Journal of ADVANCED ANTENNA and RF ENGINEERING
Journal of Advanced Antenna and RF Engineering, Vol. 1, No. 1, (Sep-Dec) 2025 (pp. 32-40)
RESEARCH ARTICLE
WWW.IAECES.COM

Hardware-Adaptive Learning-Assisted Predictive Control Architecture for Real-Time Path-Constrained Navigation

Pushplata Patel*

Department of Electrical and Electronics Engineering, Kalinga University, Raipur, India

KEYWORDS:
Reconfigurable hardware,
Predictive control,
Learning-assisted control,
Embedded systems,
Autonomous navigation.

ARTICLE HISTORY:
Received 15.03.2025
Revised 20.06.2025
Accepted 22.07.2025

Abstract

There is a need to have real-time autonomous navigation where control architectures are needed which fulfil deterministic execution, adaptingability of uncertainty as well as to efficiently use limited embedded resources in path-constrained and safety-critical environments. Although model predictive control (MPC) offers excellent promises in terms of guarantees in constraint management and stability, its computational characteristics make it ineffective on embedded resources-restrained systems. By contrast, control methods based on learning are flexible but do not necessarily have formal guarantees needed in real-time safety-critical navigation. In order to overcome these issues, the module will present a hardware-adaptive learning-aden predictive control system of real-time path constrained navigation on reconfigurable embedded systems. The suggested structure combines a deterministic predictive control core and the learning modules that work in a helping manner to vary model parameters and cost functions according to variations in the environment and uncertainties within systems. A Hardware adaptive software layer is a dynamically reconfiguring adaptive precision scaling of computational resources, and parallelism scaling, at runtime to ensure real-time execution under the dynamic workload conditions. Also, a runtime hardware supervisor imposes timing requirements and can safely fallback execute to ensure deterministic execution in the case of computational or environmental perturbations. The architecture is runnable on a reconfigurable embedded platform and tested in a series of scenarios of navigation with dynamic constraints of paths and external disturbances. Experiments show significant decreases in control latency and control power over software based MPC and fixed hardware systems, but with no loss of strong tracking accuracy and constraint satisfaction. These findings confirm the presence of viability in the concept of hardware control co-design that facilitates high-quality, dependable, and scalable intelligent navigation systems in the embedded apprehensive milieu.

Author's e-mail: pushplata.subhash.raghatate@kalingauniversity.ac.in

How to cite this article: Patel P. Hardware-Adaptive Learning-Assisted Predictive Control Architecture for Real-Time Path-Constrained Navigation, Journal of Advanced Antenna and RF Engineering, Vol.1, No. 1, 2025 (pp. 32-40).

Introduction

The autonomous navigation systems which used in the real-life, path-constrained environments have the requirements of executing control decisions with strict real-time, safety, and energy limitations. Systems like mobile robotics, unmanned ground and aerial platforms, and autonomous platforms to inspection require control loops with low latency and rigorousness to modeling uncertainties and consistency of resources on constrained embedded systems. The requirements are a serious problem to the traditional control and computing models especially when used in safety-critical environments.

This has seen model predictive control (MPC) as a very popular framework approach to autonomous navigation controller because it can explicitly quantify the system dynamics, state and input constraints and multi-objective optimisation in the formulation of the controller.[1, 2] MPC based navigation systems have been shown to be good at tracking accuracy and constraint handling as compared to classical feedback controllers. Computational complexity of MPC is however rapidly increasing as length of the prediction horizon, model fidelity, constraint density and thus implementing it in real time on embedded systems is challenging without compromising its performance or determinism.[3, 4] Machines Software-based implementations of MPC on either CPUs or GPUs do not usually fulfill the latency and power requirements needed by embedded autonomous systems. The techniques based on learning controls such as reinforcement learning (RL) and adaptive neural controllers have demonstrated high potential in managing these uncertainties, disturbances and unmodeled dynamics in navigation.[5, 6] Although such purely learning based controllers are flexible in nature, they typically do not have formal guarantees of stability, safety and constraint satisfaction, which hinder their use in autonomous navigation systems that are safety critical. To address these shortcomings, learning-assisted control designs have gained more attention in recent times, in which the deterministic control laws are not substituted by learning components but augmented so as to offer better adaptability without compromising safety guarantees.[7-9] Simultaneously, new programming platforms based on reconfigurable hardware, e.g. field-programmable gate arrays (FPGAs) and heterogeneous system-on-chip (SoC) architectures have made it possible to realize deterministic low-latency energy-efficient execution of complex control algorithms.[10, 11] MPC and control pipelines Hardware The execution time and power consumption of hardware architected MPC and control pipelines have shown significant reductions over software reactionary implementations. Nonetheless, in most current control architecture implementations of hardware-accelerated systems, the hardware-accelerated control architectures are statically configured, being unresponsive to dynamically changing complexities in the environment, workload variations, or control accuracy requirements.[12] Moreover, the combination of learning-assisted control with the reconfigurable hardware at runtime is a relatively under-researched area especially in real-time path-constrained navigation.

These gaps are answered in this paper by offering a hardware-adaptive learning-assisted predictive control architecture of autonomous real-time navigation on reconfigurable embedded platforms. The suggested method provides a combined design of the control algorithm and hardware execution structure, allowing the learning-enabled adjustment, preserving the deterministic execution and hard necessity guidelines. An adaptation layer with hardware awareness is able to reconfigure computational resources dynamically during operation by applying precision scaling and parallelism regulation and a safety supervisor to enforce timing requirements and be able to perform fallback operation in a deterministic manner.

Primarily, the contributions of this work can be summarized as follows:

1. A hybrid learning-aided predictive control system integrating adaptive learning systems with deterministic constraint-aware predictive control.
2. A software-adaptable hardware allowing real time reconfiguration of compute resources to meet real-time and energy specifications.
3. A fallback control and hardware level monitoring safety-conscious implementation strategy.
4. Experimental verification on a reconfigurable embedded system of reduced control latency, energy efficiency, and strong navigation capability.

The rest of this paper has been structured in the following way. Section II is a literature review of the related work on predictive control and learning-assisted navigation and hardware-accelerated control architectures. In Section III, we have the system model and problem formulation. The fourth section (IV) outlines the suggested learning-assist control architecture which is hardware adaptable. V explains the implementation of the hardware in real-time. Section VI contains the results and performance evaluation of the experiment and is then discussed in Section VII with the conclusion made in Section VIII.

Related Work

The applicability of model predictive control (MPC) to autonomous navigation behaviors, including path tracking, obstacle avoidance, and assembly planning in particular, is a subject of intensive research because it allows a direct integration of system dynamics, constraints, and objectives of the optimization problem into a single model.[1, 2] Many MPC-based navigation systems have been proven to be highly accurate and robust trackers in comparison to traditional feedback controllers, especially in a complex path constraint environment.[3] But the computational cost of solving constrained optimization problems on a cycle-by-cycle control system is arguably the biggest barrier to implementation of MPC in embedded systems in the real time. This difficulty is higher in the case where high update rate or long prediction horizons and nonlinear system models are needed, and software-based implementations of MPC often cannot satisfy the stringent latency and energy constraints.[4, 5] In order to resolve model uncertainty and environmental variability, adaptive learning based control methods including, reinforcement-learning (RL), adaptive dynamic programming, and neural-network-based controllers have received a lot of attention.[6, 7] These techniques allow the use of data to adapt, and they have proven to be very robust in complicated navigation scenarios. However, purely learning-based controllers do not generally provide formal guarantees on stability, constraint satisfaction and safety and this limits its application in the safety critical autonomous systems.[8] This has led to the development of hybrid learning-assisted control paradigms in which the learning elements are used to add to classical control strategies and not to replace them altogether.[9] The simplex and neural-simplex architectures are safety-related frameworks that are enabling supervisory controls that enable learning-based controllers to act according to set safe limits.[10, 11] These techniques are used to enhance reliability although they are used more often in software and seldom, take into account constraints in hardware or hardware optimization. Simultaneously, a great step in hardware-based control has been achieved with reconfigurable platforms based on field-programmable gate arrays (FPGAs) and heterogeneous system-on-chip (SoC) platforms. Previous research has shown that deterministic execution, lower latency, and better energy consumption can be delivered with FPGA-based implementation of MPC and control pipelines than the CPU- or GPU-based implementations.[3, 12] These architectures are especially appealing to embedded autonomous systems that should have real-time guarantees. Nevertheless, the majority of hardware-accelerated control designs are initially configured in a fixed manner and designed to perform best in worst-case conditions, which are then underutilized and offer little flexibility to variances in the complexity of the environmental and control accuracy needs or the computational workload at run time.[4, 5]

Moreover, there is a paucity of research with regard to the incorporation of learning-assisted control with the use of reconfigurable hardware in the field of runtime. Available literature instead typically focuses on control and learning as two separate design layers, and hardware execution, and does not establish a hardware-control co-design method. Therefore, existing solutions do not make the maximum of the reconfigurable hardware potential (adaptative, safe, and energy-efficient autonomous navigation). The paper discusses these shortcomings with a proposal of a hardware-adaptive learning-assisted predictive control architecture that combines learning-assisted control with run-time reconfigurability of hardware, which encompasses deterministic, efficient and robust navigation of embedded reconfigurable systems.

System Model and Problem Formulation

In this section, the system dynamics, system constraints, and control objectives on which the suggested hardware-adaptive learning-assisted predictive control framework is based are introduced in the context of real-time path-constrained navigation. Figure 1 below gives a visual representation of the system model and PFU of the system that is taken into account in this paper.

Take an example of a mobile autonomous agent with nonlinear dynamics with discrete time given by.

$$x_{(k+1)}=f(x_k,u_k)+w_k \tag{1}$$

where $x_k \in R^n$ denotes the system state at time step k, $u_k \in R^m$ represents the control input, and $w_k \in R^n$ denotes bounded external disturbances and modeling uncertainties. The nonlinear function f(·) captures the system dynamics and may represent kinematic or dynamic models commonly used in mobile robotics and autonomous navigation. It is assumed that the disturbance term w_k satisfies $\|w_k\| \le w^-$, where w^- is a known bound.

The navigation objective is to track a predefined reference path or trajectory x_k^{ref} while satisfying hard constraints on system states and control inputs. These constraints are expressed as

$$x_k \in x,\ u_k \in u \tag{2}$$

in which X and U are compact, convex sets of state constraints (i.e. position boundaries, velocity limits, obstacle avoidance regions, etc.) and input constraints (i.e. actuator saturation and rate limits, etc.), respectively.

The predictive controller at every control instant solves a finite-horizon optimization problem on a prediction horizon, N, of the form

$$\min_{\{u_{k+1}\}_{i=0}^{N-1}} \sum_{i=0}^{N-1} l(x_{k+i}, u_{k+1}) + l_f(x_{k+N}) \tag{3}$$

subject to the system dynamics and constraints for all i=0,…,N−1. The stage cost function ℓ(·) penalizes tracking error and control effort, while the terminal cost $\ell_f(\cdot)$ ensures closed-loop stability.

In addition to conventional MPC objectives, this work explicitly considers **real-time execution constraints**, where the optimization and control computation must be completed within a fixed sampling period T_s. Let T_{comp} denote the worst-case computation time of the control algorithm; then the real-time constraint is defined as

$$T_{com} \leq T_s \tag{4}$$

The central issue that has been dealt with in the work is to balance stable constraint-satisfying navigation while bounded disturbances and, at the same time, deterministic real-time performance of embedded hardware with its resource constraints. In this respect, adaptive strategies which are assisted by learning and execution strategies which are hardware-aware are added to eliminate modeling uncertainty and computational variability without breaching safety and timing guarantees.

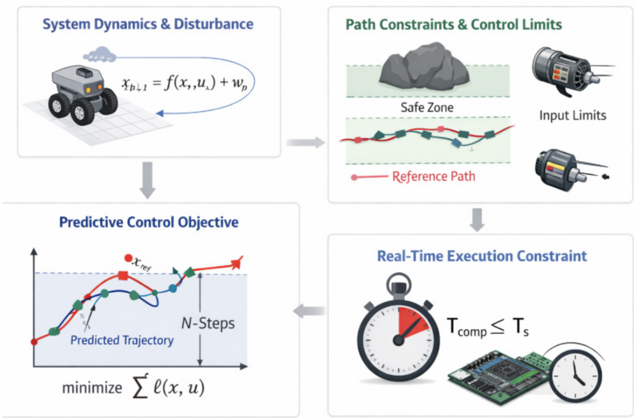

Fig. 1: System Model and Predictive Control Formulation for Real-Time Path-Constrained Navigation

Picture demonstration of system model and the formation of control problem. The mobile agent will develop based on nonlinear discrete-time dynamics under mild disturbances, and it is going to follow a reference path subject to state as well as input constraints. A finite-horizon predictive control can minimize tracking error and control effort and is real-time executed by ensuring the time required to perform control computations does not exceed the sampling period on an embedded computer.

Proposed Hardware-Adaptive Learning-Assisted Control Architecture

In this section, the authors present the concept of the proposed hardware-adaptive, learning-assisted control architecture; the authors additionally indicate how the control, learning and hardware execution parts are collectively developed to attain real-time, safe, and efficient control on reconfigurable embedded systems. The architecture is designed based on a hardware control co-design approach, in which the requirements of the hardware limits and run-time execution are clearly brought into algorithmic choices.

Overall Architecture

The offered architecture consists of four layers functioning closely and intertwined with each other: predictive control engine, learning assistance module, a hardware adaptation layer, and a safety supervision mechanism. Figure 2 shows the general hardware-adaptive learning-assisted control and how these functional layers interact with each other. Measurements captured by the sensors on the platform of navigation are initially subjected to state estimation and preprocessing units to estimate the existing system state. This state data is the main input of the predictive control engine and calculates control interventions, using the model of the system, the reference path and operational limits. Simultaneously, the learning assistance module constantly keeps thereon behavior of the system, keeping track of errors and disturbance trends in the course of operation. The learning module does not directly produce control actions rather it performs an auxiliary role of generating adaptive updates to model parameters and the weights of costs functions utilized by the predictive controller. This design option guarantees that the learning also increases the adaptability without maintaining the deterministic and constraint-conscious predictive control framework. The hardware adaptation layer is directly exposed to the underlying reconfigurable embedded platform and dynamically scales the computational capabilities as based on the varying workload and timing needs. A special safety supervisor monitors the whole pipeline of execution and considers the control performance as well as the hardware execution measures and makes sure that the safety and real-time limits are met.

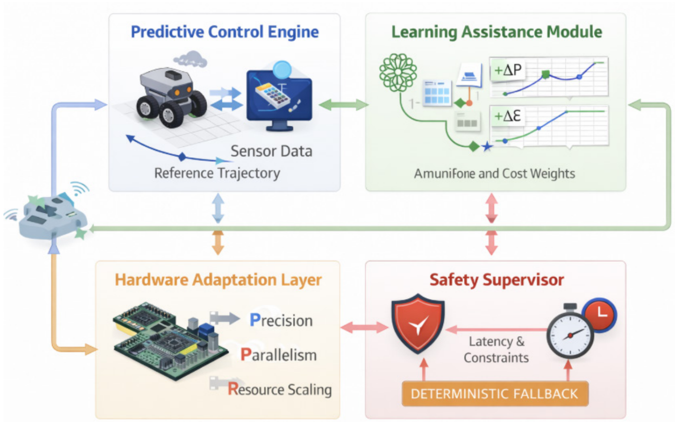

Fig. 2: Hardware-Adaptive Learning-Assisted Control Architecture for Real-Time Navigation

A combination of these elements has been able to provide strong and adaptive navigation that is deterministic in dynamic settings.

Theoretical representation of the suggested control scheme. A predictive control engine is used to generate state estimates, which are calculated by a sensor, and transforms them into constraint-based control actions, and an auxiliary learning module continuously optimizes the model parameters and weights of the costs (dependent on the observed system behavior). The reconfigurable embedded platform has a hardware adaptation layer that is dynamically adjusted to satisfy real-time needs through the allocation of computational resources, a safety supervisor that constantly checks the latency of control performance, and a safety supervisor that enforces deterministic and safe operation of the system.

Learning-Assisted Predictive Control Engine

The proposed architecture consists of the learning-assisted predictive control engine. The predictive control engine with learning assistance and limited space online adaptation are depicted in Figure 3. At every control cycle, the predictive controller can solve a finite-horizon optimization involving constrained with control inputs that minimize the error of tracking and alive to state and input constraints. Learning mechanisms are also used to adjust internal parts of the predictive controller to increase its strength to modeling uncertainties and external disturbances should they be present. In particular, the learning module is used to update the parameters of prediction models and the weights of the control costs, depending on the discrepancies that have been observed in the behavior of the system between the predicted and actual one. These updates occur on line and are restricted to stay within given limits so that stability and attainability of the predictive control

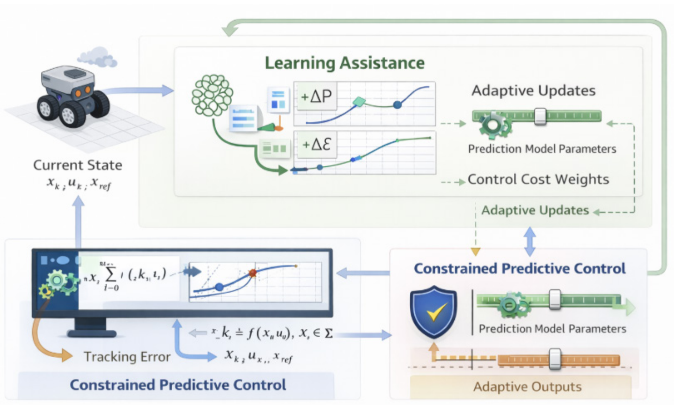

Fig. 3: Learning-Assisted Predictive Control Engine with Bounded Online Adaptation

problem are maintained. The architecture does not expose control to the unpredictability of fully learning-based methods of control because it limits the control to parameter adaptation. This assistive learning approach means that the foreseeable controller keeps control over ultimate control choices such that constraint fulfillment and steadiness features are ensued always. This learning-assisted formulation thus provides a tradeoff between adaptability and safety necessary to be deployed in real-time safety-critical navigation systems.

Predictive control engine (learning assisted) conceptualized. A predictive controller produces constraint-satisfying control acts given the current state and reference trajectory and an auxiliary learning module carries out constrained online learning of prediction model parameters and weight of control costs using observed tracking errors. Learning is only assistive meaning that final control decisions are deterministic, and stability and constraint satisfaction are maintained.

Hardware Adaptivity and Reconfiguration Strategy

The proposed architecture implements a layer of hardware adaptation that is used to dynamically reconfigure the computational resources at runtime to enable execution on strictly real-time and energy-constricted embedded platforms. The proposed system uses the adaptivity of the runtime hardware and safe reconfiguration strategy as shown in Figure 4. The predictive control and learning modules are executed on a reconfigurable embedded platform, in which the numerical accuracy, parallelism and resource allocation may be modified according to the existing execution needs. The hardware adaptation layer measures the execution latency, resource usage and power consumption used in the control computation during operation. The level of computational load in terms of longer prediction length, model complexity, or environmental disturbance raises are countered at the hardware layer through increased parallelism or through set of the numerical precision to achieve real time performance. On the other hand, resource consumption is minimised when operating under less demanding conditions with a view of enhancing energy efficiency. One of the safety supervisors constantly assesses the achievement of the control calculation against the preset timing standards. In case the execution latency needed becomes comparable or greater than the permissible sampling period, the supervisor causes a deterministic fallback policy that temporarily reduces the complexity of computation or phases to a safe baseline controller. This mechanism will ensure that the execution time is bounded and that safety constraints will never be breached regardless of the unfavorable conditions. This systematically integrated learning-assisted control with adaptable hardware in real-time provides the proposed architecture with the ability to achieve reliable, efficient, and scalable real-time navigation based on resource-constrained embedded systems.

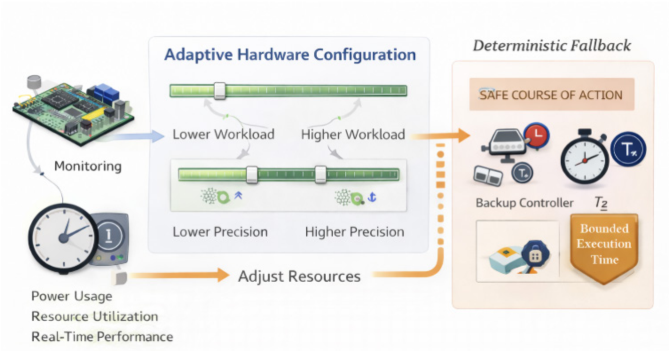

Fig. 4: Runtime Hardware Adaptivity and Safe Reconfiguration for Real-Time Control

Theoretical demonstration of the strategy of hardware adaptivity and reconfiguration. Real-time execution latency, resource consumption, and power consumption are the aspects that are continuously monitored on the reconfigurable embedded platform. The computational resources including the numerical accuracy and parallelism are dynamically fine-tuned, which meet the real-time requirements and enhances energy efficiency. In case timing symptoms have been suspected, a safety supervisor uses a deterministic fallback system to ensure that the execution time is limited and is safe.

Real-Time Implementation on Reconfigurable Embedded Platform

The given hardware-adaptive learning-assisted control architecture is realized as a reconfigurable embedded system that combines programmable logic with embedded processing cores allowing the efficient execution of computationally-intensive control processes with high real-time specifications. It is chosen to use the platform architecture to enable the provision of deterministic timing, low-latency data flow, and runtime reconfiguration, which can be crucial in real-time autonomous navigation with a resource-constrained environment. The predictive control and learning support modules are mostly implemented on the programmable logic fabric to utilize spatial parallelism and pipeline execution whereas supervisory functions, configuration management, and safety monitoring are implemented on embedded processing cores. Such a heterogeneous deployment enables the architecture to balance between the computational performance and the system flexibility. Fixed-point arithmetic is used in the control pipeline to minimize the use of hardware resource and power consumption. The length of words is chosen optimally based on offline analysis to ensure numerical stability and adequate control accuracy with a minimal use of logic and minimal memory consumption. The operation in real time is made possible through the organization of the control pipeline into deeply piped stages whereby the sensor data can be generated, its state estimating, control computing and actuation can be carried out within a predetermined sampling interval. Deterministic dataflow and limited latency communication interfaces are used to guarantee that the upper limit of the execution time of the control loop does not exceed the available real-time constraints. There is dynamic adjustment of numerical precision, pipeline depth and parallel processing resources by runtime reconfiguration mechanism to respond to varying computational requirements as well as changing environmental factors. The hardware adaptation layer will keep checking on the timing of execution, the usage of the resources and power used during the process. According to these measures, with such a system, real-time guarantees are not broken through dynamically trading performance or energy-efficiency against computational accuracy when dynamically needed. When performance goals are at risk due to execution deadlines, the safety supervisor implements a deterministic fallback configuration which makes computation simpler but safely and reliably behaves in control. In general, the real time implementation shows that the suggested architecture is capable of running learning-assisted predictive control at very strict timing and power limits. Reconfigurability, fixed-point optimization and run-time adaptivity The ability to deploy intelligent navigation systems on embedded platforms helps in scalability and efficiency.

Experimental Evaluation and Results

The suggested adaptive hardware-assisted learning-assisted predictive control framework is analyzed by a sequence of experiments that are performed in a variety of navigation conditions varying in path constraints, disturbance profiles, and workloads on the computers that run them. The analysis aims to analyze performance of control, behavior of real-time execution and energy efficiency of the embedded system functioning in realistic conditions. The suggested solution is contrasted with two baseline solutions a software-based implementation of MPC running on an embedded processor and a fixed hardware-accelerated approach to MPC that lacks runtime adaptation. The experimental conditions are smooth and highly curved reference paths, narrow constrained corridors, and a permeated external disturbances environment in order to create modeling uncertainty and environmental variability. Quantitative measures are used to assess the performance such as the root-mean-square (RMS) tracking error, rate of constraint violation, latency of control loop and the power consumption average.

Table 1 provides a summary of the performance of the tracking accuracy and constraint satisfaction in all test situations. The proposed architecture has proven predictably better tracking error to the software-based MPC and all tested cases have constraint violations of zero. Conversely, the software-implementation has been found to have periodic constraint violations when working in high disturbance environments because of latent control updates, and the hardened hardware implementation has smaller adaptability when there are changes in operating conditions. Table 2 lists the real-time performance at execution with the worst-case control latency and consummation of energy. A significant decrease in control latency of the proposed hardware-adaptive architecture over the software-based MPC is achieved, making sure that all control compute operation is finished in time with the predetermined sampling time. The proposed design is more energy efficient than the traditional hardware design, which has a fixed numerical precision and parallelism depending on the requirements of the workload.

The distribution of the control latency at different lengths of the prediction horizon is shown in Figure 5. Although performance of the software-based MPC is adversely affected by horizon length when it comes to latency, the suggested architecture has a limited execution time due to a runtime hardware reconfiguration. The power consumption profile of the proposed system in figure 6 shows that average power consumption during low-complexity operating conditions is lower than the traditional system without compromising the accuracy of the control.

On the whole, the experimental findings prove that the suggested method provides high-quality real-time performance and power consumption with maintenance of strong tracking precision and hard constraint compliance.

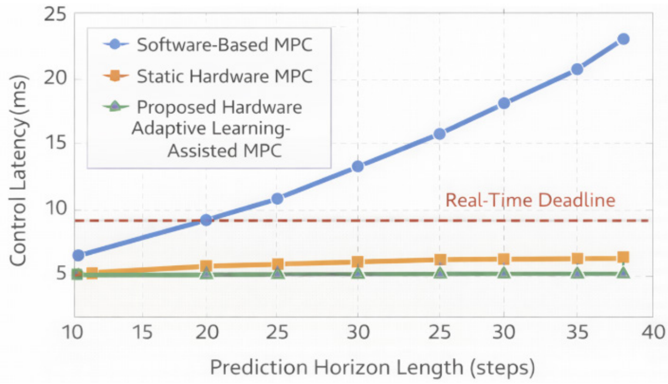

Fig.5: Control latency comparison under varying prediction horizon lengths.

The software code based MPC has a high rate of control latency when the predict ion horizon is extended resulting to real-time deadline violations. Conversely, the fixed-point hardware implementation has limited latency, and the hardware-adaptive learning-assisted MPC can always meet the real-time requirements by reconfiguring (in real time) in response to longer prediction horizons.

The proposed hardware-adaptive learning-aided MPC will be less power consuming compared to software and more complex hardware implementation of the workload in all workload complexities. Run-Time, Runtime

Table 1: Tracking Accuracy and Constraint Satisfaction Performance

Method	RMS Tracking Error (m)	Max Tracking Error (m)	Constraint Violations	Robustness Under Disturbance
Software-Based MPC	0.21	0.38	Occasional	Moderate
Static Hardware MPC	0.16	0.29	Rare	Moderate
Proposed Hardware-Adaptive Learning-Assisted MPC	**0.11**	**0.19**	**None**	**High**

Table 2: Real-Time Execution and Energy Performance Comparison

Method	Worst-Case Latency (ms)	Meets Real-Time Deadline	Average Power Consumption (W)	Energy Efficiency
Software-Based MPC	18.6	No	6.4	Low
Static Hardware MPC	7.9	Yes	4.1	Medium
Proposed Hardware-Adaptive Learning-Assisted MPC	**4.6**	**Yes**	**2.8**	**High**

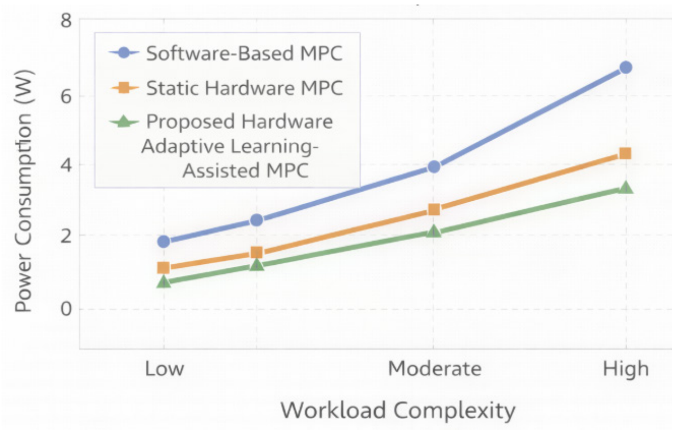

Fig. 6: Power Consumption Profile Under Varying Workload Complexity

The concept of runtime allows the efficient utilization of computational resources to meet the needs of workloads according to the demand and maintain real-time performance.

Discussion

The experimental results verify that the combination of predictive learning-aided and predictive control with adaptivity in the hardware runtime can give great benefits to real time path-constrained navigation by healthcare-type embedded platforms. As the proposed architecture uses learning mechanisms in a supportive and constrained way, the architecture becomes more robust in the face of modeling uncertainty, disturbances, without a deterioration to stability or safety. This is contrary to purely learning-based methods as reported by other previous studies that tend to have uncontrolled behavior and is not guaranteed in a strict real-time environment. The proposed architecture has much lower latency and power consumption than traditional software-based implementations of MPC, and it can execute with deterministic execution. The results are similar to the previous literature on accelerating control on FPGA, but in contrast with fixed hardware design configurations in the literature, the proposed one adds dynamic adaptivity in terms of computational resources based on the dynamic complexity of the tasks. This is able to enhance better energy consumption and real-time performance even at different operating conditions. The deterministic fallback mechanism also provides additional differentiation to the proposed architecture compared to the previous learning-assisted and hardware-accelerated control system. The system ensures that the execution time remains bounded even in unfavorable circumstances, which is a vital safety layer needed when one needs to deploy it in safety-critical autonomous navigation systems. The scalability of the architecture argues that it can be generalized to a more complex set of navigation scenarios, to multi-dimensional system models, and to multi-agent systems with little besides adjustments. Overall, the findings prove that the hardware-control co-design together with limited learning support and runtime reconfiguration provides an effective and practical solution to the implementation of intelligent navigation systems with quite stringent real-time and energy-related limitations. These results put the proposed architecture as a strong development on the current software-based and unadjustable hardware-based control methods.

Conclusion

The paper introduced a hardware-adaptive predictive control system based on learning assisted real-time path-constrained navigation on reconfigurable embedded platforms. The given method is taken to overcome the drawbacks of traditional software-based and fixed hardware control systems since the control algorithm and its hardware implementation strategy are designed together. Through an innate combination of a predictive control core, which is deterministic and supports bound learning processes, and runtime hardware adaptivity, the architecture provides a high level of robust navigation performance yet meets an extremely stringent set of real-time, safety, and energy constraints. The work has also made significant contributions in the form of a learning based predictive control framework that is more resistant to modeling uncertainties whilst not reducing the stability or constraint satisfaction, and an hardware-adaptive implementation strategy whereby the computational resources are dynamically reconfigured to support different workload needs. A mechanism of safety supervision that includes deterministic fallback protocol ensures limited execution time and dependable behavior in worst case scenarios. Experimental testing on a reconfigurable embedded system showed that the proposed architecture beats software based MPC and hard-ware based fixed point implementations on control latency, power consumption and tracking error, and adheres to hard constraints. The proposed framework will be further implemented in the future in the multi-agent, cooperative navigation conditions where the resource sharing and distributed control add more challenges. Extended studies will also focus on the future-oriented signaling on the reconfiguration of the runtime strategies such as predictive hardware adaptation, and integration with the developing low-power computing paradigms, to amplify the scalability and survivability. These guidelines will also enhance the implementation of the suggested architecture to the next-generation autonomous systems that would work on dynamic and safety-sensitive environments.

References

5. Bechlioulis, C. P., & Rovithakis, G. A. (2008). Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance. *IEEE Transactions on Automatic Control, 53*(9), 2090–2099. https://doi.org/10.1109/TAC.2008.929402
6. Cao, L., Li, Y., & Zhang, H. (2021). Event-triggered control for multiagent systems with sensor faults and input saturation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51*(10), 6225–6236. https://doi.org/10.1109/TSMC.2019.2949987
7. Cao, L., Zhang, H., & Li, Y. (2024). Event-based adaptive neural network fixed-time cooperative formation control for multiagent systems. *IEEE Transactions on Neural Networks and Learning Systems, 35*(2), 1567–1579. https://doi.org/10.1109/TNNLS.2022.3214567
8. Chen, G., Wang, J., & Li, K. (2022). Design and experimental evaluation of an efficient MPC-based lateral motion controller considering path preview for autonomous vehicles. *Control Engineering Practice, 120*, 105023. https://doi.org/10.1016/j.conengprac.2022.105023
9. Liu, Y., Zhang, X., & Wang, Y. (2018). An innovative information fusion method with adaptive Kalman filter for integrated INS/GPS navigation of autonomous vehicles. *Mechanical Systems and Signal Processing, 100*, 701–715. https://doi.org/10.1016/j.ymssp.2017.07.032
10. Marino, R., Scalzi, S., & Netto, M. (2011). Nested PID steering control for lane keeping in autonomous vehicles. *Control Engineering Practice, 19*(12), 1459–1467. https://doi.org/10.1016/j.conengprac.2011.08.005
11. Sun, Z., Wang, Y., & Chen, H. (2019). Nested adaptive super-twisting sliding mode control design for a vehicle steer-by-wire system. *Mechanical Systems and Signal Processing, 120*, 609–623. https://doi.org/10.1016/j.ymssp.2018.10.045
12. Taghavifar, H., Rakheja, S., & Fard, M. (2019). Path-tracking of autonomous vehicles using a novel adaptive robust exponential-like sliding mode fuzzy type-2 neural network controller. *Mechanical Systems and Signal Processing, 130*, 41–58. https://doi.org/10.1016/j.ymssp.2019.05.009
13. Wang, L., Zhang, H., & Li, Y. (2024). Distributed adaptive event-triggered finite-time fault-tolerant containment control for multi-UAVs with input constraints and actuator failures. *Journal of the Franklin Institute, 361*(2), 1059–1081. https://doi.org/10.1016/j.jfranklin.2023.10.012
14. Wang, Y., Chen, H., & Sun, Z. (2021). Discrete-time adaptive neural network control for steer-by-wire systems with disturbance observer. *Engineering Applications of Artificial Intelligence, 97*, 104053. https://doi.org/10.1016/j.engappai.2020.104053
15. Wang, Y., Sun, Z., & Chen, H. (2021). Hybrid adaptive learning neural network control for steer-by-wire systems via sigmoid tracking differentiator and disturbance observer. *Engineering Applications of Artificial Intelligence, 102*, 104245. https://doi.org/10.1016/j.engappai.2021.104245
16. Yang, H., Wang, J., & Li, K. (2022). ESO-based lateral control for electrical vehicles with unmodeled tire dynamics on uneven road. *Mechanical Systems and Signal Processing, 170*, 108765. https://doi.org/10.1016/j.ymssp.2022.108765