Autonomous Vehicular Systems: Architectural Strategies for Adaptive Multi-Objective Configuration

Julian Demicoli*, Nicolai Palm†, Herbert Palm‡, Oliver Kleikemper*, and Sebastian Steinhorst*
*Technical University of Munich, Munich, Germany
†Ludwig Maximilian University Munich, Munich, Germany
‡Munich University of Applied Sciences, Munich, Germany

***Abstract*—The dynamic reconfiguration of vehicular control and management systems to adapt to different scenarios, particularly those with conflicting design goals, remains a challenging task. In this context, we propose a reference architecture and a generic process to integrate advanced gain scheduling with a feedback loop that continually updates configuration lookup tables to generate scenario-related configurations at runtime. For demonstration purposes, our approach is applied to a Hyperloop vehicle's magnetic suspension system to guarantee simultaneously optimized accuracy of control, energy consumption and passenger comfort for a multitude of scenarios. Additionally, the feedback mechanism increases resilience against mechanical failures, marking an advancement in vehicular system adaptability and reliability.**

***Index Terms*—Autonomous Systems, Vehicular Systems, Multi-Objective Optimization, Real-time, Hyperloop**

I. Introduction

Dynamic configuration of vehicular systems within their multitude of relevant scenarios under multi-objective requirements represents a highly challenging system design goal. While multi objective optimization (MOO) is not a new application in vehicular system development per se [1] [2], it is important to note that its implementation (e.g. by evolutionary algorithms [3] or gaussian process based MOO [4] based methods) is typically not real-time capable [5]. However, the dynamic and unpredictable nature of real-world settings often exposes the limitations of traditional models that are not able to autonomously reconfigure during runtime.

In the field of vehicular systems, where the synergy of computational and physical processes is of central importance, the ability to respond to unexpected changes is central not only to maintaining efficiency, but also to ensuring safety. This is where the principle of autonomy is elevated from a mere capability to an essential system characteristic. In this context, *autonomy* outreaches conventional adaptive control. It involves a system's ability to self-manage and optimize various aspects, such as control parameters, software configurations, and hardware states, all without human intervention.

Against this background, we propose a novel reference framework for autonomous vehicular systems enabling real-time adaptive multi-objective-optimal system reconfigurations. This framework incorporates Configuration Scheduling, which is adapted from gain scheduling, a nonlinear control methodology. In contrast to gain scheduling, that adjusts control parameters ('gains') according to an operation graph, Configuration Scheduling dynamically adjusts control parameters based on predefined multi-dimensional scenarios, allowing us to pre-emptively provide a set of optimized configurations for various operating conditions. To obtain the multi-objective optimal configurations for the specific scenarios, we employ an optimization algorithm. This algorithm focuses on minimizing the cost of each configuration, utilizing a digital twin at its foundation. Additionally, we've incorporated a feedback loop into our framework. This loop is specifically designed to adaptively manage unexpected scenarios, like material wear and tear or alterations in interconnected systems. By continually updating the digital twin, our system can reconfigure and adjust strategies based on a holistic understanding of the operational environment. Our Configuration Scheduling approach directly addresses the need for rapid adaptation in dynamic environments, a cornerstone of autonomy in vehicular systems. Similarly, the feedback loop responds to slower, less predictable changes, embodying the autonomous system's ability to self-optimize and ensure long-term reliability. The reference frameworks also contains an architecture, describing the required components to perform Configuration Scheduling which is depicted in Figure 1.

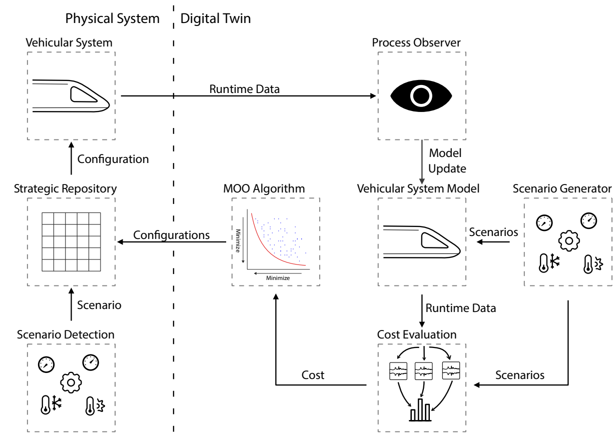

Fig. 1: Overview of reference architecture components.

We further demonstrate our framework's efficacy using the Hyperloop magnetic suspension system. This system demands fast adaptation at high-speed conditions, making Strategy Scheduling an ideal approach. Additionally, our feedback loop is designed to detect and respond to slower changes, such as track aging or mechanical deterioration, ensuring long-term system reliability and safety.

In summary, our contribution is threefold:

- We propose Strategy-scheduling, an approach adapted from nonlinear control theory that allows for Autonomous Pareto-optimal reconfiguration of a vehicular system. (Section IV-A)
- We propose a reference architecture that defines the entities required to operate Configuration Scheduling. (Section IV-B)
- We evaluate and demonstrate the architecture in a real-world scenario by applying it on a Hyperloop magnetic levitation system to show that the system performs better in all our performance metrics various scenarios when compared to the initial linear quadratic regulator (LQR) methodology (Section V).

II. Background

A. MAPE-K

The concept of autonomous computing has been a subject of interest since the introduction of the MAPE-K framework by an IBM study over two decades ago [6]. Despite its age, MAPE-K continues to be a significant concept to achieve self-adaptation [7] . Its core principle involves utilizing a feedback loop comprising four stages: Measure (M), Analyze (A), Plan (P), and Execute (E), underpinned by Knowledge (K), to facilitate system autonomy. While MAPE-K does not provide detailed methodologies for each stage, it offers a foundational framework for developing an autonomous strategy.

B. Multi-Objective Optimization

Multi-objective optimization (MOO) is a mathematical and computational approach used to find the best trade-offs when there are multiple, possibly conflicting objectives or criteria that need to be optimized simultaneously. These scenarios, where no single solution optimally satisfies all objectives, are prevalent in various domains such as power systems [4] and automated manufacturing systems [8]. Mathematically, the goal of MOO is to identify (Pareto-)optimal trade-offs between the various objectives.

Definition (Pareto point)**.** *A* $t \in T \subset \mathbb{R}^n$ *is called* Pareto optimal (or Pareto point) *if there exists no* $s \in T$ *such that* $s_i \leq t_i$ *for all* $i = 1, ..., n$ *and* $s_j < t_j$ *for some* j.

An $x \in D \subset \mathbb{R}^m$ *is* a Pareto point of $f : D \to T$ *if* $f(x) \in \{f(t) \colon t \in D\}$ *is a Pareto point.*

In other words, a point is Pareto optimal if an improvement in one component always results in a deterioration of some other component. Thus, Pareto optimal solutions represent the best possible compromises between the conflicting objectives. MOO employs algorithms such as genetic algorithms [3] and Gaussian process based MOO [4].

Typically, such algorithms iteratively evaluate an underlying (simulation) model. Given that most vehicular simulation models are computationally intensive, this makes real-time applicability of MOO methods a challenging task.

C. Gain Scheduling

As previously highlighted, current multi-objective optimization algorithms face inherent speed limitations due to their reliance on constructing models from simulation data or real-world inputs. This process, which involves sampling and model construction, inherently ties the system's reponsiveness to the execution times of these models. This is particulary challenging in vehicular systems where multiple, often conflicting design goals depend heavily on the system's configuration state. In nonlinear control theory, similar challenges arises when dealing with systems that operate across multiple points. For instance, the startup phase of a motor presents a scenario where friction dynamics are highly nonlinear until a certain threshold is reached. To manage this complexity, control theorists have developed gain-scheduling. This approach involves a look-up table (LUT) that stores the control parameters at various predefined operation points. This is done by sampling the nonlinear operation curve at regular intervals, linearizing the system at each point and optimizing the controller gains for that linearized state. This enables the control algorithm to swiftly adapt as the system transitions between different operating points. Applying this principle to vehicular systems offers a promising avenue to enhance their adaptability and response time in dynamic operational environments.

III. Related Works

The evolution of multi-objective optimization (MOO) in vehicular systems has witnessed a shift from static to dynamic configurations. This section traces this progression, positioning our contributions within this dynamic landscape, especially in autonomous system reconfiguration.

Using MOO algorithms to tune linear controller gains for a single use case has been demonstrated already in 2006 on a highly nonlinear magnetic suspension system [1]. More recent advancements in this domain demonstrate a significant shift towards dynamic optimization. Given the typical limitations of MOO algorithms in real-time applications, various strategies have emerged. Notably, Ionescu et al. (2020) introduced a prioritized multi-objective optimization approach in model predictive control (MPC) for cyber-physical systems [9]. This method strategically prioritizes specific design goals at each iteration, effectively transforming the problem into a single-objective optimization to reduce computational complexity. While this approach adapts to dynamic system contexts, it does not perform real-time MOO. In contrast, Taherinezhad (2022) investigated real-time reconfiguration for a Bi-Copter drone using MOO for dynamic parameter adjustment [2]. This approach advances beyond static models by creating a multi-objective look-up table, facilitating dynamic reconfigurability based on pre-optimized parameters. However, it primarily employs MOO in a preparatory phase, remaining susceptible to dynamic environmental changes.

Our methodology differs considerably from the state of the art in that we use a digital twin in parallel with the operational system usage and update information on system statuses and thus on use cases and operational parameters in real time on the basis of observer models. By continually optimizing parameters on MOO algorithm base, our system gains adaptive capabilities, crucial for the multi-objective design goals and dynamically changing and dynamically changing environments typical of vehicular systems.

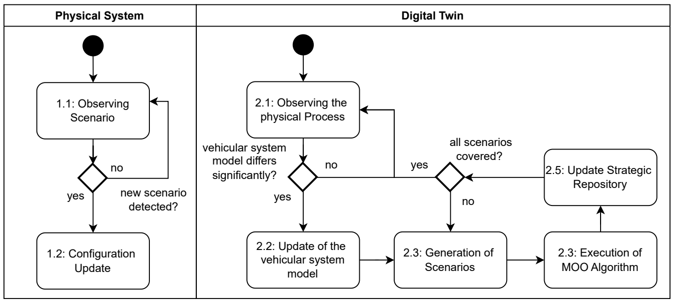

Fig. 2: Overview of the Configuration Scheduling reference process.

IV. Reference Framework

In this section, we propose a novel framework which for the first time solves the challenge of multi-objective autonomous operation of a vehicular system. The framework consists of a reference process (Configuration Scheduling) and an associated reference architecture that enables its implementation. The reference process describes how a physical system can react to changes in scenarios by reconfiguring itself according to a Strategy Repository and how this Strategy Repository is updated to achieve autonomy. The reference architecture, then, defines the necessary entities that are required to execute the process.

A. Configuration Scheduling Reference Process

Similar to gain scheduling in control theory, our methodology employs an enhanced LUT as a strategic repository for pre-computed configurations. Our proposed reference process, which we call Configuration Scheduling is articulated through an activity diagram (Figure 2) delineating two primary swimlanes that are assigned to the physical system and the digital twin as roles. The reference process encompasses two core activities: updating the configuration of the physical system based on current scenarios and constructing the strategic repository. The latter is a systematic exploration of scenarios, leveraging multi-objective optimization algorithms to identify Pareto-optimal system configurations. Each role is responsible for individual tasks.

Physical System: Configuration Scheduling on the Physical System side consists of two main activities: First, the system observes the current scenario (1.1) to update the configuration in case a new scenario is detected (1.2). This simplistic approach ensures a fast adaptation to a changing system context. However, this approach does not account for unforeseen scenarios, such as alterations in dynamics caused by aging or material deterioration. To address this limitation, a Digital Twin is employed, that updates the Strategic Repository dynamically during runtime.

Digital twin: As mentioned in the introduction of this subsection, the process also describes how the Strategy Repository can be constructed by systematically exploring various scenarios and finding Pareto-optimal system configurations. A critical aspect of this approach is the use of simulation models. These models enable the testing of different configurations in a controlled environment, avoiding any potential harm to the actual system. Therefore, our process follows the MAPE-K loop, comprising four stages: Monitor, Analyze, Plan, and Execute.

a) M: Monitor: The 'Monitor' phase in our system focuses on understanding the discrepancies between the system's actual and expected behaviors. This involves comparing system outputs with those of the simulation (Step 2.1).

b) A: Analyze: The 'Analyze' phase represents the trigger, that leads to updating the strategic repository if the system identifies a significant difference between the vehicular system and it's model. This step is crucial for updating the simulation model, which, in turn, is essential for revising the strategic repository.

c) P: Plan: Planning in our system entails updating the strategic repository. This planning step can be decomposed in three steps:

- **Update of the vehicular system model (2.2):** After a significant change in the vehicular system fidelity has been identified, we have to re-align the vehicular system model with the vehicular system. The exact procedure of this is highly dependent on the specific application of the system. The complexity can range from a shift in external system context, requiring parameter adjustments, to more intricate dynamics changes necessitating system identification methods like PILCO [10].
- **Generation of Scenarios (2.3):** After the vehicular system model is updated, scenarios are systematically generated as a basis for the MOO algorithm. This step ensures that the strategic repository will contain a comprehensive range of operational contexts.
- **Execution of MOO algorithm (2.4):** The MOO algorithm uses the updated vehicular system model and the current scenario to identify Pareto Optimal Strategies for the vehicular system. This includes generating output data from the updated model, evaluating the cost on this data and based on that identifying new configurations, that minimize this cost.

d) E: Execute: Execution within our framework is achieved by updating the strategic repository (2.5).

The decision point "all scenarios covered?" acts as a checkpoint within the MAPE-K loop. If all scenarios are not yet covered, the process loops back to scenario generation (2.3), indicating an iterative approach within the 'Plan' and 'Execute' phases to incrementally build and refine the strategic repository. If all scenarios are covered, the system may proceed to the observation of the vehicular system again (2.1), completing the loop.

B. Reference Architecture

The reference architecture defines the entities which are required to operate the Configuration Scheduling process. An overview of those entities is shown in Figure 1.

a) Vehicular System and Vehicular System Model: The vehicular system is the actual unit that interacts with the environment. It typically contains several sensors and actuators. The vehicular system model defines the digital representation of this vehicular system. It can simulate various operational scenarios, generating predictive data for the strategy assessment. It acts as a sandbox for evaluating and optimizing configurations, feeding insights back into the Strategic Repository.

b) Process observer: Focusing on the internal dynamics of the vehicular system, this component monitors performance and operational states. The data collected is used to update the Vehicular System Model and to trigger the action of updating the Strategic Repository.

c) Strategic Repository: Our methodology employs a LUT as a strategic repository for pre-computed configurations. However, unlike in control theory, the scenarios in vehicular systems cannot be encapsulated by a singular operation graph. Instead, we define a scenario as a vector comprising N elements, each representing a distinct aspect of the operational context: $\text{Scenario}_i = \begin{bmatrix} s_{i1} & s_{i2} & \cdots & s_{iN} \end{bmatrix}$ In this formulation, each element s_{ij} signifies a measurable external factor or an operational condition relevant to the system's performance. Correspondingly, a configuration aligned with such a scenario is articulated as a vector of M entries, delineating the system's parameters:

$$\text{Configuration}_i = \begin{bmatrix} c_{i1} & c_{i2} & \cdots & c_{iM} \end{bmatrix}$$

These parameters are tuned to address the specific requirements of the corresponding use-case.

Our Strategic Repository is thus conceptualized as a matrix SR that correlates each scenario vector with its respective configuration vector:

$$\text{SR} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1M} & s_{11} & s_{12} & \cdots & s_{1P} \\ c_{21} & c_{22} & \cdots & c_{2M} & s_{21} & s_{22} & \cdots & s_{2P} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{N1} & c_{N2} & \cdots & c_{NM} & s_{N1} & s_{N2} & \cdots & s_{NP} \end{bmatrix}$$

d) Scenario Observer and Generator: The Scenario Observer continuously monitors the vehicular system, checking for changes and comparing the current scenario with entries in the Strategic Repository. If a new or altered scenario is detected, the observer triggers the system's adaptation process, ensuring that the vehicular system operates under the most suitable configuration for the given conditions. Therefore, while the design of the scenario detection mechanism is application-dependent, its role in ensuring the adaptability and effectiveness of the system remains central. It acts as a critical interface between dynamic operational states and the system's configuration management, enabling a responsive and context-aware operation. The Strategy Generator generates the strategies which are required to fill the look-up table. Parallel to the Scenario Observer, the component's structure depends on the specific application.

e) Cost Evaluation: This component evaluates the efficacy and efficiency of proposed configurations. By analyzing performance, resource utilization, and cost factors, it guides the system towards cost-effective and high-performance operation.

f) MOO algorithm: The MOO algorithm is the critical for refining the Strategic Repository. Utilizing the evaluated cost data from the Vehicular System Model, it iteratively minimizes objectives to determine the most effective configurations for each scenario. This process ensures that the repository's strategies are both efficient and optimal.

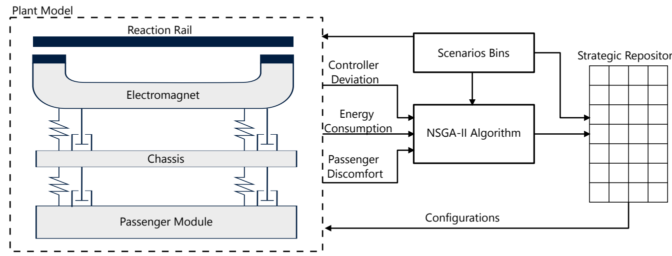

Fig. 3: Overview of the Evaluation Architecture.

V. Experimental Results

This section demonstrates the applicability of our proposed reference framework on an exemplary base. In detail, we apply the framework to a simulation model of a magnetic levitation vehicle. The structure of this section begins with detailing the implementation of the architecture and its related elements as described in section IV-B. This is followed by two evaluations: the first examines the system's response to rapid changes, and the second assesses its reaction to unknown system configurations, thereby demonstrating the framework's autonomous adaptability.

A. Architecture Implementation

The following subsection will provide an overview on how the reference architecture is implemented. Figure 3 provides an overview on that.

a) Vehicular System and Vehicular System model: For our evaluation, we use the vehicular system model also to simulate the actual vehicular system. An overview of this is vehicular system model is shown in Figure 3. It shows a quarter segment of a Hyperloop electromagnetic suspension system, comprising an electromagnet, chassis, and passenger cabinet connected via four spring-dampener systems. The electromagnet generates a force f_{mag} to counterbalance gravitational forces. System control is achieved with a cascaded controller, consisting of an inner loop for linearizing the electromagnet's current dynamics, and an outer loop using a state-space controller to maintain consistent magnet-to-rail distance.

In the initial design, the outer loop's state-space controller, configured with a LQR, simplifies the system by treating it as a single mass. This configuration allows for control over key dynamics, with k_1 affecting air gap distance, k_2 its rate of change, k_3 the magnetic force, and k_I the integral of the air gap distance. Each parameter in this state-space controller is tailored to control a specific dynamic aspect of the system.

Moving away from the optimization of the simplified model, our goal is to concurrently minimize passenger discomfort, controller deviation, and energy consumption. This multi-dimensional approach is aimed at significantly enhancing the system's overall performance and efficiency.

b) Scenario Generation and Detection: In this example, we will focus on the impact of vehicle velocity on the control system within a Hyperloop environment. The Hyperloop track is modular, composed of several short segments to form a complete track. This design is chosen due to manufacturing constraints in producing long uninterrupted segments. Each segment, however, exhibits slight variances. Notably, there is a minor step, approximately 1 mm, between the segments of the reaction rail. To simulate these variations, we introduce a periodically occurring 1 mm deviation in the current air gap between the vehicle and the track. The frequency of these deviations correlates with the vehicle's speed. With a segment length of 5 meters and a vehicle speed ranging from 0 to 900 km/h, this results in excitation system excitation frequencies between $0\ Hz \leq \frac{\text{Velocity}}{5m} \leq 50\ Hz$. For enhanced analysis, we categorize our study into seven discrete scenarios, termed 'scenario bins'. Each bin, denoted by u_i(where i ranges from 1 to 5), represents a different frequency. The specific frequencies for each bin are outlined in the Table I. To select the scenario from the Strategic Repository, a Fast-Fourier-Transformation is used to identify the primary excitation frequency of the system. This allows us to identify the correct configuration of the process.

c) Strategy Repository: The Strategy Repository is a matrix, in which each row shows a configuration and its corresponding scenario. The configuration that we are reconfiguring are the gains of the outer loop controller, whereas the scenarios the velocity bins are Each row will therefore be defined as follow:

$$\text{Strategy Repository}_i = \begin{bmatrix} k_1 & k_2 & k_3 & k_I & u_i \end{bmatrix}$$

with i indexes the corresponding scenario of the scenario bins.

d) Cost Evaluation: The cost evaluation is conducted using three metrics:

$$\text{Cost} = \begin{bmatrix} \text{Passenger Discomfort} \\ \text{Controller Deviation} \\ \text{Energy Consumption} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- **Passenger Discomfort:** This is quantified by the root mean square (RMS) of the vertical acceleration, reflecting passenger discomfort. It is weighted by the Sperling weighting factor, which highlights frequencies typically perceived as uncomfortable by humans [11].
- **Controller Deviation:** Defined as the average of the squared deviations from the target airgap, set at 10mm. This measure reflects the precision and stability of the controller.
- **Energy Consumption:** Calculated as the total power consumed by the electromagnet, minus a baseline power necessary for supporting the static weight. This approach focuses the metric on energy used for active system control, rather than static load maintenance.

e) MOO algorithm and Pareto Point selection: As already stated in section V-A, our goal is update the state-space controller parameters of the outer control loop, that controls the macro-motion dynamics of the system. We therefore perform an optimization of the initial uni-objective LQR controller gains. To do so, we use the NSGA-II genetic algorithm from the pymoo library [12].

The algorithm generates a set of Pareto points, each an optimal solution. We have formulated a metric to automatically

Bin Name	u_1	u_2	u_3	u_4	u_5
Frequency (Hz)	10	20	30	40	50

TABLE I: Scenario bins and their corresponding frequencies

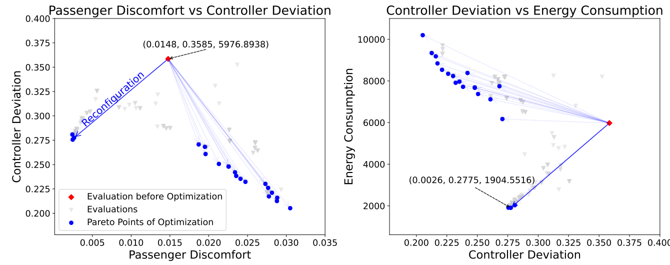

Fig. 4: Projection of the Three-dimensional target space to two dimensions showing the initial configuration (red diamond), optimization results and Pareto point selection for reconfiguration.

select the most suitable Pareto point. Generally, this metric is highly dependent on the use case. In our case, the controller deviation, which is our main indicator for safety should be not higher as the initial evaluation. Furthermore, the variables should perform a trade-off, which we model as maximizing the euclidic distance from the initial point to the Pareto points.

Before applying this metric, it is essential to normalize the dimensions of the Pareto points and the initial evaluation point to ensure a balanced contribution of each dimension to the Euclidean distance calculation. We employ MinMax normalization, scaling each dimension independently so that its values fall within the range [0, 1]. This normalization step is crucial, especially when the scales of the dimensions vary significantly.

$$p_{\text{selected}} = \underset{p_i \in \{p \in P | y_p \leq y_I\}}{\arg\max} \sqrt{(x_i - x_I)^2 + (y_i - y_I)^2 + (z_i - z_I)^2}$$

With p_{selected} representing the chosen Pareto point, that has the biggest distance from the initial point $I = (x_I, y_I, z_I)$, under the given constraint. P is the set of all Pareto points. $p_i = (x_i, y_i, z_i)$ is an individual point in P. The subset $\{p \in P \mid y_p \leq y_I\}$ denotes the set of points in P that satify to the constraint that set their second component (controller deviation) is less than or equal to that of I.

f) Process Observer: In our evaluation of the system's response to a spring break scenario, we operate under the assumption that the updated dynamics are immediately known. This is based on the integration of advanced sensors in each spring, capable of detecting and quantifying changes like spring breakage, thereby facilitating an immediate update of the model.

B. Evaluation of Reaction to Fast Changes

To evaluate the system's reaction to fast changes, we analyzed its performance across the defined scenarios in V-A. Figure 4 shows the optimization of the scenario of 10Hz. The initial system evaluation (marked as a diamond dot) represents

Frequency	x (%)	y (%)	z (%)
10 Hz	-82.11%	-22.60%	-68.13%
20 Hz	-58.35%	-18.87%	-58.37%
30 Hz	-78.37%	-13.47%	-46.67%
40 Hz	-89.35%	-14.23%	-34.55%
50 Hz	-92.21%	-17.18%	-35.09%

TABLE II: Cost Decrease at Different Frequencies

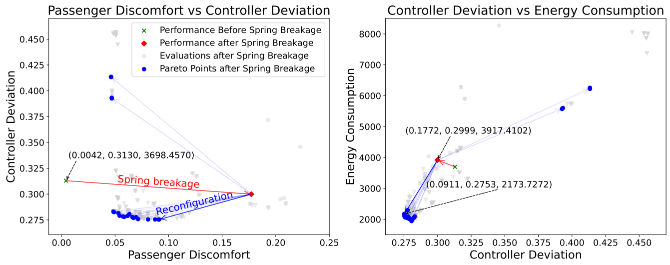

Fig. 5: Reaction and reconfiguration to a spring breakage.

the performance of the single-objective optimization of the simplified model as presented in section V-A. The Pareto points, which are produced by the optimization algorithm are marked as blue dots. We select the most appropriate Pareto point p_{selected} based on our predefined metric from V-A, with the chosen performance transition shown as a solid blue arrow. Other potential solutions, represented by semi-transparent arrows, are acknowledged but not explored in this study. Table II shows how the cost decreases for the Scenarios shown in Table I.

This evaluation underscores the robustness of our Configuration Scheduling methodology, particularly its ability to effectively discern Pareto optimal solutions in a variety of scenarios. Our approach demonstrates a marked improvement in performance across all design goals at higher frequencies. This improvement in performance at higher frequencies highlights the efficacy of the methodology, particularly when compared to traditional single-objective optimization approaches. It demonstrates not only the versatility of the system in handling diverse operational conditions but also underscores the advantages of a multi-objective optimization framework in achieving balanced and improved outcomes across various design goals.

C. Evaluation of Reaction to Unknown System Context

To assess our system's adaptability to unforeseen changes, we extend our scenario by a spring breakage in the connection between the passenger cabinet and the service module as part of the previously optimized 10Hz frequency scenario. Figure 5 depicts the system's performance metrics response to this incident. Initially, indicated by the green x, the system performs optimally. Upon spring breakage, a significant increase in passenger discomfort (by a factor of 42) is observed, while energy consumption slightly rises and controller deviation decreases. This is indicated with the red arrow.

The system, detecting a drastic reduction in passenger comfort, initiates an update to the vehicular system model, triggering the optimization algorithm to compute new controller parameters. This transition, marked by the blue arrow, follows the same rules for selecting the Pareto point as the evaluation in section V-B.

The autonomous reconfiguration effectively mitigates the escalated passenger discomfort to a significant extent. Notably, the new dynamics allow for further optimization of energy consumption and controller deviation beyond the initial settings, showcasing the system's resilience and adaptability to dynamic operational conditions.

VI. Conclusion and Future Works

In this study, we introduced a novel reference framework for autonomous reconfiguration of vehicular systems in the form of Pareto-optimal configurations. This approach is versatile, accommodating multiple use-cases and catering to both rapid and gradual changes. In a detailed experimental evaluation applying the Configuration Scheduling we showed that the optimization significantly improves the performance of a control system for our design goals of minimizing passenger discomfort, controller deviation, and energy consumption. We further showed, that the enormous increase of passenger discomfort in the scenario of a spring breakage scenario could be partially mitigated.

Looking ahead, our objective is to augment this architecture with an advanced goal-management layer. This addition will not only enable the prioritization of specific design goals under varying conditions but also offer the flexibility to emphasize one or more of these goals as needed. Furthermore, we plan to incorporate a feature for setting constraints into this layer, such as minimum design goal thresholds to ensure safety assurances of the system even under diverse operational scenarios.

VII. Acknowledgement

The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the programme of "Souverän. Digital. Vernetzt.". Joint project 6G-life, project identification number: 16KISK002

References

[1] G. K. Pedersen and Z. Yang, "Multi-objective pid-controller tuning for a magnetic levitation system using nsga-ii," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006, pp. 1737–1744.

[2] M. Taherinezhad, A. Ramirez-Serrano, and A. Abedini, "Robust trajectory-tracking for a bi-copter drone using indi: A gain tuning multi-objective approach," *Robotics*, vol. 11, no. 5, p. 86, 2022.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[4] N. Palm, M. Landerer, and H. Palm, "Gaussian process regression based multi-objective bayesian optimization for power system design," *Sustainability*, vol. 14, no. 19, p. 12777, 2022.

[5] A. Gambier and E. Badreddin, "Multi-objective optimal control: An overview," in *2007 IEEE International Conference on Control Applications*. IEEE, 2007, pp. 170–175.

[6] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer*, vol. 36, pp. 41 – 50, 02 2003.

[7] R. Ben Halima, M. Hachicha, A. Jemal, and A. Hadj Kacem, "Mape-k patterns for self-adaptation in cyber-physical systems," *The Journal of Supercomputing*, vol. 79, no. 5, pp. 4917–4943, 2023.

[8] M. Freitag and T. Hildebrandt, "Automatic design of scheduling rules for complex manufacturing systems by multi-objective simulation-based optimization," *CIRP annals*, vol. 65, no. 1, pp. 433–436, 2016.

[9] C. Ionescu, R. Alfredo Cajo Diaz, S. Zhao, M. Ghita, M. Ghita, and D. Copot, "A low computational cost, prioritized, multi-objective optimization procedure for predictive control towards cyber physical systems," *IEEE Access*, vol. 8, pp. 128 152–128 166, 2020.

[10] M. Deisenroth and C. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search." 01 2011, pp. 465–472.

[11] S. Pradhan and A. K. Samantaray, "Evaluation of ride comfort in railway vehicle due to vibration exposure," in *Advances in Structural Vibration*, ser. Lecture Notes in Mechanical Engineering, S. Dutta, E. Inan, and S. K. Dwivedy, Eds. Singapore: Springer Singapore, 2021, pp. 401–410.

[12] J. Blank and K. Deb, "Pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020.