2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2025)
October 19-25, 2025. Hangzhou, China

2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3315-4393-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IROS60139.2025.11246379

ExAMPC: the Data-Driven Explainable and Approximate NMPC with Physical Insights

Jean Pierre Allamaa[1,2] and Panagiotis Patrinos[2] and Tong Duy Son[1]

***Abstract*— Amidst the surge in the use of Artificial Intelligence (AI) for control purposes, classical and model-based control methods maintain their popularity due to their transparency and deterministic nature. However, advanced controllers like Nonlinear Model Predictive Control (NMPC), despite proven capabilities, face adoption challenges due to their computational complexity and unpredictable closed-loop performance in complex validation systems. This paper introduces ExAMPC, a methodology bridging classical control and explainable AI by augmenting the NMPC with data-driven insights to improve the trustworthiness and reveal the optimization solution and closed-loop performance's sensitivities to physical variables and system parameters. By employing a low-order spline embedding, we reduce the open-loop trajectory dimensionality by over 95%, and integrate it with SHAP and Symbolic Regression from eXplainable AI (XAI) for an approximate NMPC, enabling intuitive physical insights into the NMPC's optimization routine. The prediction accuracy of the approximate NMPC is enhanced through physics-inspired continuous-time constraints penalties, reducing the predicted continuous trajectory violations by 93%. ExAMPC also enables accurate forecasting of the NMPC's computational requirements with explainable insights on worst-case scenarios. Experimental validation on automated valet parking and autonomous racing with lap-time optimization, demonstrates the methodology's practical effectiveness for potential real-world applications.**

I. INTRODUCTION

Linear Model Predictive Control (MPC) stands out for its inherent explainability, allowing precise analysis of the instantaneous open-loop (OL) prediction and closed-loop (CL) system behavior. However, this clarity on stability and performance diminishes with complex systems, such as chaotic dynamics or those involving a plant model that is more complicated than the linear prediction model in the MPC. Moreover, MPC's prediction capabilities are limited by its prediction horizon, complicating the long-term CL performance analysis trough analytical approaches.

While data-driven control approaches such as Reinforcement Learning have gained significant traction in academia and robotics due to their minimal system knowledge requirement and their easiness to implement and use, safety-critical applications like autonomous driving demand safe, explainable, and transparent controllers. Although Nonlinear Model Predictive Control (NMPC) inherently offers these trustworthy qualities, it poses implementation and maintenance challenges for non-experts as physical insights are required, particularly when under-performance occurs due to model mismatch or real-time operation failures. Additionally, control engineers struggle to calibrate controllers for specific CL Key Performance Indicators (KPIs) that are not directly and analytically linked to the NMPC parameters but rather emerge from the interaction of the controller with the plant, environment and other unmodelled controllers. We propose leveraging operational CL data to: 1) approximate the NMPC using physics-inspired techniques, 2) expose the NMPC's decision making process within the specific environment conditions, and 3) predict and explain the complex system-level CL performance KPIs.

Training data-hungry learning-based controllers can be impractical and unsafe for systems where only operational data can be collected without disrupting the system. We propose using a small dataset to elucidate and predict system performance around current operating conditions. This approach accelerates the NMPC design and calibration for the specific operating applications, by integrating Machine Learning (ML)-based prediction and explainability with classical control methods like NMPC, complementing rather than replacing Artificial Intelligence (AI) and NMPC strengths. There exists several approaches allowing transparency and explainability of AI models, known as eXplainable Artificial Intelligence (XAI). Those include Symbolic Regression (SR) [1] that provides formulas linking outputs to input features with fast inference. Post-hoc XAI methods like Shapley Additive exPlanations (SHAP) [2] are also beneficial for interpreting feature contributions in model predictions, particularly for Regression Trees [3]. By calculating the features' marginal contributions affecting the prediction outcome of the ML module, SHAP renders the module's decision making process more transparent and interpretable.

Several research works propose the combination of NMPC with AI. Imitation Learning of the NMPC OL trajectories through B-spline based coefficients embedding to penalize linear continuous-time constraints violations has been discussed by [4]. The approximation of MPC by relying on physics-informed constraints has been explored by [5]. Furthermore, Transformed-based MPC works for generating OL trajectories have been proposed in [6], [7] or in [8] where the predicted trajectories are used to warm-start an NMPC to accelerate its online computation. Additionally, SHAP has been applied to MPC for model interpretation

[1]Siemens Digital Industries Software, 3001, Leuven, Belgium
Email: {jean.pierre.allamaa, son.tong}@siemens.com
[2]Dept. Electr. Eng. (ESAT) - STADIUS research group, KU Leuven, 3001 Leuven, Belgium
Email: panos.patrinos@esat.kuleuven.be

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement ELO-X No 953348, and Horizon 2022 research and innovation programme under grant agreement No 101120276 (SoliDAIR)

Supplementary video at: https://youtu.be/qGgeQaaEDXc

Authorized licensed use limited to: KU Leuven University Library. Downloaded on April 22,2026 at 09:49:21 UTC from IEEE Xplore. Restrictions apply.

in [9]. Finally, a review in [10] discusses the use of neural networks in MPC for optimization efficiency. Current NMPC approximation approaches face three key challenges: 1) scalability issues and high dimensionality output demands in discrete-time sequence predictions; 2) inadequate and localized explainability of the discrete sequence element's with respect to the trend of the sequence; and 3) inability to enforce Continuous-Time Constraint Penalties (CTCP), rendering the interpolation between two discrete points of the sequence in a possibility of constraints violation.

We present the Explainable and Approximate NMPC (ExAMPC) with four main contributions: 1) proactive forecasting and monitoring of NMPC's CL performance within interconnected systems, for non-experts, 2) physics-inspired NMPC approximation using a low-order encoding via Legendre-Splines embedding, providing smooth predictions with physical insights, 3) enhanced explainability for NMPC performance and OL predictions through coupling with XAI tools, and 4) experimental validation in autonomous driving and racing demonstrators.

The paper is organized as follows: in Sec. II we briefly present related work on data-driven and approximate NMPC. In Sec. III we provide a background on the continuous-time optimal control problem and the employed ML regression methods in this work. In Sec. IV we introduce the low-order embedding of time-series and the physical-inspired regression model that builds the approximate NMPC for predictions with minimal continuous-time constraints violation and provide an explainability study on the OL prediction of the approximate NMPC. We follow with Sec. V where we introduce the explainable CL performance monitor for the NMPC and demonstrate it on an autonomous driving and racing applications before concluding in Sec. VI.

II. Related work

The importance of explainable data-driven control, as discussed in [11], lies in its ability to enhance the transparency of the decision-making process in complex systems. Traditional approaches like Explicit MPC [12], enabled the real-time execution of the linear MPC by precomputing control laws through Multi Parametric Programming and storing them in look-up tables. However, this method remains memory-inefficient and unsuitable for nonlinear MPC. While Explicit MPC provides a framework for understanding system behavior in an OL fashion and was extended to provide complexity certification for a particular set of Quadratic Programming (QP) solver as in [13], it lacks flexibility to systems with unknown solvers, and the adaptability to forecast performance measures such as the NMPC's computation time. For embedded applications, where the controller's computation time is critical, [14] propose a method to determine the worst-case execution time of a particular class of linear MPC formulations, allowing an online monitoring and prediction of this KPI. We aim at extending the previous approaches into a method allowing nonlinear constraints and dynamics handling, as well as complicated system-level CL KPIs emerging from the interaction between the controller and the different system components as in Figure 1.

A. On the use of AI for approximate NMPC

We focus on learning two types of output as in Figure 1:

1) the optimized NMPC OL trajectory for the states' and control actions' predicted evolution, which is the OCP (1)'s solution at every instance t, for given state estimate $\mathbf{x}_0(t)$: $\{\mathbf{x}_0(t), \eta, p(t)\} \rightarrow \{\mathbf{x}_{(\cdot)}(t), \mathbf{u}_{(\cdot)}(t)\}$. The autonomous CL system evolves under uncertainties ξ and scenario parameters $p(t)$ (such as target states, boundary conditions, etc..).
2) the instantaneous measured CL KPIs $K_i(t)$ for $i = 1, \ldots, n_{KPI}$, emerging from system evolution and not necessarily analytically linked to MPC parameters.

Existing approaches like [6] that approximate the NMPC solution to predict discrete sequences $\mathbf{x}_{(\cdot)}(t) = \{\mathbf{x}(t), \mathbf{x}(t+T_s), \ldots, \mathbf{x}(t+NT_s)\}$ over the NMPC horizon length N with a step size T_s, face several limitations:

- Data inefficiency: poor scalability with large horizon length and different step sizes when using discrete methods such as direct multiple shooting, despite transformer-based technologies [6], [8] that can handle longer trajectories, but require extensive data.
- High output dimensionality for discrete sequences.
- Lack of CTCP, potentially compromising safety.
- Limited output explainability: individual sequence elements provide minimal insights into trajectory trends, dynamics, and sensitivity to input features and parameters, making them non-intuitive.

The authors of [4] address the first three limitation by embedding the infinite dimensional continuous sequence $\{\mathbf{x}_{(\cdot)}(t), \mathbf{u}_{(\cdot)}(t)\}$ into a finite set of B-spline coefficients, which have a convex hull property. However, B-splines are complex to construct, highly sensitive to the chosen knot sequence, and the coefficients offer limited physical insights on the trajectory trend and dynamics. Moreover, modeling time-series with high accuracy using B-splines requires a rather dense knot sequence, where poorly chosen knots can lead to ill-conditioning in the fitting problem.

III. Preliminaries

In this section we detail the Legendre-Spline encoding based on orthogonal collocation methods, we discuss the three ML regression methods used in this work, we set the learning objectives for approximate NMPC, and present the data generation scenarios for autonomous driving and racing.

A. Continuous-time Optimal Control Problems

An Optimal Control Problem (OCP) is initially posed in continuous time and seeks to optimize a cost function $J(\mathbf{x}, \mathbf{u})$ while satisfying a set constraints as in the nonlinear

Authorized licensed use limited to: KU Leuven University Library. Downloaded on April 22,2026 at 09:49:21 UTC from IEEE Xplore. Restrictions apply.

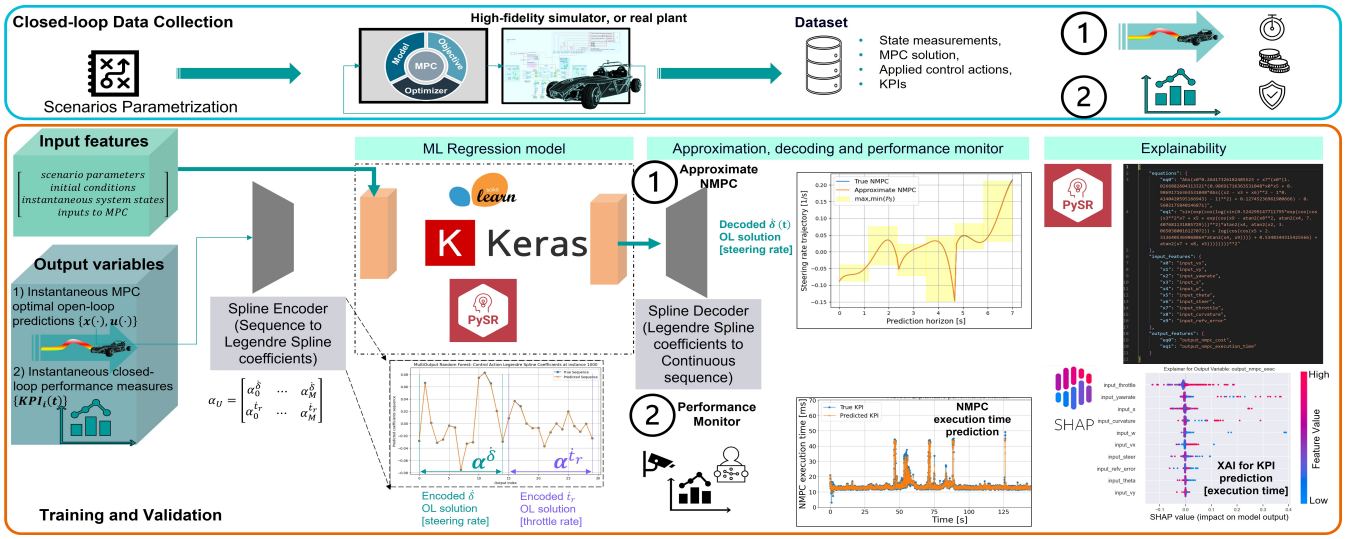

Fig. 1. ExAMPC framework: employing operational closed-loop data to approximate the NMPC and a performance monitor then explain them using XAI

continuous Bolza problem [15]:

$$\begin{cases} \underset{\mathbf{x}(\cdot),\mathbf{u}(\cdot)}{\textbf{minimize}}\ J(\mathbf{x},\mathbf{u}) = \Phi(\mathbf{x}(t_f)) + \int_{t_0}^{t_f} L(\mathbf{x}(t),\mathbf{u}(t))dt \\ \quad \text{subject to } \dot{\mathbf{x}}(t) = f(\mathbf{x}(t),\mathbf{u}(t)), \\ \qquad\qquad g(\mathbf{x}(t),\mathbf{u}(t)) \leq 0, \\ \qquad\qquad g_f(\mathbf{x}(t_f)) \leq 0, \\ \qquad\qquad \mathbf{x}(t_0) = \bar{\mathbf{x}}_0. \end{cases} \tag{1}$$

where $t \in \mathbb{R}$ denotes time, $\mathbf{x} \in \mathbb{R}^{N_x}$ is a state of the system, $\mathbf{u} \in \mathbb{R}^{N_u}$ is a vector of control inputs. The function $\Phi : \mathbb{R}^{N_x} \to \mathbb{R}$ is the terminal cost function and $L : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \to \mathbb{R}$ is the running or stage cost. The continuous-time system dynamics are given by the function $f : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \to \mathbb{R}^{N_x}$. The function $g : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \to \mathbb{R}^{N_g}$ is the path linear and nonlinear constraints function, and $g_f : \mathbb{R}^{N_x} \to \mathbb{R}^{N_{g_f}}$ is the terminal constraints function. Finally, $\bar{\mathbf{x}}_0$ is an input parameter to the OCP setting the initial states condition $\mathbf{x}(t_0)$. The OCP in (1) is solved in a receding horizon fashion resulting in an NMPC CL control framework: at every control iteration of step size T_s, new state measurements or estimations are fed into the OCP to solve for (1) over a horizon $T_H = t_f - t_0$ and the first control action $\mathbf{u}(t_0) \to \mathbf{u}(t_0 + T_s)$ is applied.

We define the orthogonal collocation scheme based on the Truncated Legendre Series (TLS) of degree M as in (2) to approximate the solutions $\mathbf{x}(\tau)$ and $\mathbf{u}(\tau)$ of the continuous time OCP in a compact representation through the coefficients $\boldsymbol{\alpha}$ rather than a discrete set of points, over the normalized time horizon $\tau \in [-1, 1]$ of $t \in [t_0, t_f]$:

$$\mathbf{x}(\tau) = \sum_{j=0}^{M} \alpha_j^x \mathcal{L}_j(\tau) = (\boldsymbol{\alpha}^x)^\top \mathbf{L}_M v(\tau). \tag{2}$$

The matrix $\mathbf{L}_M \in \mathbb{R}^{(M+1)\times(M+1)}$ is formed by the coefficients of $\mathcal{L}_j$ with respect to τ. In particular, the spanning basis are Legendre polynomials $\mathcal{L}_j(\tau)$, which have the fundamental orthogonality property. This property renders the basis terms independent from each other and less sensitive to perturbation. Moreover, the TLS is parametrized by the coefficients $\boldsymbol{\alpha}^x = \begin{bmatrix} \alpha_0^x & \cdots & \alpha_M^x \end{bmatrix}^\top \in \mathbb{R}^{(M+1)\times N_x}$, $\boldsymbol{\alpha}^u \in \mathbb{R}^{(M+1)\times N_u}$, and $v(\tau) = \begin{bmatrix} 1 & \tau & \tau^2 & \cdots & \tau^M \end{bmatrix}^\top$ is a vector with a geometric progression of the normalized time instance $\tau = (2t/t_f - 1)$ with $t_0 = 0$. Equations (2) and (3) also apply to control trajectories $\mathbf{u}(\tau)$ with $\boldsymbol{\alpha}^u$.

For non-smooth systems and solutions that cannot be fit with a single TLS, the normalized NMPC horizon $[-1, 1]$ can be divided into smaller finite elements to create a piecewise polynomial, namely a Legendre-Spline. On each element, every state and input is parametrized by a TLS, the coefficients of which serve as optimization variables $\boldsymbol{\alpha}$:

$$\mathbf{x}(\tau) = \begin{cases} \sum_{j=0}^{M} \alpha_{1,j}^x \mathcal{L}_j(\tau^*), & \tau_1 = -1 \leq \tau < \tau_2 \\ \vdots & \\ \sum_{j=0}^{M} \alpha_{N_S,j}^x \mathcal{L}_j(\tau^*), & \tau_{N_S} \leq \tau \leq 1 \end{cases} \tag{3}$$

where τ^* converts the section limits back into $[-1, 1]$. The normalized interval is crucial as the Legendre polynomials' definition and orthogonality properties are derived over it.

In the generic case we refer to the Legendre-Spline coefficients of a state x as $\alpha_{i,j}^x$ where $i \in [1, N_S]$ is the element number and $j \in [0, M]$ is the coefficient of order j at this element. A TLS is by design composed out of one single element, and is described by the coefficients $\alpha_{0,j}^x = \alpha_j^x$. The vector containing all the coefficients of section i is denoted by $\boldsymbol{\alpha}_i^x$, and the concatenation of the coefficients over all the sections that represent the Legendre-Spline are denoted by $\boldsymbol{\alpha}^x$. Furthermore, in [16, Theorem 1], it is proven that the continuous-time trajectory of the state, control trajectories and linear/nonlinear constraints over them can be bounded in a finite approach through the regional convex hulls $\mathcal{P}^k$ of the Legendre-Spline through constant matrices C_M^k. For every TLS, the time interval $[-1, 1]$ is divided into K regions

Authorized licensed use limited to: KU Leuven University Library. Downloaded on April 22,2026 at 09:49:21 UTC from IEEE Xplore. Restrictions apply.

that are not necessarily equidistant, for less conservatism on the TLS's extrema approximation:

$$\min\{g(\mathcal{P}_x^k, \mathcal{P}_u^k)\} - \epsilon \leq g(\mathbf{x}, \mathbf{u}) \leq \max\{g(\mathcal{P}_x^k, \mathcal{P}_u^k)\} + \epsilon, \tag{4a}$$

$$\text{where } \mathcal{P}_x^k = \mathcal{C}_M^k \alpha^x,\ \mathcal{P}_u^k = \mathcal{C}_M^k \alpha^u. \tag{4b}$$

Within the context of approximate NMPC, we propose the use of RESAFE/COL [16] to overcome the B-spline limitations presented in Sec. II-A (data efficiency, dimensionality, lack of CTCP and explainability) by using spectral collocation with orthogonal basis polynomials to achieve high solution accuracy and impose continuous-time nonlinear constraints by relying on a linear mapping between the coefficients embedding Legendre-Spline and its extrema as in (4). This approach provides interpretable coefficients sequence revealing physical evolution information on the trajectory through the zero-order bias term α_0 or the ith derivative terms. Moreover, it benefits from a better numerical conditioning and a low-order fitting: as the order M increases, the high order-terms naturally vanish to zero if they do not increase the accuracy, as the coefficients α_j decay faster than any polynomial in j [17]. We refer to this method as a low-order embedding.

B. MultiOutput Regression Trees, Neural Networks and Symbolic Regression

We aim to learn the sequence of Legendre-Spline coefficients that embed the continuous-time trajectories. The coefficients of each state trajectory are independent, but might correlate to each other between the different states (e.g. a state which is the derivative of another). For that, we employ three methods: Recurrent Neural Network (RNN) with Long-Short Term Memory (LSTM) layers, MultiOutput Regressor Random Forest (MORRF) and SR. Although the focus of this paper is not about the ML algorithm selection, we give a brief overview about the employed methods.

We implement and train RNN using the Keras library. RNN is chosen due to its capability to handle sequential data and capture the dependencies between the predicted sequence's elements. This is helpful as we aim to predict the NMPC's OL trajectories, embedded as coefficients as in Figure 1. The OL trajectories are based on a physical system with coupling between states and controls, thus the choice of RNN. The architecture consists of two layers (256 and 128 neurons respectively), followed by a reshaping, then an LSTM layer (64 units) and the output layer.

The MORRF implementation uses *scikit-learn* [18]. Random Forest is an ensemble learning method that relies on the output of multiple decision trees to produce a more accurate prediction. MORRF is suitable for small datasets that are exemplary of the current operational data and provides robust predictions with minimal hyperparameter tuning. The training of MORRF is relatively fast and allows explainability and an almost online training then inference, which is important for engineers to rapidly understand their operating NMPC. We use 20 estimators per Random Forest Regressor.

While RNN and MORRF are non-transparent by nature, XAI tools like [2] provide explainability to the trained algorithm, making the decision-making process of the regression model more intuitive for non-experts, on the basis that the model provides high validation and testing accuracy. This allows understanding the physical phenomena and correlations between features and outputs, which is crucial for explaining model-based controllers in CL through XAI.

Finally, SR is another regression model that is explainable by design. SR offers interpretable analytical equations linking outputs to inputs, providing physical insight to the control engineer through explicit mathematical formulas. Moreover, the approach learns a structure of the underlying physics if the used basis functions capture the pattern well. We use *PySR* [1] with the binary operators $\{+, -, \times, \text{atan2}(y, x)\}$ and unary operators $\{\cos(x), \sin(x), \exp(x), |x|, x^2\}$, with maximum 200 iterations and 500 cycles per iteration. We apply SR specifically for KPIs prediction and monitoring, where the output dimensionality is manageable.

C. Data generation: autonomous driving and racing

We demonstrate our work on autonomous driving and racing control applications. The NMPC is implemented in CasADi [19] using an SQP method with OSQP as the underlying QP solver. The OCP is transcribed into a nonlinear programming problem using RESAFE/COL [16]. The NMPC uses a fused kinematic-dynamics bicycle model with a Pacejka tire formulation. The verification uses 15 Degrees-of-Freedom high-fidelity Digital Twin (DT) of the vehicles in Simcenter Amesim. Two demonstration setups are created:

1) Autonomous Valet Parking (AVP) at speeds up to 20kph using an electric 2 seater prototype vehicle, a SimRod. The NMPC handles velocity tracking, path following and parking positioning with collision avoidance capabilities using Control Barrier Functions as in [16]. The dataset comprises 200 scenarios of 60 seconds each, featuring randomized scenario parameters for start position, parking locations, and speed.
2) Autonomous racing demonstrator at speeds reaching 330kph. Here the virtual NMPC driver focuses on path tracking and lap-time optimization by maximizing the evolution along the path within a prediction horizon. The employed vehicle is a one-seater racing vehicle. A single lap around a racing track for 2 minutes and 40 seconds, sampled at 20ms, proves sufficient to demonstrate the method's effectiveness in terms of approximation and explainability with small data requirements for cases with a small operating domain.

Both setups integrate the NMPC as a standalone C-code library for co-simulation with the Digital Twins (DTs) with Simcenter Amesim for vehicle dynamics and Simcenter Prescan for environment simulation and sensor modeling for obstacle detection of crossing pedestrians and road users as visualized in Figure 7. The NMPC serves as the lowest-level control, executing trajectories from a high-level planner through steering, brake and throttle commands. The collected data is divided into training (64%), validation (16%) and

Authorized licensed use limited to: KU Leuven University Library. Downloaded on April 22,2026 at 09:49:21 UTC from IEEE Xplore. Restrictions apply.

testing (20%). Finally, the data is normalized per feature and per output to $[-1, 1]$.

IV. AI AS AN EXPLAINABLE NMPC APPROXIMATION

In this section we present and demonstrate the approximation of the NMPC's OL solutions using ML regression, by relying on a physics-informed, data-efficient and low-order method. Moreover, we discuss the use of XAI techniques to gather physical insights on the optimization routine and on the trend of the OL trajectories. Finally we demonstrate the use of ExAMPC in the autonomous driving and racing scenarios and present the respective results.

A. RESAFE/COL for physics-informed approximate NMPC

The NMPC solves for continuous-time trajectories that are embedded in form of Legendre-Splines' coefficients (3). As explained in Sec. II-A, the use of a Legendre-Spline with orthogonal basis offers two key advantages: 1) naturally regularized low-order embedding through decoupled and independent coefficients, and 2) physical constraint enforcement via a linear coefficient mapping, enabling CTCP without discrete sampling of the time-series trajectory.

As illustrated in Figure 1, we collect CL data of the NMPC controlling a high-fidelity DT to train ML regression models to imitate the NMPC's OL solution. Using similar input parameters being fed into the NMPC, the trained regression model would mimic the optimization process. Unlike standard imitation learning approaches that learn only the NMPC's policy or first control action $\mathbf{u}(t = t_0)$, we propose to learn the complete time-series of this policy evolution $\mathbf{u}_{(.)}(t)$, and that of the states $\mathbf{x}_{(.)}(t)$. This approach captures the optimization framework linking the predicted states and control actions. It also enables an effective warm-starting strategy for the NMPC which is known to have benefits on the numerical efficiency and helps speeding-up the computation in methods such as SQP. A baseline approach to this OL trajectory regression predicts the sequence of the sampled discrete trajectory points, or the embedding coefficients using a Mean Squared Error (MSE) loss:

$$\mathcal{L} = \mathcal{L}_{MSE} = \sum_{i=1}^{N_{batch}} \|(\bar{\boldsymbol{\alpha}}_i - \boldsymbol{\alpha}_i)\|^2 / N_{batch}, \tag{5}$$

where $\boldsymbol{\alpha}_i \in \mathbb{R}^{(1 \times N_{predict})}$ contains the sequence of coefficients from all the states and control respectively at instance i with $N_{predict} = (M + 1) \times (N_x + N_u)$ elements, and we train over batches of size N_{batch}, and $\bar{\boldsymbol{\alpha}}_i$ is the predicted coefficients sequence. We enhance this loss function with a physics-informed loss using the convex hulls $\mathcal{P}^k$ from (4) to penalize the continuous-time constraint violations:

$$\mathcal{L} = \mathcal{L}_{MSE} + \gamma \mathcal{L}_{RESAFE}, \tag{6a}$$

$$\mathcal{L}_{RESAFE} = \sum_{i=1}^{N_{bacth}} \sum_{k=1}^{K} \max(0, g(\mathcal{P}_x^k, \mathcal{P}_u^k) - \epsilon_{tol}), \tag{6b}$$

where ϵ_{tol} defines the violation threshold tolerance. Note that linear state and control constraints of the form $\underline{\mathbf{x}} \leq \mathbf{x}(t) \leq \overline{\mathbf{x}}$ are expressed in the generic form of $g(\mathbf{x}, \mathbf{u}) \leq 0$ for conciseness. The extrema elements of the convex hulls $\mathcal{P}_u$ and $\mathcal{P}_{v_x}$ are shown in the shaded yellow areas in Figures 1 and 3, over the decoded control actions and velocity trajectories.

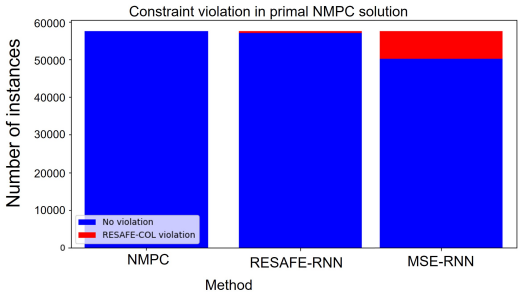

Fig. 2. Physics-inspired continuous-time constraints penalty with RESAFE/COL's convex hull in comparison with a baseline method

B. Multistep prediction using coefficients: a data efficient and explainable approach

The proposed Legendre-Spline embedding addresses the challenges stated in Sec. II-A by encoding physical information through the coefficients: the zero order coefficient α_0 for the trajectory bias or mean term, the first order coefficient α_1 for information about the rate of change, the second-order coefficient α_2 about acceleration characteristics and the higher-order terms on additional dynamic features of the trajectory. This representation enables engineers to interpret and shape the OCP's OL behavior through physically meaningful parameters that can be explained using SHAP. Therefore, by combining (3) with (6) and XAI tools, we allow multistep sequence prediction in one shot while requiring little data due to the low dimensionality of the prediction, and while having physical insights as shown in the framework of Figure 1.

C. ExAMPC as a warm-starter for NMPC

We train two RNNs to approximate the NMPC for the AVP use case: one using the baseline with MSE (MSE-RNN) and another with the RESAFE/COL-type of loss as in (6) for CTCP with $\gamma = 1$ (RESAFE-RNN). As shown in Figure 2, the approximate NMPC as RESAFE-RNN achieves 556 continuous-time constraint violations out of 57632 testing instances. The total loss is equal to 2.05e-04 divided into a MSE of 2.0e-04 and CTCP of 4.9e-06. In contrast, the baseline MSE-RNN using only coefficients learning results in 8113 violations out of 57632 instances, with a MSE on the coefficients of 1.7e-04 but a CTCP of 8.5e-03. Overall, the RESAFE/COL approach with RESAFE-RNN demonstrates significant improvements equivalent to a 93% reduction in the number of continuous-time constraint violations using the approximate NMPC. In terms of the magnitude of those violations, a reduction of 99.94% is calculated. As the approximate NMPC with RESAFE-RNN 1) accurately predicts the solution of the NMPC, and 2) exhibits little constraints violations, it could be used as an initial feasible guess for the online NMPC.

Authorized licensed use limited to: KU Leuven University Library. Downloaded on April 22,2026 at 09:49:21 UTC from IEEE Xplore. Restrictions apply.

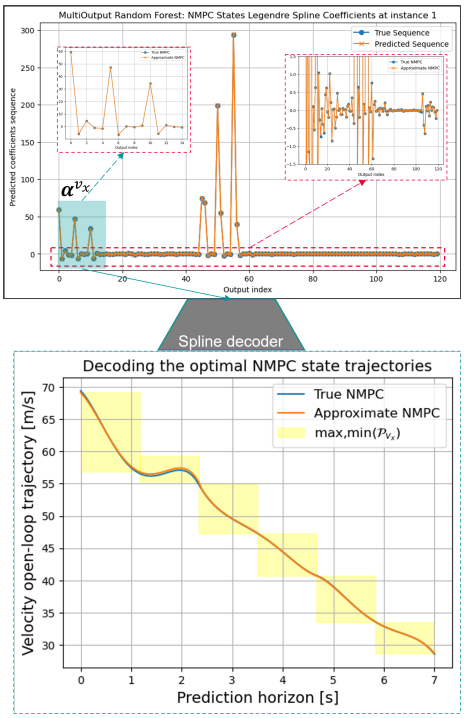

Fig. 3. Approximate NMPC: Continuous states trajectory embedding through Legendre-Spline coefficients, prediction using Random Forest Regressor, and spline decoding with the extrema of the regional convex hulls

D. Results and explainability for autonomous racing

For the autonomous racing use case, the NMPC solves for 8 states and 2 control actions OL trajectories over $T_H = 7$ seconds. The Legendre-Spline (c.f. (3), Figure 3) over the 7 seconds has $N_S = 3$ sections, with an order $M = 4$. That is the continuous-time trajectory of every state and control action is represented by a total of $N_{predict} = 15$ coefficients. In contrast, a traditional discrete sequence prediction as in [7] using a sampling time $T_s = 20$ms, would require 350 points to represent the same trajectory. The proposed method thus achieves 95.71% reduction in dimensionality while maintaining the trajectory accuracy.

Note that the proposed method remains compatible with discrete-time NMPC solutions such as direct multiple shooting through a least-squares fitting into Legendre-Splines. The orthogonal basis properties ensure that coefficients remain independent, leading to localized error effects. For instance, when prediction errors occur in coefficient $\alpha_{1,0}^{v_x}$, they only affect the offset of the first rolled-out TLS, from time $t = 0s$ to $t = 2.33s$, while maintaining trajectory smoothness, as illustrated in Figure 3. This property stems from the orthogonal basis, where errors in j^{th} coefficient only impact the corresponding j^{th} derivative locally. The SHAP analysis reveals key insights into the OL solution particularly for the first Legendre-Spline coefficients as illustrated in Figure 4, for the steering angle command δ and normalized acceleration or throttle command $t_r \in [-1, 1]$ that combines the throttle and brake commands into one variable. SHAP values on the x-axis show each feature's impact on the model's prediction relative to the baseline (average). Positive values (right) push predictions higher, and negative values (left) lower them, with a magnitude equal to the SHAP value. Influential and important features are ordered top to bottom. Each dot represents a data point, colored, by the feature value (red for high, blue for low). As the NMPC solves for the steering rate $\dot{\delta}$ and normalized throttle rate $\dot{t}_r$ as control actions $\mathbf{u}(t)$, the current steering angle $input_steer = \delta$ is fed as a parameter to the NMPC and approximate NMPC in the initial state estimation. For the steering trajectory, $\alpha_{1,0}^{\delta}$ shows the strongest explainability with the current steering measurement as in Figures 4 (a). This is expected as the first coefficient of the sequence holds the zero-order information about the time-series which are mainly influenced by the bias term in the spline, and the OCP (1) solves for the Legendre-Spline of the steering trajectory to be equal to $input_steer$ at $t = 0$. Notable patterns include increased steering under braking conditions (low feature value of $input_throttle$, tending to -1). Moreover, the velocity $input_vx$ has major impact on the output of the approximate NMPC for $\alpha_{1,0}^{\delta}$. The high values of steering $\alpha_{1,0}^{\delta}$ (both positive and negative) occur at lower velocities. This aligns with the expected behavior, as the NMPC minimizes steering at high-speeds of over 300kph to maintain path stability. Due to orthogonality, the steering rate (Figures 4 (b)) is mainly represented by the second coefficient of the sequence, $\alpha_{1,1}^{\delta}$ which carries information on the first-order derivative with respect to time. It correlates strongly with the yaw rate, showing a compensation behavior: negative yaw rates (blue or low feature value of $input_yawrate$) trigger positive steering rates and vice versa. That is, when the vehicle is rotating counterclockwise, the NMPC reacts by steering clockwise and vice versa. Path deviations ($input_w$) also influence steering rates, with leftward deviations (red or high feature value) triggering a clockwise (negative) corrections and steering rates. A deceleration maneuver (negative $input_throttle$ in blue) causes higher steering rates, mainly as the vehicle attacks corners at reduced speeds. The predominance of left-hand corners in the racing track is reflected in the asymmetric distribution of the SHAP values in steering and steering rates towards the positive right-hand side of the plot. The orthogonality of Legendre-Spline coefficients enables this clear separation between zero-order behavior and dynamic responses, providing interpretable insights into the NMPC's decision-making process.

V. AI AS PERFORMANCE MONITOR FOR THE NMPC

After demonstrating the approximate NMPC, we focus in this section on the use of explainable performance monitors for system level KPIs monitoring. The importance of employing XAI tools such as SHAP and SR are also highlighted as we extract important insights on the NMPC's CL operational capability by relying on a small dataset.

A. Explainable AI for performance prediction

An Explainable NMPC aids users in visually understanding key parameters influencing the decision-making process,

Authorized licensed use limited to: KU Leuven University Library. Downloaded on April 22,2026 at 09:49:21 UTC from IEEE Xplore. Restrictions apply.

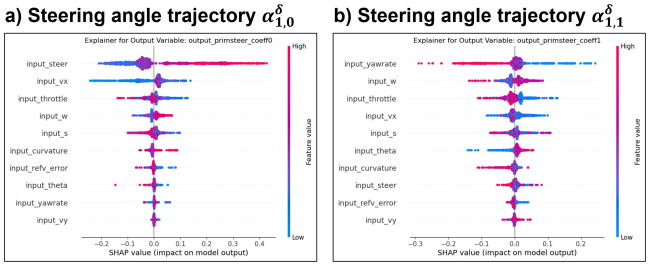

Fig. 4. Insights into the control action trends through a SHAP explainability of the approximate NMPC's Legendre-Spline coefficients

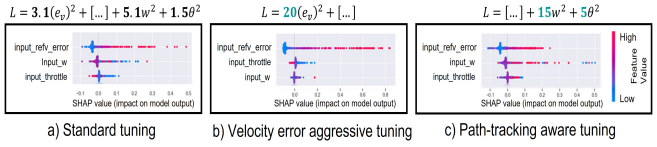

Fig. 5. Performance prediction on the cost function KPI: reverse engineering different controller tuning in an AVP use case

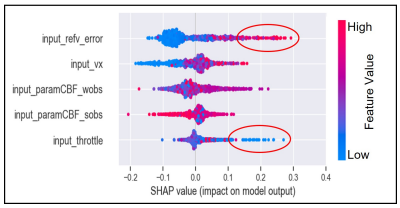

Fig. 6. Explainable performance monitor and prediction for the NMPC computation time in an AVP use case: isolating the dominant features

enabling real-time monitoring of the performance and suggesting when fallback controllers are necessary. As shown in Figure 1, several instantaneous CL KPIs can be monitored and forecast before occurring. We focus on two KPIs: the NMPC optimal cost-function $K_1(t)$ indicating optimization feasibility and system energy, and the NMPC execution time $K_2(t)$ reflecting real-time computation capabilities.

B. Symbolic performance monitor

We train MORRF and SR for performance prediction on K_1, K_2. In general, the performance prediction using MORRF achieves superior accuracy and faster training (MSE: 1.8e-04, quasi-instantaneous) compared to SR (MSE: 2.3e-03, couple of minutes). This indicates that MORRF is able to capture better the complex coupling between those KPIs and the input features. However, SR provides explicit models linking K_1, K_2 to the input features by optimizing for both the structure and parameters of the model as shown in the explainability block of Figure 1. This can enable cluster creation, and output reverse engineering. That is, if a desired computation time is to be met, an analytical operational domain of the input features can be computed by using the equations from *PySR* (Figure 1). Furthermore, one could employ this approach to reverse engineer the designed cost function of an operating blackbox NMPC by relying on the measured features to imitate the NMPC's internal optimization.

C. Results and analysis

Analysis of three NMPC tuning in the AVP demonstrator reveal interesting patterns for reverse engineering the NMPC cost function. In Figure 5, the first (baseline) tuning indicates high sensitivity to velocity tracking error $input_refv_error$, and asymmetric responses to the path deviations $input_w$ with deviations to the right of the path (low-feature values) impacting the cost function more than the left-hand ones (high-feature values). Increasing the velocity tracking error weight e_v from 3.1 to 20 amplifies the velocity error component's influence on the monitor prediction in the second tuning, confirming the method's ability to capture the internal NMPC optimization priorities. Increasing the lateral tracking error weights on path deviation w and heading deviation θ in the third tuning, produces the expected quadratic cost behavior with $input_w$ becoming a dominant feature in the explainable monitor, where both high and low-feature values increase the predicted NMPC cost or the consumed energy.

Explainability of the computation time prediction for the AVP demonstrator reveals four dominant features as in Figure 6: velocity tracking error ($input_refv_error$), path deviation ($input_w$), obstacle position and heading in the frame tangent to the path ($input_paramCBF_*obs$), and the normalized throttle command ($input_throttle$). Large velocity tracking errors (red) significantly increase computation time, while deceleration and braking commands (low feature value, blue) demand more computational resources than acceleration and throttling commands (high feature value, red), suggesting potential numerical challenges of the NMPC to solve at low speeds as the NMPC is more sensitive to braking ($t_r = -1$) than accelerating ($t_r = +1$).

Furthermore, we run the performance monitor for the autonomous racing demonstrator and use SHAP to understand the edge cases, as in Figure 7. MORRF effectively captures both the NMPC's optimal CL cost function value and computation time KPIs. While the monitor accurately predicts significant cost function fluctuation as for e.g. around $t =$ 90s, the absolute computation time values are hardware-dependent and might be less generalizable. Yet, MORRF's capability to handle outliers, allows it to detect sudden computational peaks proving its crucial importance for the system safety monitoring. Those peaks, although rather limited, are important for edge cases studies and understanding NMPC's handling near the limits. A critical instance occurs near $t = 45$s, where the vehicle exits the apex of a tight corner at 60kph before transitioning to an acceleration phase towards 330kph, as predicted by the NMPC over the next 7s. SHAP analysis of this instance reveals that yaw rate $input_yawrate$ and path heading deviation $input_theta$ are the primary contributors to the well predicted high computation time. This insight provides control engineers with three actionable options: controller redesign to better handle high yaw rate scenarios, investigation of the bicycle model's

Authorized licensed use limited to: KU Leuven University Library. Downloaded on April 22,2026 at 09:49:21 UTC from IEEE Xplore. Restrictions apply.

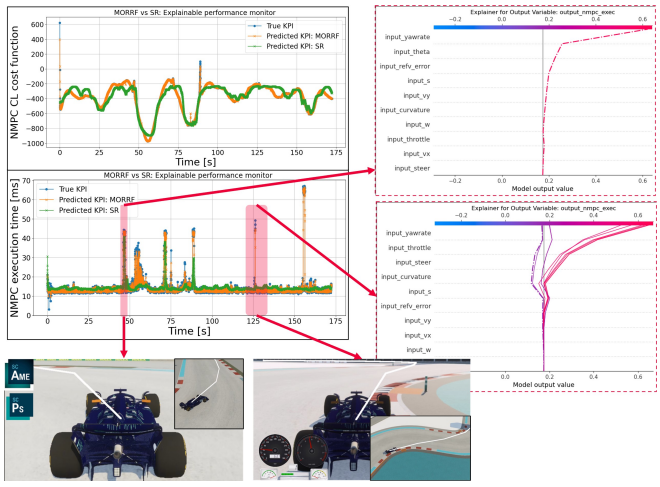

Fig. 7. Performance prediction and explainability: NMPC closed-loop optimal cost and execution time edge cases for the racing use case

numerical behavior during the optimization under large yaw rate, or implementation of a fallback controller during such challenging cornering scenarios. Further analysis at the high-speed chicane (near $t = 125$s) reveal more insights on the computation time patterns. While some instances maintain normal execution times around the mean, critical cases emerge from the combined effects of large normalized throttle (i.e. acceleration) and yaw rate. As shown in the DT snapshot of Figure 7, the NMPC commands a 50% full throttle, and this occurs simultaneously under large rotation or yaw rates, as depicted in the SHAP plot. This causes a sudden surge in computation time. The sudden activation of yaw rate and non-slip constraints in this dynamic corner scenario alters the optimal solution. This makes the warm-started primal and dual variables from the previous iteration a less effective initial guess for the new problem instance, thereby increasing the number of OSQP iterations. Finally, the complete execution time monitor's explainability plot is presented in Figure 1.

VI. Conclusion

This work introduces ExAMPC, an explainable and approximate NMPC and monitor framework for NMPC-controlled autonomous systems operating under model mismatch and environmental uncertainties. One aim of the work is to assist users without deep technical expertise to easily comprehend and operate an NMPC and its behavior. We propose the embedding of time-series through a Legendre-Spline encoding for dimensionality reduction, to approximate and explain the open-loop primal solutions of the NMPC as state and control trajectories, through a physics-inspired loss, enhancing the continuous-time safety satisfaction by 93%, and achieving close to no constraints violation. Additionally, by combining SHAP and Symbolic Regression, ExAMPC provides an explainable performance monitor to uncover the physical insights affecting performance indicators such as closed-loop cost and predicted energy value and the impact of measurements such as vehicle yaw rate and tracking errors on the computation time. Future work could leverage these explainability results for targeted data generation in edge cases using DTs, and integrate SR-derived analytical KPI models directly into the NMPC optimization.

References

[1] M. Cranmer, "Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl," May 2023.

[2] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017.

[3] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable AI for trees," *Nature Machine Intelligence*, vol. 2, no. 1, pp. 2522–5839, 2020.

[4] F. S. Acerbo, H. Van der Auweraer, and T. Duy Son, "Safe and computational efficient imitation learning for autonomous vehicle driving," in *2020 American Control Conference (ACC)*, 2020, pp. 647–652.

[5] D. Xu, R. Aerts, P. Karamanakos, and M. Lazar, "Constraints-Informed Neural-Laguerre Approximation of Nonlinear MPC with Application in Power Electronics," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, 2024, pp. 7466–7471.

[6] D. Celestini, D. Gammelli, T. Guffanti, S. D'Amico, E. Capello, and M. Pavone, "Transformer-Based Model Predictive Control: Trajectory Optimization via Sequence Modeling," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 9820–9827, 2024.

[7] J. Park, M. R. Babaei, S. A. Munoz, A. N. Venkat, and J. D. Hedengren, "Simultaneous multistep transformer architecture for model predictive control," *Computers & Chemical Engineering*, vol. 178, p. 108396, 2023.

[8] V. Zinage, A. Khalil, and E. Bakolas, "TransformerMPC: Accelerating Model Predictive Control via Transformers," 2024.

[9] P. Henkel, T. Kasperski, P. Stoffel, and D. Müller, "Interpretable data-driven model predictive control of building energy systems using SHAP," in *Proceedings of the 6th Annual Learning for Dynamics &; Control Conference*, ser. Proceedings of Machine Learning Research, A. Abate, M. Cannon, K. Margellos, and A. Papachristodoulou, Eds., vol. 242. PMLR, 15–17 Jul 2024, pp. 222–234.

[10] C. Gonzalez, H. Asadi, L. Kooijman, and C. P. Lim, "Neural Networks for Fast Optimisation in Model Predictive Control: A Review," 2024.

[11] G. Riva and S. Formentin, "Toward eXplainable Data-Driven Control (XDDC): The Property-Preserving Framework," *IEEE Control Systems Letters*, vol. 8, pp. 478–483, 2024.

[12] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[13] G. Cimini and A. Bemporad, "Exact Complexity Certification of Active-Set Methods for Quadratic Programming," *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6094–6109, 2017.

[14] D. Arnström, D. Broman, and D. Axehill, "Exact worst-case execution-time analysis for implicit model predictive control," *IEEE Transactions on Automatic Control*, vol. 69, no. 10, pp. 7190–7196, 2024.

[15] G. T. Huntington, "Advancement and analysis of Gauss pseudospectral transcription for optimal control problems," Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[16] J. P. Allamaa, P. Patrinos, T. Ohtsuka, and T. D. Son, "Real-time MPC with Control Barrier Functions for Autonomous Driving using Safety Enhanced Collocation," in *8th IFAC Conference on Nonlinear Model Predictive Control (NMPC)*, vol. 58, no. 18. IFAC-PapersOnLine, 2024, pp. 392–399.

[17] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, 2nd ed., ser. Dover Books on Mathematics. Mineola, NY: Dover Publications, 2001.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, "Scikit-learn: Machine Learning in Python," 2018.

[19] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

Authorized licensed use limited to: KU Leuven University Library. Downloaded on April 22,2026 at 09:49:21 UTC from IEEE Xplore. Restrictions apply.