluid, magnetic	Non*	Ranking and selection	Non
[199]	Top-down	Solid, circuit	ADS	ANN	Python
[194]	Top-down	Solid, fluid, electrical	COMSOL	Ranking and selection	Non
[179]	Top-down	Fluid, solid	COMSOL	CMOEA-MS	MATLAB
[30]	Bottom-up	Traffic, analysis, emissions	VISUM, MOVES	Ranking and selection	Non
[66]	Top-down	Hardware/Software, Control	Xilinx, Simulink, MATLAB	PSO	MATLAB
[145]	Top-down	Fluid, mechanical	Gleeble	Ranking and selection	Non
[187]	Top-down	IoT, fog computing	iFogSim, python, Docker	Ranking and selection	Python
[170]	Top-down	Multi-disciplinary	ANSYS	GOBI	Python
[54]	Top-down	Boolean networks	Python, Pyboonnet	FOA	Isight
[201]	Top-down	Dynamics, soil and water	SWAT, SDM	Genetic algorithms	MicroGP
[126]	Bottom-up	Building, control	UCeF, EnergyPlus	Ranking and selection	Lingo
[210]	Bottom-up	Dynamics, mechanical, numerical	UM, Simulink, Abaqus	MPC	Python
[67]	Bottom-up	Engine, solid	GT-POWER, FLUENT	Ranking and selection	Non
[171]	Bottom-up	Optics, solid	MATLAB, COMSOL	LWM	Non**
[116]	Top-down	Numerical, fluid	Particle Flow Code, FLUENT	Genetic algorithm	MATLAB
[22]	Top-down	General	Python, API REST, Docker	RSM	ANSYS
[74]	Bottom-up	Solid, fluid, dynamic	ANSYS, Fluent, EDEM	FFD, BFD	Python
[155]	Top-down	Power, communication, control	GridLAB-D, NS-3, python	Ranking and selection	Python
[192]	Bottom-up	Solid, fluid	EDEM, FLUENT	BCD	Python
[156]	Bottom-up	Dynamics, mechanical, numerical	MATLAB, ANSYS	Ranking and selection	Non
[47]	Bottom-up	Energy, sound	TRNSYS, MATLAB	PSO	GenOpt
[204]	Bottom-up	Hydrological, numerical	AnnAGNPS, C++	TS algorithm	C++
[24]	Top-down	Real-time, computer	OPAL-RT, ePHASORSIM	Simplex	MATLAB
[128]	Top-down	Power, electric	MATLAB, OpenDSS	Genetic algorithm	MATLAB
[131]	Bottom-up	Groundwater, pollution, hydrological	MODFLOW, MT3DMS, SWAT	Multi-objective PSO	MATLAB
[12]	Top-down	Photonic, electronic	Solcore, Python	DEA	MATLAB
[36]	Top-down	Control, building	Dymola, Energyplus	Ranking and selection	Non
[197]	Top-down	Solid, dynamic	Aspen plus	COBYLA	Python
[25]	Top-down	Control, building	Dymola, Energyplus	Genetic algorithm	Python
[119]	Bottom-up	Circuit, solid	Simplorer, Maxwell	Ranking and selection	Non
[68]	Bottom-up	Electrical, dynamics	MATLAB, Simulink	Ranking and selection	Non
[209]	Top-down	Transport planning	VISUM, ARENA	Control meta-heuristic	MATLAB
[211]	Top-down	Solid, fluid	Pro ENGINEER, FLUENT	GDO	ANSYS

TABLE 6. Literature sample classification (part 2 of 2).

Article	Approach	Simulation domains	Simulation software	Optimization method	Opt. Software
[191]	Top-down	Building, Energy	EnergyPlus, Grasshopper	RBFOpt	Opossum
[206]	Top-down	Transport planning	VISSIM	Control meta-heuristic	MATLAB
[115]	Top-down	Building, control	EnergyPlus, MATLAB	MPC	MATLAB
[107]	Bottom-up	Physic, control	LabView	PSO	LabView
[64]	Bottom-up	Fluid, solid	EDEM, FLUENT	Ranking and selection	Non
[120]	Bottom-up	Numerical, circuit	Non *	Ranking and selection	Non
[202]	Bottom-up	Numerical, circuits	Simulink, Xilinx	Ranking and selection	Non
[42]	Bottom-up	Numerical, energy	MATLAB, TRNSYS	Ranking and selection	Non
[106]	Top-down	General	OpenModelica	Genetic algorithm	MATLAB
[109]	Bottom-up	Numerical, optics	Non *	Genetic algorithm	ModeFRONTIER
[112]	Top-down	Electromagnetic, circuit	ADS	Ranking and selection	Non
[129]	Bottom-up	Circuit	HFSS, MoM	Surrogate opt.	ADS
[195]	Top-down	Solid, fluid	FLUENT, MATLAB	Ranking and selection	Non
[117]	Top-down	Hardware/Software, language	SystemC, Simulink, Eclipse	Pareto frontier	ModeFRONTIER
[208]	Top-down	Transport planning	VISUM, ARENA	Ranking and selection	Eclipse
[159]	Bottom-up	Circuit, signal, power	HSPICE, CST	Control meta-heuristic	MATLAB
[121]	Bottom-up	Electric	RSCAD	Ranking and selection	Non
[150]	Top-down	Vehicle, control	CarSim, Simulink	VVO	MATLAB
[69]	Bottom-up	Thermal, electrical	Sigrity PowerDC	EXTREMUM-SEEKING	Simulink
[122]	Bottom-up	Network, Control	RTDS, MATLAB	Ranking and selection	Non
[72]	Top-down	Mechanical, electromagnetic	ANSYS	VVO	MATLAB
[46]	Top-down	Building, ventilation	TRNSYS, TRNFLOW	Ranking and selection	Non
[41]	Top-down	Energy, fluid	TRNSYS, FLUENT	PSO	MATLAB
[158]	Bottom-up	Mechanical, hydraulic, control	SimulationX, LS-DYNA	Ranking and selection	Non
[214]	Bottom-up	Solid, mechanical	Solidworks, Adams, Simulink	Control meta-heuristic	OptiY
[40]	Bottom-up	Mechanical, numerical	Adams, Simulink	Ranking and selection	Simulink
[1]	Top-down	Mechanical, chemical, thermal	Non *	Sensitivity analysis	ModelCenter
[125]	Top-down	Mechanical, numerical	OSCAR, Adams	Ranking and selection	Non
[198]	Top-down	Electromagnetic, network	HFSS	Ranking and selection	Non
[26]	Bottom-up	Electromagnetic, network	EMTP-RV, Opnet	Ranking and selection	Non
[5]	Top-down	Mechanical, physic	Solidworks, Adams	Ranking and selection	Non
[37]	Top-down	Electric, control	PSIM, Simulink	Ranking and selection	Non
[172]	Bottom-up	Mechanical, hydraulic	Adams, UG	VVO	Opnet
[23]	Top-down	Hardware/Software, language	SystemC, Simulink, Eclipse	CMA-ES	MATLAB
[71]	Bottom-up	Engine, fluid	GT-Power, FLUENT	PSO	MATLAB
[132]	Top-down	Numerical, fluid	VSI Fortran, FrontFlow/Blue	Ranking and selection	Non
[44]	Bottom-up	Fluid, electrochemical	AMPERES, IPS	Ranking and selection	Non
[154]	Bottom-up	Mechanical, control, hydraulic	Simpack, Simulink, AMESim, DSHPlus	Parametric opt.	DAKOTA
[63]	Bottom-up	Electromagnetic, thermal	SPEED, MotorCAD	Ranking and selection	Non
[111]	Bottom-up	Electromagnetic, circuit	FEKO, ADS	Genetic algorithm	MATLAB
[123]	Top-down	Electromagnetic, circuit	Not mentioned*	Surrogate opt.	FEKO
[15]	Top-down	Thermodynamic, solid	GT-Power	Ranking and selection	Non
[110]	Bottom-up	Electromagnetic, circuit	FEKO, ADS	Ranking and selection	Simulink
[28]	Bottom-up	Forging, mechanical	Forge	Surrogate opt.	FEKO
[6]	Top-down	Robotic, control	MONSTER, Simulink	Ranking and selection	Non
[215]	Top-down	Fluid, chemical	FLUENT, Aspen Plus	Ranking and selection	Non
[157]	Top-down	Solid, numerical	Simpack, Simulink	Parametric opt.	APECS
[20]	Bottom-up	Numerical	Non *	Ranking and selection	Non

- **Non\***: Numerical simulation or system analysis where the author does not mention any software involved in the process.
- **Non\*\***: Manual numerical optimization.
- **Not Mentioned\***: The authors only mention they used “a commercial 3D EM simulator”.

## APPENDIX B. ABBREVIATIONS

TABLE 7. Abbreviations.

Abbreviation	Full name
ANN	Artificial Neural Networks
BCD	Block Coordinate Descent
BFD	Best-Fit Decreasing
CMA-ES	Covariance matrix adaptation evolution strategy
CMOEA-MS	Constrained multi-objective evolutionary algorithm with multiple stages
COBYLA	Constrained Optimization By Linear Approximation
DEA	Differential Evolution Algorithm
E-S	EXTREMUM-SEEKING
FFD	First-Fit Decreasing
FOA	Fruit fly Optimization Algorithm
GDO	Goal-Driven Optimization
GOBI	Gradient-Based Optimization using Back-propagation to Input
IPOPT	Interior-Point Optimizer
LWM	Linear Weighted Method
MOR	Model Order Reduction
MPC	Model Predictive Control
NNIA	Non-Dominated Neighborhood Immune Algorithm
NSGA-II	Non-dominated Sorting Genetic Algorithm
OPF	Optimal Power Flow
PSO	Particle swarm optimization
RBFOpt	Radial Basis Function Optimization
RSM	Response Surface Methodology
SNOPT	Sparse Non-linear Optimizer
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution
TS	Tabu Search algorithm
VVO	Volt-var optimization

## APPENDIX C. SOFTWARE COMPILATION

TABLE 8. Software description (part 1 of 2).

Software	Definition of the software	Ref
Abaqus	Finite Element Analysis Software	[173]
Adams	Multibody dynamics simulation	[13]
ADS	Advanced Design System, Electronic circuit simulation	[181]
AMESim	Thermo-hydraulic simulation software	[165]
AMPERES	Multi-Physics simulation for Electrochemical and Renewable Energy Storage	[164]
ANSYS	Engineering simulation software package	[82]
AnnAGNPS	Simulation software to evaluate management decisions on watersheds	[135]
APECS	Advanced Process Engineering Co-Simulator	[130]
API REST	Set of protocols for building and integrating applications software	[166]
AQWA	Hydrodynamic simulation for offshore and marine structures (Part of ANSYS)	[83]
ARENA	Discrete event simulation software	[18]
Aspen Plus	Process Simulation Software for the Chemical Industry	[90]
CarSim	Simulation Software of vehicle behavior	[35]
COMSOL	Software for Multiphysics Simulation	[32]
CST	3D electromagnetic software for design, analysis and optimization	[174]
C++	High-level and general-purpose programming language	[49]
DAKOTA	Software for optimization using advanced parametric analyses	[183]
DNP3	Client-Master Simulator	[52]
Docker	OS-level virtualization to deliver software in packages called containers.	[93]
DSHplus	Dynamic simulation of fluid power systems	[78]
Dymola	Dynamic complex systems simulation	[175]
Eclipse	Integrated development environment (IDE) used in computer programming	[8]
EDEM	Discrete Element Method (DEM) Software	[77]
EMTP-RV	Electromagnetic Transients Program	[51]
EnergyPlus	Building simulation software	[138]
ePHASORSIM	Real-time transient stability simulation	[98]
EXata CPS	Network communications emulator	[182]
FEKO	Electromagnetic Simulation	[79]
Flomaster	Thermo-fluid simulation	[162]
FLUENT	Fluid simulation software (Part of ANSYS)	[84]
FMI	Functional Mock-up Interface standard for model exchange and co-simulation	[146]
Forge	Hot and cold forging simulation	[185]
FrontFlow/Blue	FEM solver for flow prediction in complicated geometries and resulting sound	[48]
Gazebo	2D/3D robotics simulator	[152]
GenOpt	Optimization software that connects to external simulation programs	[113]
Gleebler	Thermal-Mechanical Simulators	[94]
Grasshopper	Software for Parametric Modeling of Structural Engineering	[100]
GridLAB-D	Power distribution system simulator	[137]
GT-Power	Engine performance simulation	[53]
HFSS	High-Frequency Structure Simulation Software (Part of ANSYS)	[85]
HSPICE	Industry's 'gold standard' for accurate circuit simulation	[102]
iFogSim	Resource Management for IoT, Edge, and Fog Computing Simulation	[31]
IPS	Integrated Plasma Simulator Framework	[147]
Isight	Software for automating and optimizing simulation process flows.	[176]
Java	High-level, class-based, object-oriented programming language	[143]
LabView	Dynamic systems simulation	[34]
Lingo	Optimization Software for Linear, Non-linear, and Integer Programming	[3]

UNIVERSITÉ DE LORRAINE, CNRS, LORIA

*Email address:* [diego-alejandro.vega-vega@loria.fr](mailto:diego-alejandro.vega-vega@loria.fr)

UNIVERSITÉ DE LORRAINE, CNRS, LORIA

*Email address:* [vincent.chevrier@loria.fr](mailto:vincent.chevrier@loria.fr)

TABLE 9. Software description (part 2 of 2).

Software	Definition of the software	Ref
LS-DYNA	Multiphysics simulation software package (Part of ANSYS)	[118]
LTspice	Simulation Program with Integrated Circuit Emphasis	[81]
MATLAB	Numeric multi-paradigm programming language	[97]
MicroGP	A multi-purpose extensible self-adaptive evolutionary optimization tool	[169]
Maxwell	Low-Frequency EM-Field Simulation (Part of ANSYS)	[86]
ModeFRONTIER	Simulation software for process automation and optimization	[167]
ModelCenter	Multi-model optimization software (Part of ANSYS)	[87]
MODFLOW	Geological modular finite-difference simulation for groundwater flow	[140]
MoM	3D planar electromagnetic (EM) simulator (part of ADS)	[151]
MONSTER	Microkernel for scabrous terrain exploring robots	[168]
MotorCAD	Electric Machine design simulation	[88]
MOVES	MOtor Vehicle Emission Simulator (MOVES)	[4]
MT3DMS	Transport Model for Simulation of Contaminants in Groundwater Systems	[14]
NS-3	Discrete-event network simulator for Internet systems	[133]
OPAL-RT	Open Real-Time Digital Simulator	[99]
OpenDSS	Electric power distribution system simulator (DSS)	[95]
OpenModelica	Mathematical equations based modeling and simulation environment	[142]
Opnet	Network simulator	[148]
Opossum	Optimization Solver with Surrogate Models (Part of Grasshopper)	[45]
OptiY	Multidisciplinary design environment with modern optimization strategies	[55]
OSCAR	Pantograph/catenary interaction simulation software	[161]
Particle Flow Code	Discrete-Element Method simulation for synthetic materials	[33]
Pro ENGINEER	Simulation software for mechanical engineering design	[149]
PSIM	Electronic circuit simulation software package	[80]
PyBoolNet	Python package for the simulation of Boolean networks	[108]
Python	High-level, general-purpose programming language.	[50]
RecurDyn	Interdisciplinary computer-aided engineering software	[96]
RSCAD	Real-time simulation software	[10]
RTDS	Electromagnetic transient (EMT) simulations	[11]
SDM	Java program for System Dynamics models (.mdl)	[160]
Sigrity PowerDC	Electrical and thermal simulation software	[91]
Simpack	Multibody system Simulation	[177]
Simplorer	Multi-physics circuit simulator (Part of ANSYS)	[89]
SimulationX	Mechanical, hydraulic, pneumatic, electrical, and combined systems simulation	[60]
Simulink	Numerical simulation software	[103]
Solcore	Python-based library for modeling solar cells and semiconductor materials	[9]
Solidworks	Mechanical modeling en 2D y 3D	[178]
Spectre	Circuit simulator from Cadence	[92]
SPEED	Electric Machine Design Simulator	[43]
SWAT	Quasi-physically-based water quality simulator	[134]
SystemC	System design and verification language that spans hardware and software	[105]
TRNFLOW	Ventilation dynamics simulator	[7]
TRNSYS	Transient systems simulation	[186]
UCEF	Universal Cyber-Physical Systems Environment for Federation Simulation	[139]
UG	Advanced high-end Design, engineering, and manufacturing simulation software	[61]
UM	Dynamics of planar and spatial mechanical systems simulation	[136]
VISSIM	Microscopic traffic simulation software	[56]
VISUM	Macroscopic traffic simulation software	[57]
VSI Fortran	Implementation of the Fortran programming language software	[104]
Xilinx	Simulink add-on that enables the development of FPGA designs	[76]