# Decreasing Risk in the Design of Large Coupled Systems via Co-Simulation-Based Optimization and Adaptive Stress Testing

Severin Sadjina[1] Lars T. Kyllingstad[2] Stian Skjong[1]

[1]SINTEF Nordvest, Ålesund, Norway {severin.sadjina,stian.skjong}@sintef.no
[2]SINTEF Ocean, Trondheim, Norway, lars.kyllingstad@sintef.no

## Abstract

Optimization and stress testing are key aspects of the design and verification process for large, high-risk systems. Optimization is about improving the capabilities and performance of a system; stress testing is about uncovering its weaknesses and faults. Both require a quantitative representation of the system's behavior, and for complex, multi-physical systems, co-simulation can be a very powerful method to create such a representation. However, co-simulation frequently involves the use of black-box subsystem models, which poses challenges to traditional optimization and stress testing methods. Here, we review the state of the art in co-simulation-based optimization and stress testing, focusing especially on *adaptive stress testing* in the latter case, and discuss open research questions and promising research directions. In particular, we make the case that a co-simulation is not an entirely black box even when some or all of its subsystems are; it may be possible to exploit the visible system structure, coupling variable values, and partial subsystem information. We use examples from the maritime industry to motivate and illustrate the discussion, centering on the highly contemporary design case of an autonomous ferry.
*Keywords: co-simulation, optimization, stress testing*

## 1 Background

Like most other industries, the maritime sector finds itself in the middle of a "twin transition": a green transition to a low-carbon future, intertwined with a digital transition towards data-driven decision making and autonomous systems. And like any other, the maritime industry has its own, particular set of challenges in that regard.

The main obstacle to decarbonization lies in finding alternatives to the fossil fuel oils that power virtually all of today's ships. At this point in time, there are few readily-available zero- or low-carbon options for ships that spend many days or weeks away from shore, such as long-distance transport ships and large fishing vessels. In the meantime, fuel-saving and transitional solutions like hybrid and dual-fuel power systems significantly increase the complexity of a ship in terms of design, construction, operation, and maintenance.

When it comes to maritime digitization, the frontier is moving towards remotely operated, semi-autonomous, and eventually autonomous shipping. While automating the navigation of a ship through well-regulated seaways is arguably an easier problem than autopiloting a car through chaotic city streets, the consequences of any one accident are likely to be orders of magnitude larger – possibly in terms of human life, depending on the ship type, but also in terms of environmental and financial impacts. Reducing risk and increasing safety are therefore prime concerns.

Hence, the maritime industry is, now more than ever, in need of new methods for design, verification, and assurance of highly complex, energy efficient, and *safe* systems. Such methods must be able to account for the fact that the design, construction, and commissioning of a large ship is a significant undertaking which involves a great number of stakeholders, a wide array of engineering disciplines, and tight regulatory and budgetary constraints.

This is the context within which we have launched the research project *OptiStress: System optimization and stress testing in co-simulations*. Our basic premise is that co-simulation,[1] as a modeling and simulation paradigm, fits the bill in terms of its suitability for multilateral virtual prototyping of multi-physical and cyber-physical systems. This is in large part because it enables one to build simulations of complex systems from loosely coupled black-box subsystems. However, this becomes a *disadvantage* when it comes to design optimization and whole-system assurance, because some of the assumptions underlying existing numerical methods no longer hold.

In *OptiStress*, and in this paper, we will look at two key design and verification processes and explore how co-simulation may be embedded within them in a way that leverages its strengths and mitigates its drawbacks. The first is *system optimization*, in particular the use of numerical optimization algorithms to choose the system configuration and parameters that best fulfill the system design goals. The second is *stress testing*, which is concerned with pushing a system to the limits of its safe operating space to see what breaks, how, and why. Here, we'll look specifically at *adaptive stress testing* (AST), where reinforcement learning is applied to find the most likely paths to a failure event (Lee, Mengshoel, et al. 2020).

[1]The term "co-simulation" is sometimes used loosely in the academic literature to mean different things. Here, we refer specifically to *continuous-time* co-simulation, as defined by Kübler and Schiehlen (2000) and Gomes et al. (2019).

The goal of this paper is to provide a concise summary of prior work and outline the main research needs. To this end, separate discussions on optimization and AST are given in the context of co-simulation in sections 2 and 3, respectively. Section 4 concludes by commenting on how both can be combined in the same design workflow or even the same algorithmic loop.

We will use an autonomous battery-electric car ferry as an illustrative example throughout. Challenging situations such as waves and wind from the side or crossing traffic in conjunction with strict safety and energy-efficiency requirements, make its design, verification, and assurance daunting tasks. With the present paper, we hope to be able to convince the reader that co-simulation-based optimization and adaptive stress testing are fitting tools for decreasing the risk in the design of such large coupled systems.

# 2 Optimization

There are four main challenges when it comes to optimization of large coupled systems: the number of design parameters to consider is large; there are multiple, possibly contradictory, objectives for the system performance; the objective functions are globally non-convex so gradient-based optimization may yield only local optima; and substantial computational cost can severely restrict the exploration of the solution space. Significant progress has been made on these issues in the field of *multidisciplinary design optimization* (MDO) (Martins and Lambe 2013; Wang, Fan, and Qiang 2023), which has so far found most of its applications in the aerospace and automotive industries, and more recently in the maritime domain (Ojo, Collu, and Coraddu 2022; Serani, Scholcz, and Vanzi 2024).

## 2.1 Background

Co-simulation-based optimization has recently awoken interest across various domains and applications as a natural extension of the utilization of co-simulation's strengths: the decoupling of the component modeling and system simulation processes, the ability to hide sensitive information within black-box models, the use of software tools and solvers most suited for the individual subsystems, and the ability to distribute the simulation burden across several parallel processes. Additional challenges emerge when the objective function is computed on the basis of co-simulation, however:

- *Complex or unpredictable behavior.* Knowledge of how the physics inside a subsimulator is implemented is often spotty, or may lack entirely. More so, models need not be based on physics representations at all. A wide variety of model types may be used in co-simulations, including discrete event models, stochastic models, agent-based models, or even look-up tables. Hence, assumptions about the functional form of objective functions, such as linearity, convexity, or smoothness, can often not be made.
- *Lack of information.* Access to gradients is typically sparse, or lacking completely for most models. The same goes for access to system states, which are usually not accessible at all with co-simulation.
- *Lack of features.* Important features such as the ability to save and restore the simulation state are often not available.
- *Coupling errors.* Errors stemming solely from the co-simulation coupling lead to significant uncertainty with respect to the validity of the system response. In turn, this can cause issues with traditional algorithms due to numerical instability or invalid model outputs.
- *System reticulation.* How the system is reticulated (split into subsystems) for the purpose of co-simulation and how the coupling between these subsystems is realized can have a significant influence on accuracy and stability (Sadjina, Kyllingstad, Skjong, et al. 2017; Sadjina, Kyllingstad, Rindarøy, et al. 2019). This is especially worth keeping in mind if system reticulation and model interfaces are themselves subject to optimization (discrete component optimization).

In light of this, it is natural to turn towards algorithms that fall under the broad heading of *black-box optimization methods*.

**Black-box optimization**

Perhaps the simplest, and thus highly popular, method is *ranking and selection*, whereby a small number of preselected scenarios are compared manually via an objective function. It is used, for example, by Rüdenauer, Han, and Geimer (2012) to optimize the design of a tractor's front axle in a collaborative co-simulation setting, and by Wirth et al. (2017) to find (Pareto front) optimal geometry designs of an airfoil in a fluid-structure co-simulation coupling.

Among algorithmic methods, *meta-heuristic algorithms*, especially genetic algorithms and swarm intelligence optimization methods, are popular choices for co-simulation optimization (Vega and Chevrier 2024). The *constrained multi-objective evolutionary algorithm with multiple stages* (CMOEA-MS) is used by Tan et al. (2022) to optimize the design and the thermal–hydraulic performance of nuclear plate-type fuel. Another variant, *covariance matrix adaptation evolution strategy* (CMA-ES), is used by Ahmed, Oekermann, and Kirchner (2014) to minimize forces acting on a steering mechanism. An evolutionary algorithm is also used by Arslan, Suveren, and Moghaddam (2023) to optimize the design of a co-simulated cart-pole test system. Another example is the *non-dominated sorting genetic algorithm II* (NSGA-II) which is used by Hou et al. (2023) to optimize a valve of a ship HVAC system, or the *multi-objective genetic algorithm* (MOGA) used by Anguek and Bounab (2022) to


DOI 10.3384/ecp218785

optimize the design of a gun station bracket with respect to minimal mass and frequency response under stress constraints. The *differential evolution algorithm* (DEA) is used by Anderson et al. (2020) to find optimal dimensions and bandgaps of semiconductor layers to maximize the efficiency of a solar cell. *Swarm intelligence optimization* comprises nature-inspired meta-heuristic algorithms. One popular candidate is *particle swarm optimization* (PSO), an *a posteriori* algorithm which generates a set of alternative solutions (Pareto optimal set) and prunes them with multi-criteria methods to present to the decision maker for a final choice. It is used by Zadeh, Sayadi, and Kosari (2019) to introduce a multi-objective collaborative optimization framework based on surrogate models, albeit not in conjunction with co-simulation.

#### Surrogate models

It is worth noting that the term *surrogate model* can have two meanings in the context of simulation-based optimization. In the above-mentioned paper by Zadeh, Sayadi, and Kosari (2019), it refers to an approximation of the objective function(s) which is being maintained and updated by the optimization algorithm based on selective evaluation of the actual objective function. The purpose is generally to reduce the number of computationally expensive direct evaluations needed. Other methods in this vein include the *radial basis function optimization* (RBFOpt), which is used by Waibel, Evins, and Carmeliet (2019) to sample an unknown black-box cost function for the study of the coupling between building energy demand and supply, and *constrained optimization by linear approximation* (COBYLA), used by Yamanee-Nolin et al. (2020) to optimize an evaporator process with respect to minimal product stream mass flow oscillations.

The other meaning of "surrogate model" refers to models used *within* the simulation(s) that form the basis for the objective function(s). Here, the point is to replace high-fidelity models with approximated models that are faster to calculate, to replace black-box models with white-box approximations that provide more information (e.g. derivatives), or both. Examples for co-simulation optimization include the use of weighted sums of several individual surrogate models based on either radial basis functions, response surface models, or Kriging surrogate models (Hou et al. 2023), or using feed-forward artificial neural networks to approximate objective functions and calculate gradients (Tuli et al. 2022).

#### Looking inside the box

Despite the challenges that co-simulation brings to the optimization table, things are not necessarily quite as bad as they may first appear: A co-simulation itself is not entirely a black box, even when its individual subsystems are. We do have access to the connection structure and the signals exchanged between the models, and this can be used to estimate errors and speed up the simulations (Kyllingstad, Sadjina, and Skjong 2024). Moreover, at least some of the subsystems may just as well *not* be black-box models. Instead, they may reveal some of their internal workings to us, granting access to gradients and, though rarer, access to system states. For example, if both the decision variables and the objective function are associated with such model types, it may still be possible to use gradient-based methods.

Gradient-based optimization generally requires derivatives and may only yield local optima because of globally non-convex objective functions. Used in conjunction with co-simulation, examples include *back-propagation of gradients with respect to input* (GOBI) which is proposed by Tuli et al. (2022) to find optimal scheduling decisions for fog computing, or *block coordinate descent* (BCD) used by Sadnan et al. (2021) to solve the decentralized optimal power flow problem in a distribution grid.

#### Combining optimization and co-simulation

Most examples of co-simulation optimization use an optimization routine *on top* of the co-simulation loop. Ranking and selection currently seems to be the most common such strategy, solely making up almost 45% of all methods investigated by Vega and Chevrier (2024). It consists of simply running a batch of pre-selected co-simulation scenarios, collecting and analyzing the results, adjusting the design, and repeating until satisfied. Ranking and selection can also be supplemented with post-processing techniques such as Pareto front (Wirth et al. 2017), *goal-driven optimization* (GDO) (Zhaoju et al. 2019), or by using a weighted sum of multiple objective functions (He et al. 2021). Verification by co-simulation is a slightly different take used by Assadi et al. (2023) to gauge the thermal-management performance of a high-density electric-vehicle fast charger based on the topology of a heatsink that was first optimized in a (monolithic) thermo-fluid simulation.

A different approach is *optimization-in-the-loop* whereby optimization happens inside of a co-simulation. One such example is *control optimization* which aims to move the system along a desired path as it evolves dynamically. The objective function then depends on a set of actions taken at discrete time instances. This could, for example, be relevant for dynamic positioning or routing tasks. Control optimization is used by Bharati, Chakraborty, and Darrah (2021) to find the optimal control of consumers and generators in a hardware-in-the-loop co-simulation of the interaction of a real-World microgrid facility with a real-time model of a power grid. It is also used by Tuli et al. (2022) for scheduling optimization, or by Sadnan et al. (2021) to minimize communications between agents controlling subgrids in a distribution power system for optimal decentralized power flow.

## 2.2 Research needs

Some black-box optimization methods seem to remain relatively unexplored in the context of co-simulation still. A prominent example is the class of methods that go under the heading of *Bayesian optimization* (Garnett 2023), which has seen a resurgence in recent years due to its usefulness in machine learning, in particular for tuning neural networks. In Bayesian optimization, the objective function is treated as a random variable to be inferred in light of prior expectations (i.e., assumptions) and data collected (i.e., function evaluations). In our opinion, this seems like a promising approach when co-simulation is involved, one that should be investigated further.

A fact that does not seem to be discussed much in the current literature on co-simulation-based optimization is that co-simulation coupling gives rise to errors that are not present in a monolithic simulation of the same system. As mentioned, this exacerbates issues with simulation accuracy and stability, both of which are difficult to estimate or control for co-simulations (Kyllingstad, Sadjina, and Skjong 2024). A systematic way needs to be found to make sure that the optimization is robust enough to deal with this situation and to avoid over-optimization. In addition, methods for co-simulation error control can help speed up simulations and thus prove useful in optimization contexts.

Lastly, exactly how optimization and co-simulation can or should be used together is a matter that appears to be far from settled. More generally, a systematic analysis identifying which optimization methods work best for system optimization based on co-simulation, under which conditions, and for which cases seems to be lacking. This is also reflected in a large variety of strategies to be found in the literature, see Section 2.1.

### 2.3 Example: Optimizing the design of an autonomous battery-electric ferry

A fully battery-electric ferry needs to be fitted with a sufficiently large battery capacity and be as energy-efficient as possible to handle the vast range of environmental conditions it will encounter during operation. Maneuvering, acceleration, transit, and deceleration must all use as little energy as possible, while different environmental and loading conditions typically require different operational modes for optimality. The ferry design is also constrained by safety considerations. For example, it may have to handle (temporarily) reduced or completely unavailable charging capacity and stay maneuverable even under extreme weather conditions. Lastly, it has to be designed in a way that minimizes seasickness-inducing motions.

When setting up a co-simulation of such a system, a typical approach is to split it into subsimulators representing

- the hull (including effects of waves and currents),
- the power system,
- individual propulsors (e.g. propellers),
- individual pieces of heavy machinery (e.g. car ramp hydraulics),
- individual sensors (e.g. GPS and radar), and
- individual control systems (e.g. autopilot and power management system).

Depending on the vendor constellation involved in its construction, it may also make sense to split the power system further into subsystems representing battery packs, frequency converters, switchboards, back-up generators, and so on.

This system is characterized by strong interdependencies, foremost of which is that all on-board energy consumers are powered by the same, strictly limited, energy source. Multiple control systems are at play, controlling individual components, groups of components, or the entire ship, forming a hierarchy with both upwards and downwards dependencies. Hence, even though the system is loosely coupled and quite naturally separable for purposes of co-simulation, it is very tightly coupled in a system optimization perspective.

## 3 Adaptive Stress Testing

Traditionally, safety has been based on strategies such as introducing redundancy for critical components or by implementing safety functions that are triggered by a controller to prevent unsafe system states (Meulen and Myhrvold 2022). Such strategies are insufficient when dealing with complex and autonomous systems, however, where safety – or lack thereof – emerges as a consequence of the interactions of a large number of interconnected and interdependent controllers (Leveson 2012). This makes simulation-based validation of system safety and compliance a great tool for identifying the effects of these interactions at the system level.

Several approaches have been proposed for trying to find failures[2] in systems in complex interactions with their environment. They can broadly be divided into two categories (Lee, Mengshoel, et al. 2020): *Formal verification* attempts to use mathematical proofs and is able to identify failures as well as absence of failures. Of course, it is often ill-suited for complex engineering systems with stochastic environments. *Simulation-based sampling*, on the other hand, relies on the (manual or automatic) creation of environment instances and initial conditions, and performs searches over the parameter space to find failures in the simulated system–environment interactions. The main challenge is to handle the overwhelming set of possible parameters to test, not least figuring out which are the relevant ones. This calls for methods for automatically running, evaluating, and re-running simulations in a sequence which is *intelligently* guided towards the violation of desired system properties. One such method is *adaptive stress testing* (AST) (Lee, Kochenderfer, et al. 2015) which we believe is a promising candidate for use with co-simulation.

[2]Here, we understand a failure as a violation of a safety property (Corso, Moss, et al. 2021).


DOI
10.3384/ecp218785

### 3.1 Background

The principal idea behind AST is to try to identify the most likely failures of a complex system in an environment through actively guiding failure path sampling. This is especially useful when a formal mathematical falsification or the exhaustive search over the space of initial conditions, parameters, and over a significant length of time are not feasible. This will typically be the case for complex systems in stochastic environments for which the governing equations are not known. AST is a powerful method with applications in airspace safety (Lee, Kochenderfer, et al. 2015; Durling et al. 2021; Guo, Brittain, and Wei 2023), autonomous road vehicle safety and control (Koren, Alsaif, et al. 2018; Corso, Du, et al. 2019; Koren and Kochenderfer 2019; Corso, Lee, and Kochenderfer 2020; Corso and Kochenderfer 2020), software testing (Corradini et al. 2022; Durling et al. 2021; Hellhake et al. 2022), validation of AI systems (Julian, Lee, and Kochenderfer 2020), and collision avoidance of autonomous ships (Torben et al. 2023; Hjelmeland et al. 2022).

The classic formulation of AST (Lee, Kochenderfer, et al. 2015; Koren, Alsaif, et al. 2018) models the problem as a sequential Markov decision process and uses *deep reinforcement learning* with tree search to find action paths with a high failure likelihood. This approach either requires full access to the system states or, when unavailable, full control over the generation of random numbers in the system and the environment (Lee, Mengshoel, et al. 2020). The former case is rendered irrelevant by most industrial co-simulation use cases, while the latter is at least challenging to implement in practice.

### 3.2 Research needs

Unlike for optimization, we have been able to find little to no prior work on combining stress testing with co-simulation. Again, the black-box nature of co-simulations poses challenges, in this case when it comes to the interpretability of the results and the inference of causal chains, as well as issues regarding numerical accuracy and stability. At the same time, stress testing based on reinforcement learning (AST) presents itself as a good fit for co-simulation because it relies only on subsimulator inputs and outputs and thus goes together well with black-box models.

Depending on the specific restrictions and requirements of the case at hand, other approaches and formulations could prove useful and should be further investigated for use with co-simulation:

- If the failure paths need to be interpretable, *signal temporal logic* can be used together with *genetic programming* (Corso and Kochenderfer 2020) or *Gaussian processes* (Torben et al. 2023) to optimize action paths which lead to system failure. This approach also allows to encode domain-relevant requirements and compliance (such as minimal safety distances).
- The *backwards algorithm* (Koren, Nassar, and Kochenderfer 2021) can significantly speed up failure discovery with AST by first running it on a low-fidelity simulator before improving on the results using a high-fidelity simulator. Fidelity differences can induce spurious errors and disagreement in the discovered failures, though the algorithm is in principle able to learn to overcome these issues.
- The use of domain-specific heuristics for the reward function can improve AST performance, for example, by encoding improper system behavior or the inclusion of a dissimilarity measure between paths to try to avoid finding the same failure types repeatedly (Corso, Du, et al. 2019).
- If heuristic rewards are not available, AST performance can suffer from impeded learning due to a lack of reward information before a failure is found. The *go explore* (GE) algorithm can be used to improve AST performance under such conditions (Koren and Kochenderfer 2020).
- The use of a *recurrent neural network* with *long short-term memory* layers can yield a more robust and efficient detection of failures across the entire space of initial conditions (Koren and Kochenderfer 2019) by being able to learn relationships between different paths.
- AST also allows for differential stress testing (Lee, Mengshoel, et al. 2020) to compare failure paths against a baseline. This is useful, for example, when changes to an existing design need to be validated.

Other interesting approaches include the direct estimation of the distribution of failures (Corso, Lee, and Kochenderfer 2020), trying to find the boundary between compliant and non-compliant behavior based on parametric scenarios (Petrov et al. 2022), or extending AST to complex failures involving several autonomous agents and improper environment conditions together (Guo, Brittain, and Wei 2023).

### 3.3 Example: Stress testing the design of an autonomous battery-electric ferry

Our prior discussion of a battery-electric autonomous ferry in section 2.3 came short in terms of making sure that the design solutions are not only optimal but also *safe*. The ferry will have to handle all kinds of potentially dangerous situations, such as: vital sensor signals dropping out, becoming unreliable, or even actively being manipulated; collisions with crossing traffic; propulsors becoming unavailable; battery errors and other power electronics errors; or communications issues with a remote operation center. Worse, it will have to handle all sorts of *combinations* of such situations because the most severe system failures in complex systems stem from the interdependencies between subsystems.

A typical AST approach here would be to have a reinforcement learning agent define a policy that determines how a simulated scenario unfolds, for example how a crossing ship behaves or how the weather changes. The agent would then learn from the scenario outcomes, in particular how close the system came to failure, and in what sense. A proper evaluation of the latter could quite naturally depend on information which would normally be hidden in a co-simulation. One example might be the onboard control systems' internal representation of the state of the system and its environment; another might be the internal causal structure of a complex subsystem (e.g. the power system).

# 4 Outlook

Optimization and stress testing are not completely separable aspects of a complex system design workflow. Safety is an important design objective, and therefore something to be optimized for. Conversely, optimization of other factors could easily affect safety. Keeping to the maritime domain, one example would be a ship power plant which is tightly optimized for energy efficiency in ordinary operating conditions, but which does not have sufficient reserve power and therefore causes a critical black-out in extreme weather conditions. Having found good methods to implement optimization and AST with co-simulation-in-the-loop, the next obvious step is therefore to integrate the two processes in one design loop.

At least two strategies seem to be worthy of further investigation here:

- *nested*: including the results of stress testing when computing the objective function(s) in an optimization loop
- *sequential*: optimize the system, perform stress testing, evaluate, adjust optimization constraints/goals, repeat

We suspect that the first would, at least in principle, be capable of yielding the best results, but be very computationally expensive. The second might be more tractable in terms of computation time, but determining how to constrain or target the optimization procedure to reduce the likelihood of identified failure paths seems a highly nontrivial task, especially when dealing with black-box systems. Lastly, some synergies could potentially be exploited when it comes to surrogate models (optimization) or the use of low-fidelity simulators (AST with the backwards algorithm) to speed up implementations of co-simulation optimization and stress testing. But for now, this all remains conjecture and a highly interesting direction for future research.

This being a contribution to the *International Modelica & FMI Conference*, we would be amiss if we did not comment on FMI's role. Luckily, the standard already provides the complete feature set needed for co-simulation optimization and stress testing. The abilities to read and set derivatives and to store and restore model states, for example, are both already part of the FMI specification (Blochwitz et al. 2011). Therefore, it is up to the model developers to make use of the already implemented features to allow for their models to be used in optimization and stress testing settings.

# Acknowledgements

This work was supported by the Research Council of Norway under the project *System optimisation and stress testing in co-simulations (OptiStress)* (grant no. 344238). We are also thankful for the supplementary funding from DNV, HAV Design, Norwegian Electric Systems, Corvus Energy, and HD Hyundai Europe Research & Development Center.

# References

Ahmed, Mohammed, Christian Oekermann, and Frank Kirchner (2014-01). "Cosimulation environment for mechanical design optimization with evolutionary algorithms". In: *2014 World Symposium on Computer Applications & Research (WSCAR)*. Sousse, Tunisia: IEEE, pp. 1–6. ISBN: 978-1-4799-2806-4. DOI: 10.1109/WSCAR.2014.6916781.

Anderson, Tom H. et al. (2020-04). "Coupled optoelectronic simulation and optimization of thin-film photovoltaic solar cells". In: *Journal of Computational Physics* 407, p. 109242. ISSN: 00219991. DOI: 10.1016/j.jcp.2020.109242.

Anguek, Oussama and Belkacem Bounab (2022-12). "Multi-objective design optimization of a Turret's U-bracket mounted on moving platform". In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 236.24, pp. 11371–11388. ISSN: 0954-4062, 2041-2983. DOI: 10.1177/09544062221115106.

Arslan, Erdem, Memduh Suveren, and Seyed Tofigh Hosseini Moghaddam (2023-11). "ROS Gazebo and MATLAB/Simulink Co-simulation for Cart-Pole System: A Framework for Design Optimization". In: *2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS)*. Istanbul, Turkiye: IEEE, pp. 1–7. ISBN: 979-8-3503-8306-5. DOI: 10.1109/ISAS60782.2023.10391735.

Assadi, Seyed Amir et al. (2023-03). "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". In: *2023 IEEE Applied Power Electronics Conference and Exposition (APEC)*. Orlando, FL, USA: IEEE, pp. 1808–1815. ISBN: 978-1-6654-7539-6. DOI: 10 . 1109 / APEC43580 . 2023 . 10131627.

Bharati, Guna R., Sudipta Chakraborty, and Julien Darrah (2021-02). "A Co-simulation-based Hardware-in-the-loop Architecture for Validating Power Systems Optimization with Large-scale Grid Models". In: *2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. Washington, DC, USA: IEEE, pp. 1–5. ISBN: 978-1-7281-8897-3. DOI: 10.1109/ISGT49243.2021.9372282.

Blochwitz, T. et al. (2011-06). "The Functional Mockup Interface for Tool independent Exchange of Simulation Models". In: *Proceedings of the 8th International Modelica Conference*. Linköping Electronic Conference Proceedings 63, pp. 105–114. DOI: 10.3384/ecp11063105.

Corradini, Davide et al. (2022-08). "Automated black-box testing of nominal and error scenarios in RESTful APIs". In: *Software Testing, Verification and Reliability* 32.5, e1808. ISSN: 0960-0833, 1099-1689. DOI: 10.1002/stvr.1808.

Corso, Anthony, Peter Du, et al. (2019-10). "Adaptive Stress Testing with Reward Augmentation for Autonomous Vehicle Validatio". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. Auckland, New Zealand: IEEE, pp. 163–168. ISBN: 978-1-5386-7024-8. DOI: 10.1109/ITSC.2019.8917242.

Corso, Anthony and Mykel J. Kochenderfer (2020-09). "Interpretable Safety Validation for Autonomous Vehicles". In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. Rhodes, Greece: IEEE, pp. 1–6. ISBN: 978-1-7281-4149-7. DOI: 10.1109/ITSC45102.2020.9294490.

Corso, Anthony, Ritchie Lee, and Mykel J. Kochenderfer (2020-09). "Scalable Autonomous Vehicle Safety Validation through Dynamic Programming and Scene Decomposition". In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. Rhodes, Greece: IEEE, pp. 1–6. ISBN: 978-1-7281-4149-7. DOI: 10.1109/ITSC45102.2020.9294636.

Corso, Anthony, Robert Moss, et al. (2021-10). "A Survey of Algorithms for Black-Box Safety Validation of Cyber-Physical Systems". In: *Journal of Artificial Intelligence Research* 72. ISSN: 1076-9757. DOI: 10.1613/jair.1.12716.

Durling, Michael et al. (2021-10). "Certification Considerations for Adaptive Stress Testing of Airborne Software". In: *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. San Antonio, TX, USA: IEEE, pp. 1–9. ISBN: 978-1-6654-3420-1. DOI: 10.1109/DASC52595.2021.9594395.

Garnett, Roman (2023). *Bayesian optimization*. Cambridge: Cambridge university press. ISBN: 978-1-108-42578-0.

Gomes, Cláudio et al. (2019-05). "Co-Simulation: A Survey". In: *ACM Computing Surveys* 51.3, pp. 1–33. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3179993.

Guo, Wei, Marc Brittain, and Peng Wei (2023-10). "Safety Validation for Deep Reinforcement Learning Based Aircraft Separation Assurance with Adaptive Stress Testing". In: *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*. Barcelona, Spain: IEEE, pp. 1–10. ISBN: 979-8-3503-3357-2. DOI: 10.1109/DASC58513.2023.10311238.

He, Haibin et al. (2021-05). "Structure Optimization of Natural Gas Engine's Intake Pipe Based on One-Dimensional and Three-Dimensional Coupled Simulation". In: *IOP Conference Series: Earth and Environmental Science* 781.4, p. 042029. ISSN: 1755-1307, 1755-1315. DOI: 10.1088/1755-1315/781/4/042029.

Hellhake, Dominik et al. (2022-06). "Towards using coupling measures to guide black-box integration testing in component-based systems". In: *Software Testing, Verification and Reliability* 32.4, e1811. ISSN: 0960-0833, 1099-1689. DOI: 10.1002/stvr.1811.

Hjelmeland, Hanna W. et al. (2022). "Identification of Failure Modes in the Collision Avoidance System of an Autonomous Ferry using Adaptive Stress Testing". In: *IFAC-PapersOnLine* 55.31, pp. 470–477. ISSN: 24058963. DOI: 10.1016/j.ifacol.2022.10.472.

Hou, Jianjun et al. (2023-03). "Co-Simulation Modeling and Multi-Objective Optimization of Dynamic Characteristics of Flow Balancing Valve". In: *Machines* 11.3, p. 337. ISSN: 2075-1702. DOI: 10.3390/machines11030337.

Julian, Kyle D., Ritchie Lee, and Mykel J. Kochenderfer (2020-09). "Validation of Image-Based Neural Network Controllers through Adaptive Stress Testing". In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. Rhodes, Greece: IEEE, pp. 1–7. ISBN: 978-1-7281-4149-7. DOI: 10.1109/ITSC45102.2020.9294549.

Koren, Mark, Saud Alsaif, et al. (2018-06). "Adaptive Stress Testing for Autonomous Vehicles". In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. Changshu: IEEE, pp. 1–7. ISBN: 978-1-5386-4452-2. DOI: 10.1109/IVS.2018.8500400.

Koren, Mark and Mykel J. Kochenderfer (2019-10). "Efficient Autonomy Validation in Simulation with Adaptive Stress Testing". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. Auckland, New Zealand: IEEE, pp. 4178–4183. ISBN: 978-1-5386-7024-8. DOI: 10.1109/ITSC.2019.8917403.

Koren, Mark and Mykel J. Kochenderfer (2020-09). "Adaptive Stress Testing without Domain Heuristics using Go-Explore". In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. Rhodes, Greece: IEEE, pp. 1–6. ISBN: 978-1-7281-4149-7. DOI: 10.1109/ITSC45102.2020.9294729.

Koren, Mark, Ahmed Nassar, and Mykel J. Kochenderfer (2021-09). "Finding Failures in High-Fidelity Simulation using Adaptive Stress Testing and the Backward Algorithm". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Prague, Czech Republic: IEEE, pp. 5944–5949. ISBN: 978-1-6654-1714-3. DOI: 10.1109/IROS51168.2021.9636072.

Kübler, R. and W. Schiehlen (2000-06). "Two Methods of Simulator Coupling". In: *Mathematical and Computer Modelling of Dynamical Systems* 6.2, pp. 93–113. ISSN: 1387-3954. DOI: 10.1076/1387-3954(200006)6:2;1-M;FT093.

Kyllingstad, Lars T., Severin Sadjina, and Stian Skjong (2024). *Error estimation and step size control with minimal subsystem interfaces*. arXiv:2406.17353v1. DOI: 10.48550/ARXIV.2406.17353.

Lee, Ritchie, Mykel J. Kochenderfer, et al. (2015-09). "Adaptive stress testing of airborne collision avoidance systems". In: *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*. Prague, Czech Republic: IEEE, pp. 6C2–1–6C2–13. ISBN: 978-1-4799-8940-9. DOI: 10.1109/DASC.2015.7311450.

Lee, Ritchie, Ole J. Mengshoel, et al. (2020-12). "Adaptive Stress Testing: Finding Likely Failure Events with Reinforcement Learning". In: *Journal of Artificial Intelligence Research* 69, pp. 1165–1201. ISSN: 1076-9757. DOI: 10.1613/jair.1.12190.

Leveson, Nancy G. (2012-01). *Engineering a Safer World: Systems Thinking Applied to Safety*. The MIT Press. ISBN: 978-0-262-29824-7. DOI: 10.7551/mitpress/8179.001.0001.

Martins, Joaquim R. R. A. and Andrew B. Lambe (2013-09). "Multidisciplinary Design Optimization: A Survey of Architectures". In: *AIAA Journal* 51.9, pp. 2049–2075. ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.J051895.

Meulen, Meine van der and Tore Myhrvold (2022). *Demonstrating safety of software-dependent systems*. DNV AS. ISBN: 978-82-515-0324-2.

Ojo, Adebayo, Maurizio Collu, and Andrea Coraddu (2022-12). "Multidisciplinary design analysis and optimization of floating offshore wind turbine substructures: A review". In: *Ocean Engineering* 266, p. 112727. ISSN: 00298018. DOI: 10.1016/j.oceaneng.2022.112727.

Petrov, Aleksandar et al. (2022-10). "HiddenGems: Efficient safety boundary detection with active learning". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Kyoto, Japan: IEEE, pp. 5147–5154. ISBN: 978-1-6654-7927-1. DOI: 10.1109/IROS47612.2022.9982243.

Rüdenauer, Andreas, Song Han, and Marcus Geimer (2012). "Optimization of the development process of mobile machines using a standardized co-simulation". In: *Landtechnik* 67.2, pp. 122–126.

Sadjina, Severin, Lars T. Kyllingstad, Martin Rindarøy, et al. (2019-02). "Distributed Co-simulation of Maritime Systems and Operations". In: *Journal of Offshore Mechanics and Arctic Engineering* 141.1, p. 011302. ISSN: 0892-7219, 1528-896X. DOI: 10.1115/1.4040473.

Sadjina, Severin, Lars T. Kyllingstad, Stian Skjong, et al. (2017-07). "Energy conservation and power bonds in co-simulations: non-iterative adaptive step size control and error estimation". In: *Engineering with Computers* 33.3, pp. 607–620. ISSN: 0177-0667, 1435-5663. DOI: 10.1007/s00366-016-0492-8.

Sadnan, Rabayet et al. (2021-07). "Distributed Optimization for Power Distribution Systems with Cyber-Physical Co-Simulation". In: *2021 IEEE Power & Energy Society General Meeting (PESGM)*. Washington, DC, USA: IEEE, pp. 1–5. ISBN: 978-1-6654-0507-2. DOI: 10.1109/PESGM46819.2021.9638251.

Serani, Andrea, Thomas P. Scholcz, and Valentina Vanzi (2024-12). "A Scoping Review on Simulation-Based Design Optimization in Marine Engineering: Trends, Best Practices, and Gaps". In: *Archives of Computational Methods in Engineering* 31.8, pp. 4709–4737. ISSN: 1134-3060, 1886-1784. DOI: 10.1007/s11831-024-10127-1.

Tan, Jiatong et al. (2022-05). "Multi-Objective Optimization Design of Plate-type Fuel via Co-simulation Method". In: *Annals of Nuclear Energy* 169, p. 108914. ISSN: 03064549. DOI: 10.1016/j.anucene.2021.108914.

Torben, Tobias Rye et al. (2023-04). "Automatic simulation-based testing of autonomous ships using Gaussian processes and temporal logic". In: *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 237.2, pp. 293–313. ISSN: 1748-006X, 1748-0078. DOI: 10.1177/1748006X211069277.

Tuli, Shreshth et al. (2022-01). "COSCO: Container Orchestration Using Co-Simulation and Gradient Based Optimization for Fog Computing Environments". In: *IEEE Transactions on Parallel and Distributed Systems* 33.1, pp. 101–116. ISSN: 1045-9219, 1558-2183, 2161-9883. DOI: 10.1109/TPDS.2021.3087349.

Vega, Diego Alejandro and Vincent Chevrier (2024-08). *Co-simulation-based optimization: state of the art*. hal-04668193, version 1.

Waibel, Christoph, Ralph Evins, and Jan Carmeliet (2019-05). "Co-simulation and optimization of building geometry and multi-energy systems: Interdependencies in energy supply, energy demand and solar potentials". In: *Applied Energy* 242, pp. 1661–1682. ISSN: 03062619. DOI: 10.1016/j.apenergy.2019.03.177.

Wang, Chong, Haoran Fan, and Xin Qiang (2023-10). "A Review of Uncertainty-Based Multidisciplinary Design Optimization Methods Based on Intelligent Strategies". In: *Symmetry* 15.10, p. 1875. ISSN: 2073-8994. DOI: 10.3390/sym15101875.

Wirth, Nadja et al. (2017). "Analysis and Optimization of Flow Around Flexible Wings and Blades Using the Standard Co-simulation Interface MpCCI". In: *Recent Progress in Flow Control for Practical Flows*. Ed. by Piotr Doerffer, George N. Barakos, and Marcin M. Luczak. Cham: Springer International Publishing, pp. 283–321. ISBN: 978-3-319-50568-8. DOI: 10.1007/978-3-319-50568-8_15.

Yamanee-Nolin, Mikael et al. (2020-03). "Trajectory optimization of an oscillating industrial two-stage evaporator utilizing a Python-Aspen Plus Dynamics toolchain". In: *Chemical Engineering Research and Design* 155, pp. 12–17. ISSN: 02638762. DOI: 10.1016/j.cherd.2019.12.015.

Zadeh, Parviz Mohammad, Mohsen Sayadi, and Amirreza Kosari (2019-01). "An efficient metamodel-based multi-objective multidisciplinary design optimization framework". In: *Applied Soft Computing* 74, pp. 760–782. ISSN: 15684946. DOI: 10.1016/j.asoc.2018.09.014.

Zhaoju, Qin et al. (2019-11). "Diesel engine piston thermo-mechanical coupling simulation and multidisciplinary design optimization". In: *Case Studies in Thermal Engineering* 15, p. 100527. ISSN: 2214157X. DOI: 10.1016/j.csite.2019.100527.