Autonomous Hyperloop Control Architecture Design using MAPE-K

Julian Demicoli, Laurin Prenzel and Sebastian Steinhorst
Technical University of Munich, Germany, Email: firstname.lastname@tum.de

***Abstract*—In the very recent past, there has been a trend for passenger transport towards electrification of the vehicles to reduce greenhouse gas emissions. However, due to the low energy density of battery technology, electrification of airplanes is not possible with current technologies. Here, Hyperloop systems can offer a climate-friendly alternative to short-haul flights but face some technical challenges to be resolved. In contrast to conventional rail systems, the Hyperloop concept uses magnetic propulsion and levitation to operate and has no physical contact with the environment. Consequently, mechanical backup solutions do not suffice to avoid catastrophic events in case of failure. Software solutions must, therefore, ensure fail-operational behavior, which requires autonomous adaptability to uncertain states. The MAPE-K approach offers a solution to achieve such adaptability. In this paper, we present a hierarchical architecture that combines the MAPE-K concept with the Simplex concept to achieve self-adaptive behavior. We impose our autonomous architecture on the controller design for the levitation system of a Hyperloop pod and show that this controller, designed using our methodology, outperforms a conventional PID controller by up to 76%.**

I. Introduction

Commercial aviation contributes 4.9% of the total anthropogenic forcing in the world [1]. Airplanes still rely on fossil fuels, as the energy density of other energy storage systems is too low with current technologies. Hyperloop systems can reach comparable velocities and, therefore, pose a possible replacement for air travel. The Hyperloop concept combines travel in an evacuated tube with a magnetic levitation system to achieve such speeds.

While this combination is promising, it presents a variety of new challenges, especially in the domain of safety. Since the pods have no contact with the track, the system relies on electrical systems to ensure safety rather than on mechanical systems. As a consequence, the permanent functionality of the electrical system and, therefore, the resilience towards changing system context is extremely important.

Another big challenge is to take a faulty pod out of the system since airlocks are required to extract the passengers from the vacuum conditions. Therefore, the pod must be able to reach a certain extraction point. Since the active part of the propulsion system is in the track, pods can only propel with prevailing communication to the operation control station. A failure in communication cannot be excluded since it is wireless and, therefore, prone to temporary failures [2]. In such a case, the pod has to operate autonomously to reach the next extraction point, using only the residual kinetic energy.

Imposing an architecture onto the Hyperloop system that allows it to adapt to unforeseen changes in external or internal system states increases resilience. Consequently, the probability of failure decreases.

To make adaptation possible, the vision of autonomic computing [3] presented architectural adaptation, which has become the most recognized approach to enable self-adaptability. The concept is to use a feedback loop consisting of four steps: Measure (M), Analyze (A), Plan (P), and Execute (E) while including system and environmental knowledge (K) to achieve adaptability. Several publications have extended the initial idea of MAPE-K to allow formal modeling of MAPE-K loops [4, 5], to map MAPE-K to an architecture by using design patterns [6], to control the individual requirements of the stakeholders by adding goal management [7], or by combining MAPE-K with conventional control theory and machine learning [8].

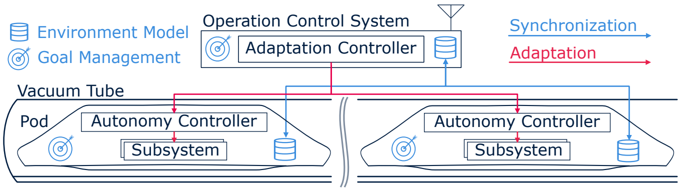


Fig. 1: The operation control system orchestrates the Hyperloop pods on a system level. Its adaptation controller adapts the lower-level autonomy controllers based on global system data. The pods inside the evacuated tube consist of several subsystems that are adapted by the Autonomy Controller based on local system data and goals.

However, none of these approaches targets the safety of the system. While safety is not a requirement for many use cases, Hyperloop systems highly depend on electrical systems to guarantee safety.

For this purpose, we make three contributions in this paper. In Section II, we extend the limited research on the challenges of Hyperloop systems, their requirements for the electrical system, and why autonomy is of such importance. Afterwards, we introduce an architecture in Section III that is organized in two hierarchical levels, the operation control system that orchestrates the entire system, and the autonomy controllers that coordinate the functionality of the pod. Both of those provide adaptive algorithms that use MAPE-K loops to optimize services. The autonomy controllers also incorporate a safety management logic into those MAPE-K loops that extend the simplex architecture [9] to prioritize a safety-driven reflexive adaptation. Figure 1 shows an overview of this architecture. As a third contribution, we evaluate this architecture in Section IV. Using a simplified model of the magnetic levitation system, we show two key aspects of the architecture that increase the controller's performance. The first aspect is to use adaptation in general. The second aspect also includes prior system knowledge collected by other vehicles in the system into the adaptation step. We can show that our architecture reduces the error of the magnetic levitation system by 66% for the scenario of a breaking spring during the operation of the levitation unit. The error reduces by 76% for the track misalignment since the

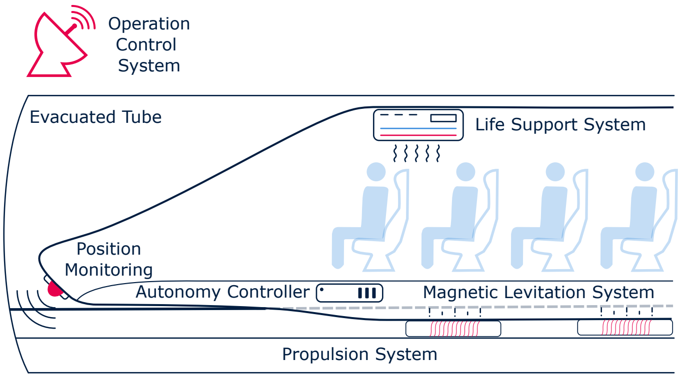


Fig. 2: Overview of the components of a Hyperloop system.

adaptation step also uses prior system knowledge.

II. Challenges of Hyperloop systems

The Hyperloop concept that is depicted in Figure 2 combines travel in an evacuated tube with a contactless pod, which leads to many challenges in the system, especially in the domain of safety, energy transmission, propulsion, and levitation.

We want to present some of the main challenges of those different domains. Subsequently, we derive requirements for a top-level system architecture based on those challenges.

A. Safety

The environment outside the pod is hazardous to the passengers since there is almost no air in the tunnel, making it difficult to extract the passengers in an emergency since the pod must be able to reach a checkpoint that inherits an airlock. Since this is required, even if the connection to the operation control station is lost, the pod must operate autonomously without interaction with the operation control station.

Another challenge of the vacuum environment is the dramatic reduction of convection to cool down the systems in the pod. Therefore, conduction to heat sinks and radiation must be sufficient to keep the vehicle operational. Consequently, the required power must be kept minimal at all times.

B. Energy

Since the pod does not have direct contact with the outside world, the system must provide the required energy in a contactless fashion to the pod. An inductive charging system can provide this, but the amount of power that can be transmitted depends on the velocity of the pod [10]. Consequently, a hybrid solution that includes stored energy is necessary to operate the life-support system even at a standstill.

C. Propulsion System

Since the pod is contactless, a linear motor is required to propel it. To do so, the magnetically levitating systems with the highest technological maturity, namely the SC-Maglev, and the Transrapid, use a long-stator linear motor [10]. Those systems have a static magnetic field in the pod and a moving magnetic field in the track, which ensures that the required energy in the pod is minimal. In the case of Hyperloop systems, this also has the advantage that most of the generated heat retains in the track, where cooling is more convenient.

A big challenge of this system is that the absolute position of the pod has to be available to the motor controller at any given time since this information is required for the control to work. Since this controller is outside the pod, the position must be transmitted wirelessly from the vehicle to the motor controller in real-time.

If the connection between the motor controller and the pod is interrupted, the propulsion system can no longer operate.

D. Magnetic Levitation System

There are several different principles to achieve levitation [11]. Since Hyperloop systems target track lengths of multiple hundreds of kilometers, the most reasonable approach is Electromagnetic Suspension (EMS) which uses controlled electromagnets in the pod to attract steel bars in the track.

The magnetic force equation that acts as a base for the control is highly nonlinear, so deviations from the expected dynamics can lead to unstable control behavior. Such deviations could be internal system context, such as a different weight or mechanical problems in the suspension system, like broken dampeners. Next to the internal system context, a changing environment can alter the controller's performance since misalignments of the track or eigenfrequencies of the mechanical structure of the tube can alter the dynamics. It is, therefore, crucial for such a control system to be able to adapt to changes in dynamics.

EMS has the advantage that the dynamics, like the dampening factor of the system, can be tuned to achieve higher passenger comfort. Higher passenger comfort works by reducing unpleasant oscillations in the passenger cabin, which means that the EMS system follows route changes more slowly. If the control adapts in such a way to certain track and system conditions, this can impair safety. Therefore, deciding on such an adaptation should be made depending on the context of the system.

E. Requirements

Based on the previous discussion, we can identify three main requirements the system must fulfill. In the following section, we will propose an architecture that satisfies those requirements.

- **Safe Adaptation**: Adaptation can impair safety if one of the optimization criteria conflicts with safety. Since the Hyperloop system must ensure safety at all times, it requires a mechanism that restores a safe configuration in such a case.
- **Autonomous Operation**: The system must ensure permanent operation to achieve safety. If the pod loses communication with the operation control station, the propulsion system is not functioning anymore. The pod must then reorganize its resources autonomously to reach a safe state so the passengers can exit the tube.
- **Knowledge Synchronization**: Local knowledge about the track state can provide advantages for an adaptation step. However, this must be available to the pod when it passes over the position. System context must be collected locally and synchronized with the operation control system that can exchange this information between different pods.

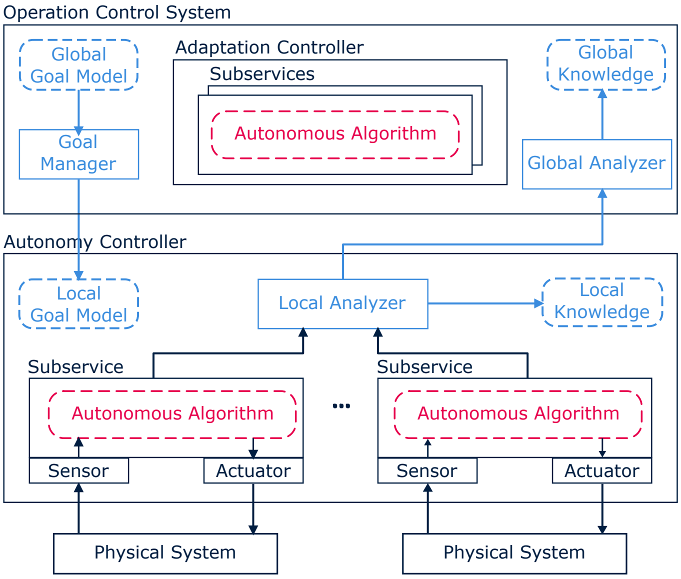


Fig. 3: Hierarchical two-layer architecture proposed for the Hyperloop system

III. System Architecture Proposition

In this section, we present an architecture that covers the requirements which we derived in Section II. Figure 3 depicts the overall hierarchical architecture. The operation control system sits on top to orchestrate the whole system. It is connected to a power grid and, therefore, incorporates more resources like computation power and storage than the pods. Subordinated to the operation control system is the autonomy controller, which is the central computation unit of the pods. To be independent of the operation control system that is connected wirelessly, the autonomy controller also offers an adaptation of algorithms as a service for subordinated control units.

A. Knowledge

Knowledge is the processed usable information that the system has extracted from the data available to it. One can distinguish between a system model and an environment model. It is generated from prior system information and by extracting runtime data from the MAPE-K analyze (A) step. We differentiate between local knowledge of the autonomy controller and global, more complete knowledge of the operation control system.

a) System model: The system model represents the current system state and the behavioral model.

In the Hyperloop context, the suspension highly influences the passenger comfort and the stability of the levitation controller since it influences the system dynamics. It also includes the system's capabilities to influence the external context, such as the dynamics of the actuators.

b) Environment model: The environment model represents the external system context. Due to the amount of data required to model the past and future of the system the amount of data is highly constrained for the autonomy controller.

The track model is such an environment model. For example, one could divide the track into small track sections. For each track section, a certain number of parameters are then available, such as the material, which significantly determines the dynamics, and the position, which can change slightly over time due to wear off. The autonomy controllers collect such information and forward it to the operation control system to use it as global knowledge.

B. Goals

Goals define a formal model to asses system properties. Each stakeholder of the system has its own set of goals. The operator of the Hyperloop system wants a high throughput and low energy consumption to increase the revenue generated. The safety system must keep the safety at a maximum, and the passengers desire a high passenger comfort. The individual subsystem behavior has to be adapted to achieve these goals.

An example of this is to adapt the control parameters of the levitation system to compensate for unevenness in the track more slowly, which leads to increased passenger comfort. Here, a goal model translates semantic information into an arithmetic expression. An optimization algorithm can use this expression to increase the system performance.

C. Operation Control System

The operation control system orchestrates the entire Hyperloop system. It has a global goal model that aims towards high system performance. A goal manager translates those system goals into local goal models for the autonomy controllers since all subsystems influence the fulfillment of the global goal.

The operation control system receives local knowledge from all subsystems to build global knowledge. A pod that notices changes, such as novel misalignments in the track due to wear off, shares this information with the operation control system. Consequently, subsequent pods can prepare adaption steps beforehand and execute updates at the position of the misalignment. Also, it allows for future extensions of algorithms like predictive maintenance.

The operation control station has large amounts of energy and computing power. Because of that, it offers the ability to access those for pods to optimize their subservices. This way, the optimization can be done using global system knowledge and more complex optimization algorithms.

D. Autonomy Controller

The autonomy controller is the primary computation unit of the pod. It includes several subservices that implement the functionalities of the pod, such as the levitation system, the life-support system, and the battery management system. All of those subservices include an autonomous algorithm that can safe-adapt. They monitor (M) and influence the internal states and/or the environment. The subservices report to a local analyzer that processes the data for further utilization as local system knowledge.

E. Adaptive Algorithm

Figure 4 depicts an adaptive algorithm that takes over the planning (P) step of the MAPE approach and optimizes the system performance based on the current state and the system goals. The algorithm uses a digital twin that consists of a simulation framework in combination with a local environment model and a local system model. This simulation framework

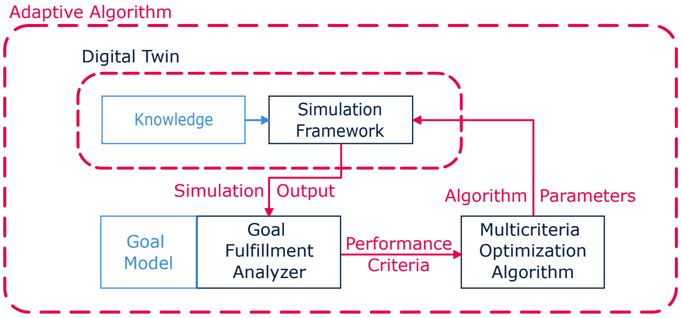


Fig. 4: Optimization framework for an adaptive algorithm.

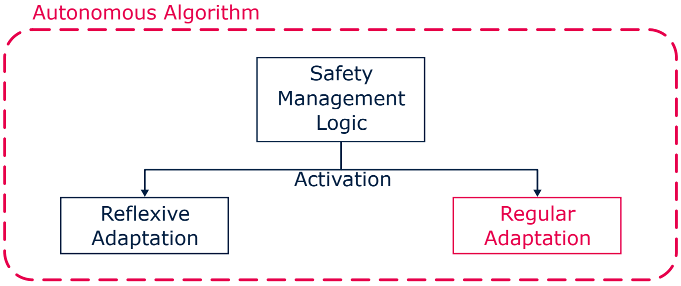


Fig. 5: Structure of an autonomous algorithm.

takes algorithm parameters as input and simulates how the system would behave in such a case. A goal fulfillment analysis system uses these outputs to compute an algebraic representation of the system performance based on the goals. A multi-criteria optimization algorithm then uses this expression to generate optimized parameters for the system.

In a Hyperloop context, the control algorithm of the levitation system could use this methodology. The levitation system mainly consists of a PID controller that calculates a magnetic force output based on distance values. Such a PID controller has parameters that represent the influence of the proportional (P), integral (I), and differential (D) values. Those parameters strongly impact system stability, energy consumption, and passenger comfort. The goal fulfillment analysis system determines to what degree the controller achieves those parameters. Subsequently, the optimization algorithm calculates adapted parameters that shift the system towards another direction, e.g., lower energy consumption.

F. Autonomous Algorithm

Figure 5 illustrates an autonomous algorithm consisting of a safety management logic that can choose between a reflexive and a regular algorithm. The regular adaptation increases the performance of a certain subservice by multiple criteria using the adaptive algorithm. In contrast, the goal of the reflexive algorithm is to only adapt towards safety. Its separately computed parameters can either use the adaptive algorithm to optimize towards the safety criterion or a pre-computed set of safe parameters to make the certification of the algorithm easier. The safety management logic takes this decision based on the system context. It has access to the knowledge of the system and the sensor values to identify if the system is entering an unsafe state. Via the exchange of knowledge, multiple safety management logic units can syncronize to achieve system-wide safety. With the proposed methodology to design autonomous

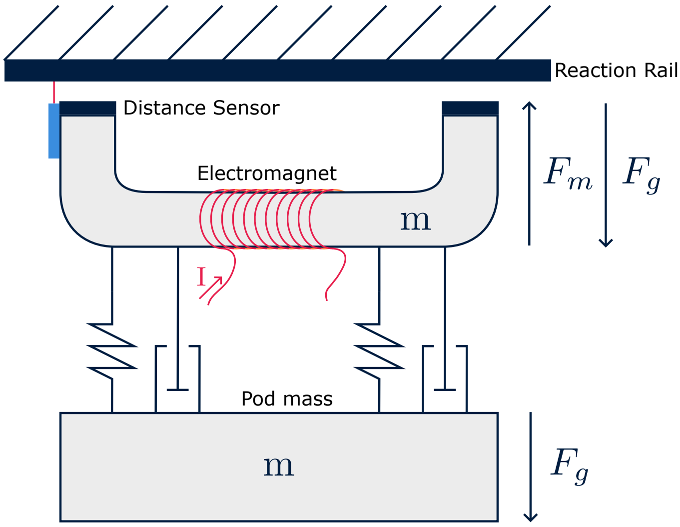


Fig. 6: Physical model of a single magnet module that is used to evaluate the advantages of autonomous adaptation.

architectures, we expect a significant improvement in resilience due to the ability of the system to adapt to uncertain system states. We evaluate the performance based on a case study of the levitation system in the next section.

IV. Evaluation

We evaluate the impact of autonomous adaptation on the system performance made possible due to our architecture. To do so, we implemented two case studies that compare the performance of a conventional PID controller for distance control of an electromagnetic circuit with a PID controller, which we designed using our architecture methodology. This controller self-adapts to changing system context during runtime using the autonomy controller.

The first case study evaluates the impact of a broken spring in the suspension system on the levitation controller. Since the change is only local, the adaptation can be carried out entirely on the autonomy controller and does not require global system knowledge of the operation control system.

The second case study evaluates how the control error of a track misalignment scenario reduces by adapting the controller. This time, we assume that a succeeding pod detected the misalignment position beforehand. Consequently, the adaptation step can directly take place at this position, further improving the controller's performance.

A. System Description

This subsection describes the overall system for the two case studies. First, we introduce the dynamic model, which acts as a basis for the controller. Afterwards, we describe both scenarios for the evaluation in the case study.

a) Model description: In this study, we simulate the closed-loop behavior model of a single magnet module. Figure 6 shows the physical model of this module. The model consists of an electromagnet, a pod, and a reaction rail. The electromagnet and the pod have certain masses that are decoupled by two spring-damper systems. The electromagnet excites a force opposing the force of the gravitational field. This force depends on the electrical current and the air gap between the magnet and the reaction rail.

The open-loop system is inherently unstable. A PID controller computes the electromagnet's current based on the air gap. We introduce a second controller that we designed into our architecture. The second controller has exposed parameters that an autonomous algorithm adapts based on the system states. The metric to which our algorithm measures the performance is a deviation from the target values.

We simplified several parameters in the model of the system. We neglect the coil inductance, which specifies the rate at which the controller can change the current since this allows us to decouple the current controller from our problem. Also, we neglect sensor deviations and delays and assume a controller with an infinitely high control loop frequency. In reality, both controllers would perform worse than in the simulation.

b) Scenario description: Two scenarios are evaluated. In the first scenario, we show the behavior of the system if a spring in the electromagnet breaks. We do this by reducing the spring constant, and thus the force of the spring, by half at $t_{\text{bSpring}} = 15s$. Since the case study does not focus on the adaptation and planning steps, we take the assumption that the analysis and planning step can be done in $0.5s$. The adaptation step is therefore occurring at $t_{\text{adapt}} = 15.5s$. Since our model is two-dimensional, a tilting of the magnet is not taken into account, which could further reduce the performance of the controller.

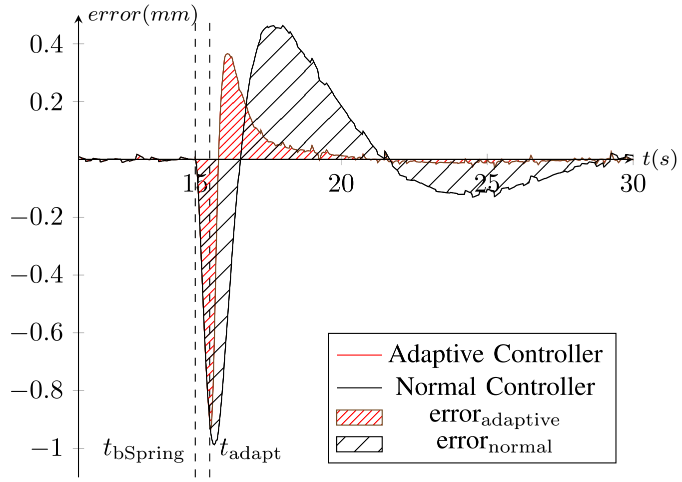
The second scenario depicts the behavior of the system in the event of a misalignment between two track segments of $1mm$. To do so, we set the air gap to $9mm$ at $t_{\text{displ}} = 15s$. Since we take global system knowledge into account, the adaptation step is performed directly at the appearance of the misalignment t_{displ}. Again, due to the two-dimensional model, the system behaves slightly differently than a real system, since one limb of the magnet would arrive earlier at the misalignment.

In both simulations, the adaptation step increases the controller's response time to react faster to the changes. The simulation is carried out for $30s$ for the first and for $50s$ for the second scenario.

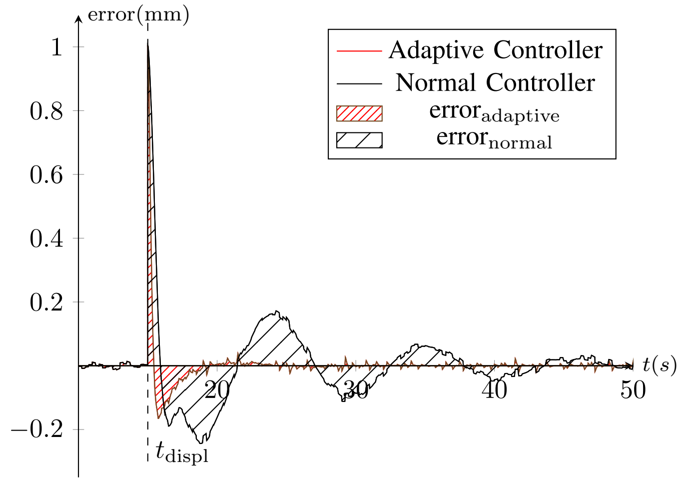
B. Results

Figure 7a shows the error of the two controllers after a spring of the magnet module breaks at $t_{\text{bSpring}} = 15s$. The controller that uses our architecture adapts its control parameters at t_{adapt}, exactly $0.5s$ after the break happened. The simulation shows that the airgap of both simulated controllers increased by approximately $1mm$ immediately after the break occurs. However, the adaptive controller reacts much faster to the new dynamics than the conventional controller. Also, the overshoot is significantly reduced for the adaptive controller compared to the other. Another interesting behavior is that the conventional controller still resonates for approximately $20s$ after the spring breaks, which reduces stability, increases energy consumption, and decreases passenger comfort. The quantitative improvement becomes clear when integrating the two error curves from $15s$ to $30s$. We can thus show that the error of the conventional PID controller reduces by 66% using the new architecture.

Figure 7b shows the error of the two controllers if the system travels over a misalignment of the reaction rails of



(a) Behavior of the normal controller and the adaptive controller after a spring breaks at $t_{\text{bSpring}} = 15s$.



(b) Behavior of the conventional controller and the adaptive controller with a misalignment of the excitation rails of 1mm at $t_{\text{displ}} = 15s$.

Fig. 7: Results of the case studies comparing the controller that is designed using our autonomous architecture concept with a conventional controller.

$1mm$ at $t_{\text{displ}} = 15s$. The adaptive controller changes its dynamics to respond more rapidly directly at t_{displ}. After the displacement, the error of the two controllers jumps to $1mm$, since the airgap changes by $1mm$. Both controllers respond to the misalignment by compensating for the error. However, the conventional controller resonates for $25s$, while the adaptive controller reacts much faster due to the increased response time. By integrating the error curves from $15s$ to $50s$, we can show that the error of the conventional PID controller reduces by 76% due to the adaptive behavior.

C. Discussion

The two case studies show that adaptation enabled by an adaptive-by-design architecture has an advantage on the levitation system of the Hyperloop by increasing the tolerance for deviations of the control parameters. It also shows that information exchange between pods via the autonomy controller can improve the performance of the levitation system. The architecture structure allows for a separation of concerns between data ingestion, communication, adaptation, and execution and yet allows the full benefits of adapting the system to be realized.

V. Related works

Several recent publications address the topic of how to increase the resilience of a system by using of MAPE-K. Towards Resilience by Self-Adaptation of Industrial Control Systems [12] demonstrates the advantage of dynamic adaptation of a system against the conventional approach of restarting the system in terms of quality of service. For Hyperloop systems, compared to an industrial control system, the quality of service is even more important since an impairment could lead to a fatal accident.

Resilience learning through self adaptation in digital twins of human-cyber-physical systems [13] also uses digital twins to adapt human-cyber-physical systems to ensure resilience in uncertain system states. Our architecture uses a multicriteria optimization algorithm on several control parameters that a digital twin simulates. The presented RESILTRON concept instead uses multiple parallel simulations using multi-agent-reinforcement learning in a trial-and-error fashion to find a suitable sequence of policies. This trial-and-error approach can identify configurations to completely unknown system states. However, using optimization algorithms is more directed and, therefore, requires less computation power, making it more suitable for a Hyperloop system, whose pods have limited computation capabilities.

The Simplex architecture [9] describes a similar structure to the autonomous algorithm that includes a complex algorithm and a less complex algorithm that is only tuned for safety. An extension of this approach uses a machine learning algorithm as the complex algorithm to achieve adaptive robot path planning [14]. However, none of those approaches consider the less complex algorithm to be adapted, which impairs the resilience towards uncertain system states.

VI. Conclusion

Hyperloop systems pose new challenges to the safety and, therefore, the resilience of software systems to changing internal and external system states. Self-adaptability enables the ability to deal with unknown system contexts. In this paper, we have introduced a concept that enables architecture-based self-adaptability for Hyperloop systems.

We derived the requirements for this architecture from the main challenges in energy transmission, vacuum technology, electromagnetic levitation, and propulsion the new technology poses.

In two case studies, we compared a conventional PID controller to a second controller that is integrated into the autonomous architecture and can adapt to changing system states. We simulated the break of a spring in a levitation module and the occurrence of a misalignment between two track segments in those case studies. We could show that the performance of the controller that we integrated into the autonomous architecture increased by 66% for the first scenario and by 76% for the second case since we took global system knowledge into account.

The main contribution of our work is mainly on how the architecture of a possible autonomous Hyperloop system could look like. The internal structures of the components that are required to implement this architecture are only described in their necessary characteristics.

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] D. S. Lee et al. “Aviation and global climate change in the 21st century”. In: *Atmospheric environment (Oxford, England : 1994)* 43.22 (2009).

[2] M. Luvisotto, Z. Pang, and D. Dzung. “Ultra High Performance Wireless Control for Critical Applications: Challenges and Directions”. In: *IEEE Transactions on Industrial Informatics* 13.3 (2017), pp. 1448–1459.

[3] J. O. Kephart and D. M. Chess. “The vision of autonomic computing”. In: *Computer* 36.1 (2003).

[4] D. Weyns, S. Malek, and J. Andersson. “FORMS”. In: *ACM Transactions on Autonomous and Adaptive Systems* 7.1 (2012).

[5] D. Weyns and M. U. Iftikhar. *ActivFORMS: A Formally-Founded Model-Based Approach to Engineer Self-Adaptive Systems*. 2019.

[6] R. de Lemos, H. Giese, Hausi A. Müller, and M. Shaw. *Software Engineering for Self-Adaptive Systems II*. Vol. 7475. Springer Berlin Heidelberg, 2013.

[7] V. Braberman, N. D’Ippolito, J. Kramer, D. Sykes, and S. Uchitel. “MORPH: a reference architecture for configuration and behaviour self-adaptation”. In: *Proceedings of the 1st International Workshop on Control Theory for Software Engineering*. ACM, 2015.

[8] D. Weyns et al. *Towards Better Adaptive Systems by Combining MAPE, Control Theory, and Machine Learning*. 2021.

[9] D. Seto, B. Krogh, L. Sha, and A. Chutinan. “The Simplex architecture for safe online control system upgrades”. In: *Proc. of American Control. ACC*. IEEE, 1998.

[10] S. Ding, W. Han, J. Sun, F. Jiang, G. Deng, and Y. Shi. “Modeling and Analysis of a Linear Generator for High Speed Maglev Train”. In: *IEEE Access* 9 (2021).

[11] J. K. Noland. “Prospects and Challenges of the Hyperloop Transportation System: A Systematic Technology Review”. In: *IEEE Access* 9 (2021).

[12] L. Prenzel and S. Steinhorst. “Towards Resilience by Self-Adaptation of Industrial Control Systems”. In: *Conf. on Emerging Technologies*. IEEE, 2022.

[13] E. Bellini et al. “Resilience learning through self adaptation in digital twins of human-cyber-physical systems”. In: *Conf. on Cyber Security and Resilience (CSR)*. IEEE, 2021.

[14] T. B. Ionescu. “Adaptive Simplex Architecture for Safe, Real-Time Robot Path Planning”. In: *Sensors (Basel, Switzerland)* 21.8 (2021).