ARL-TR-10125 ● JULY 2025

Data-Driven Control Methods for High-Velocity Vehicle Control Surfaces

by D. Johann Djanal-Mann and Muthuvel Murugan

NOTICES

Disclaimers

ARL-TR-10125 ● JULY 2025

Data-Driven Control Methods for High-Velocity Vehicle Control Surfaces

D. Johann Djanal-Mann
Oak Ridge Associated Universities

Muthuvel Murugan
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
July 2025	Technical Report	2024 May 16	2025 May 15

4. TITLE AND SUBTITLE
Data-Driven Control Methods for High-Velocity Vehicle Control Surfaces

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
D. Johann Djanal-Mann and Muthuvel Murugan

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-VA Aberdeen Proving Ground, MD 21005	ARL-TR-10125

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCIDs: D. Johann Djanal-Mann, 0009-0005-1374-7765; Muthuvel Murugan, 0000-0002-8314-3650

14. ABSTRACT
This report investigates data-driven control strategies for high-velocity vehicle (HV) control surfaces, focusing on the demanding requirements imposed by high-velocity flight environments. Traditional control methods, reliant on analytical models, often struggle with the nonlinearities and uncertainties inherent in HV dynamics. To address these challenges, the study develops and compares two data-driven approaches: a classic proportional–integral–derivative (PID) controller and a modern differentiable predictive controller (DPC). Both are designed using machine-learned models trained on experimental data from a rapid control prototyping platform. The PID controller, tuned via neural state-space model, demonstrates robust, fast, and accurate reference tracking with negligible overshoot and steady-state error. The DPC uses a neural ordinary differential equation model to enable gradient-based optimization and end-to-end policy learning, achieving even faster rise and settling times than the PID in closed-loop simulations. While both controllers deliver high-performance actuation under the tested conditions, the DPC offers greater flexibility and efficiency for complex, nonlinear systems. These findings highlight how data-driven control and learning-based modeling are advancing the reliability and responsiveness of HV control surfaces, supporting future high-velocity maneuverability and survivability.

15. SUBJECT TERMS
data-driven engineering, high-velocity vehicle, PID control, predictive control, neural network

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	**b. ABSTRACT**	**C. THIS PAGE**		
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	20

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Muthuvel Murugan	(520) 691-5892

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

Acknowledgments

This research supports an Applied Research to Address Science and Technology Priorities program conducted by ARL in collaboration with the Naval Research Laboratory (NRL), Air Force Research Laboratory, and Missile Defense Agency under the sponsorship of the Office of the Under Secretary of Defense for Research and Engineering and Oak Ridge Associated Universities. We thank Ryan McGowan and Mike Kirkland (ARL); Jeff Fisher (NRL); Yan Borden and Dan Inman (University of Michigan); Richard McKee and Sun Yi (North Carolina Agricultural and Technical State University); the support teams at The MathWorks, Inc. and Speedgoat GmbH; and all the project team members and their collaborators for their support and valuable inputs on actuation concepts and control techniques suitable for the program.

1. Introduction

Conventional reentry vehicles, constrained by predictable trajectories, are vulnerable to interception by anti-ballistic missile systems. By contrast, high-velocity vehicles (HVs) use exceptional speed and maneuverability to achieve unpredictable flight paths, significantly enhancing survivability. This maneuverability is critically dependent on the control authority provided by control surfaces. Even small adjustments generate substantial side forces at high velocities[1]; thus, control surfaces are essential for maintaining stability and achieving desired trajectories in this demanding flight regime. Precise and responsive adjustment of these surfaces is paramount for successful HV operation.

However, the extreme environment of high-velocity flight introduces significant complexities to control system design. High temperatures and large, turbulent aerodynamic loads create highly nonlinear and challenging conditions.[2] Traditional control law development relies heavily on analytical mathematical models of system plant dynamics. These analytical approaches often necessitate accurate a priori knowledge of system behavior and can become unwieldy in the nonlinear scenarios of high-velocity flight. Fueled by advances in ML, the emergence of data-driven methods offers a compelling alternative. By leveraging experimental data to model system dynamics, these methods reduce the reliance on complete prior knowledge and complex analytical formulations, providing increased adaptability in highly variable conditions.

This research explores the application of data-driven techniques to the actuation of an HV control flap. We develop and compare two distinct control strategies: a classic proportional–integral–derivative (PID) controller and a modern differentiable predictive controller (DPC). PID control is a well-understood, robust, and easily implementable technique. The feedback-based mechanism continuously calculates the error between a desired setpoint and the actual measured value of a system by applying corrective actions u defined as

$$u(t) = K_P e(t) + K_I \int e(t)dt + K_D \frac{d\,e(t)}{dt}\,, \qquad (1)$$

where e is the error and K_P, K_I, and K_D are the gains of the proportional, integral, and derivative terms, respectively. PID control provides a strong baseline for comparison and allows for quick assessment of the basic controllability of the flap. However, it is a linear technique and has limitations, especially when dealing with the complex dynamics and constraints inherent in high-velocity flight. This motivated the development of the DPC, which uses a differentiable model of the system and a differentiable cost function to compute optimal control actions over a future horizon.[3] It builds upon the principles of model predictive control but uses differentiability to enable efficient gradient-based optimization and integration with learning-based approaches. Figure 1 illustrates the fundamental framework of the DPC. Both controllers are designed using data-driven methods and their performance is evaluated through simulations employing machine-learned models of the control surface dynamics. Sections 2–4 detail our methodology, present the simulation results, and discuss the comparative performance of these control approaches in the challenging high-velocity flight environment.

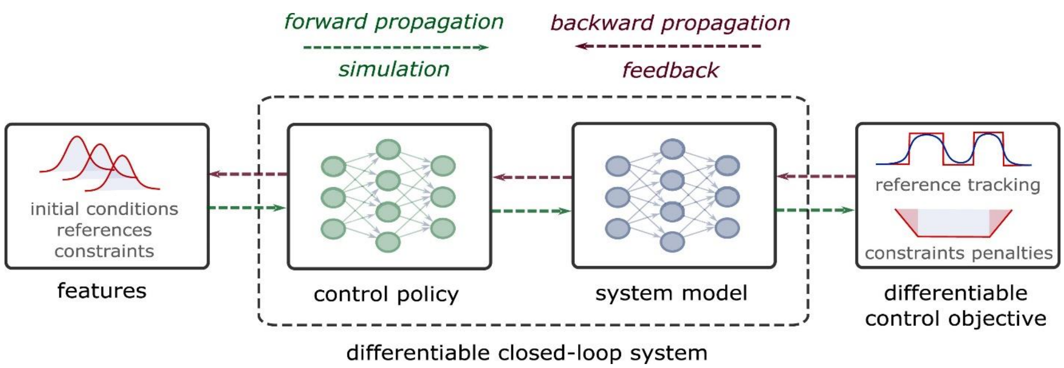

Figure 1. Conceptual methodology of the nonlinear DPC.[3]

2. Methodology

The controllers were developed and validated using telemetry data collected from a rapid control prototyping (RCP) platform established in prior work.[4] This platform, based in MATLAB/Simulink and using a real-time processor, facilitates the development and testing of control algorithms for a 16- × 10-inch titanium–zirconium–molybdenum control surface. Actuation is provided by a brushless DC motor and surface displacement is monitored by a laser profilometer.

2.1 Data Acquisition

Two distinct datasets were generated for system identification. Forty real-time simulations were conducted for the PID controller, each with a 4-s duration and 0.4-ms sampling time. The control surface was excited with a range of input signals $u(t)$, including sums of sinusoids and other waveforms at various amplitudes and frequencies. The corresponding state trajectories $x(t)$ were recorded, resulting in an input/output dataset $\{u(t), x(t)\}$. Figure 2 displays a sample of the dataset used. The inputs were chosen due to their representative nature of a typical mission profile. A separate dataset was collected for the DPC using similar input waveforms, but with a longer simulation duration of 10 s and a 5-ms sampling time to reduce computational overhead. This dataset was then partitioned into training, validation, and testing sets, each comprising 500 batches of 12 continuous time steps.

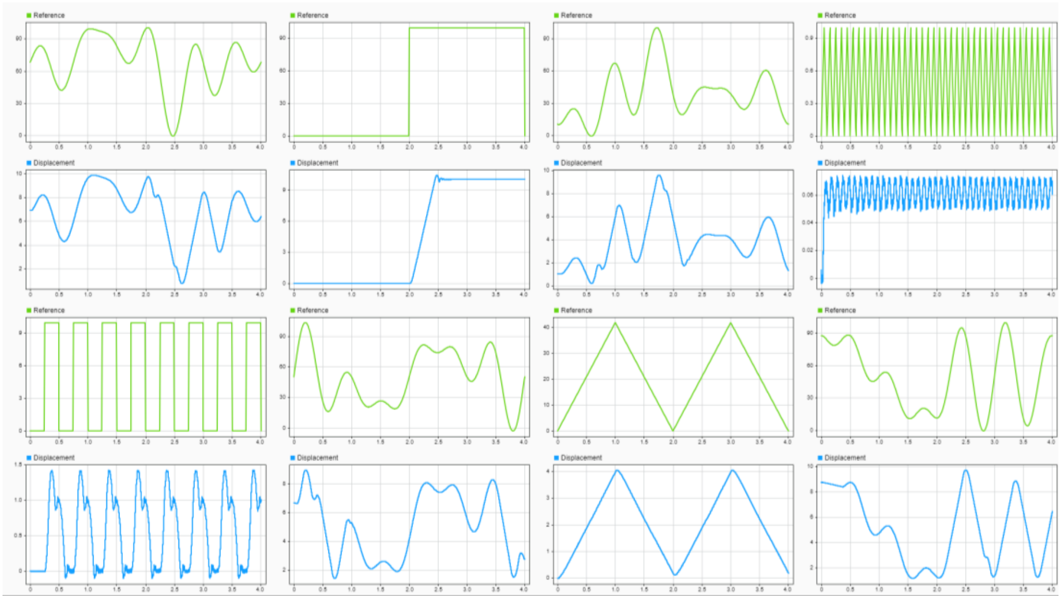

Figure 2. **Control surface system identification dataset featuring various waveforms with predictors (inputs) in green and responses (outputs) in blue.**

2.2 PID Controller Design

The PID controller was developed using MATLAB's Control System Toolbox.[5,6] A neural state-space (NSS) model was trained on the PID dataset to approximate the nonlinear system dynamics, $\dot{x} = f(x, u)$, by minimizing the mean squared error (MSE) between the predicted and measured state trajectories. Subsequently, a linear transfer function model of the form was estimated using step response data from the RCP platform. K_{DC}, T_Z, and T_{Pi} represent the gain, zero, and time constant associated with the i^{th} pole, respectively. This linear model served as the basis for tuning PID gains in the MATLAB PID Tuner app, prioritizing reference tracking performance and minimizing rise time.

$$G(s) = K_{DC}\frac{1 + T_Z\, s}{(1 + T_{P1}s)(1 + T_{P2}s)} \tag{2}$$

2.3 DPC Design

The DPC was implemented in Python using the NeuroMANCER scientific ML library. A neural ordinary differential equation (NODE) model was trained on the DPC dataset to learn a differentiable, black-box representation of the system dynamics $\dot{x} = f_\theta(x, u)$, where θ are the network parameters. The NODE was trained by minimizing the MSE between predicted and actual trajectories. A neural network control policy, $u = \pi(x)$, was then optimized to minimize a quadratic tracking error cost function. Gradient-based optimization methods were used, leveraging the differentiability of the NODE model to efficiently compute gradients. The learned dynamics enabled end-to-end simulation and policy refinement within the training loop.

3. Results

The NSS model architecture consisted of a multilayer perceptron (MLP) with one input (the command signal), one output (the plant's displacement), and two hidden layers. The hidden layers each contained 128 neurons and hyperbolic tangent activations, with parameters optimized using the Adam solver algorithm.[7] Figure 3 depicts the performance of the trained model on the test set. The NSS was trained over 350 epochs, dropping from an initial value above 2.7×10^4 to below 1.5×10^3. This trend indicates stable learning dynamics and successful minimization of the prediction error. The model structure was sufficient to capture the complex nonlinear behavior of the control surface, including regions with sharp transitions and oscillatory motion. Quantitatively, the model achieved a root MSE (RMSE) of 0.422675 and a coefficient of determination (R^2) of 0.819794 on the test set. These metrics demonstrate reasonable generalization and can reliably simulate the system's behavior for unseen input signals.

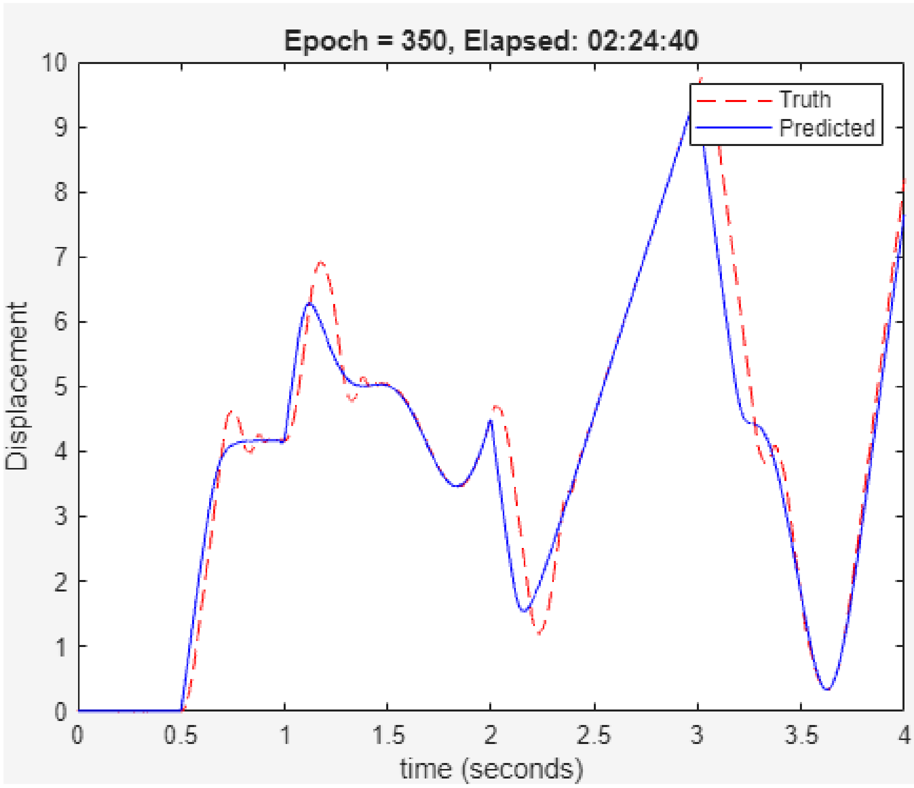

Figure 3. **NSS model performance on test dataset.**

Figure 4 and Table 1 summarize the results of the system identification step used to support PID controller tuning. The identified model closely tracks the measured step response, as shown in the overlay plot, confirming that the chosen structure captures the dominant system dynamics. Parameter values extracted from the fit—particularly, the small time constants T_{P1} and T_{P2}—indicate a fast dynamic response, which was essential for the subsequent controller design.

Using this model, PID gains were tuned with an emphasis on reference tracking and rise-time minimization. The resulting gains, shown at the bottom Figure 4, are characterized by a dominant integral term and negligible derivative action, effectively producing a proportional–integral (PI) controller:

$$[K_P, K_I, K_D] = [20.7,\ 2303.1,\ 0.0]\,. \tag{3}$$

This configuration is consistent with a system that requires swift response to errors without the need for damping predictive terms and reflects the high-speed, low-noise nature of the actuator response. The proportional gain K_P promotes aggressive error correction, quickly responding to deviations from the desired setpoint.[8] The high integral gain K_I further enhances performance by eliminating steady-state error.[9] The absence of derivative action ($K_D = 0$) is justified by the low-noise characteristics of the actuator response, avoiding potential noise amplification that can occur with derivative control.[10]

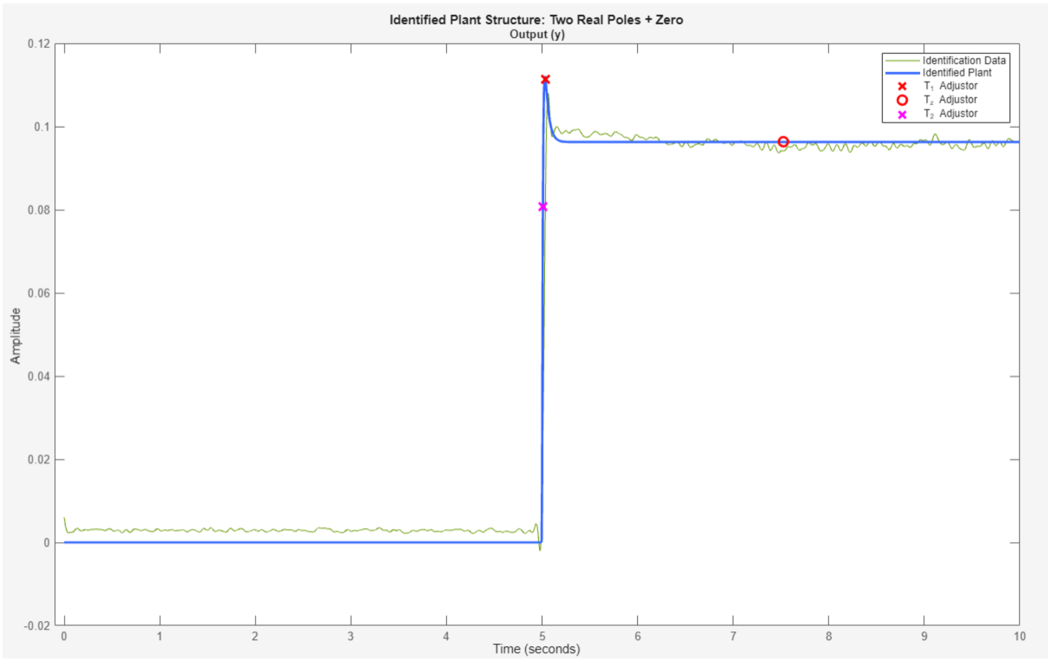

Figure 4. MATLAB PID Tuner transfer function system identification from RCP step response. The green curve is the input data; the blue curve is the identified plant.

Table 1. Estimated values for identified linear transfer function.

Parameter	Value
K_{DC}	0.0963
T_Z	0.0451
T_{P1}	0.0225
T_{P2}	0.0197

To evaluate the effectiveness of the tuned controller, a closed-loop simulation was conducted using the Simulink model (Figure 5). This integrates the designed PI controller with the NSS model of the actuator, along with a feedback connection scaled by a gain of 10 for compatibility with the input range.

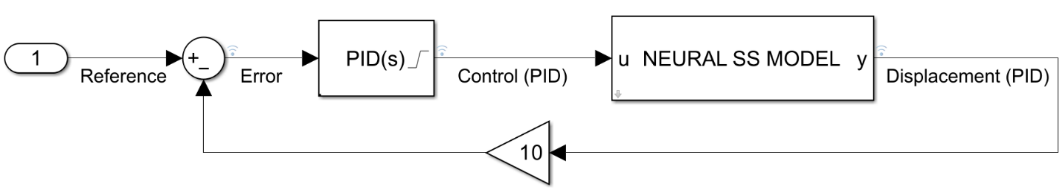

Figure 5. Simulink block diagram of the closed PID–NSS control loop.

The system's transient performance was assessed using a step response analysis, with results shown in Figure 6. Compared with the open-loop system, the closed-loop response exhibited significantly improved dynamic characteristics. Specifically, the rise time was reduced from 13.7 to just 1.03 ms, and the settling time improved from 118 to 14.2 ms. The overshoot decreased from 17.4% to 1.29%, and the steady-state error was fully eliminated (reduced from 0.0037 to 0.0). These improvements suggest that the controller achieves the primary tuning objectives of fast rise time and accurate reference tracking. The controlled system exhibits well-damped, high-speed performance without sacrificing steady-state accuracy, indicating the viability of the controller for real-time implementation.

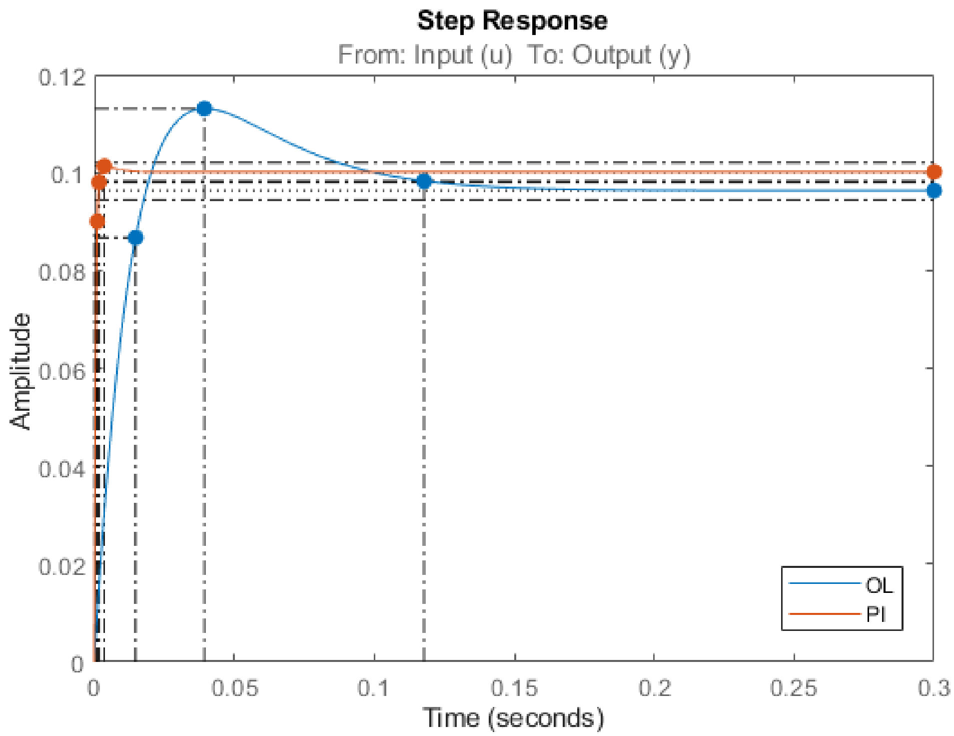

Figure 6. **Linear analysis of the PID–NSS system. The blue curve is the open-loop (OL) step response; the orange curve (PI) is the closed-loop step response.**

Figure 7 shows the ML architecture of the NODE model for the DPC. The NODE was composed of a recurrent neural network with one input, one output, and three hidden layers of 20 neurons each with hyperbolic tangent activation. A fourth-order Runge–Kutta integrator was used for time-stepping, and the model parameters were optimized using the Adam optimizer.

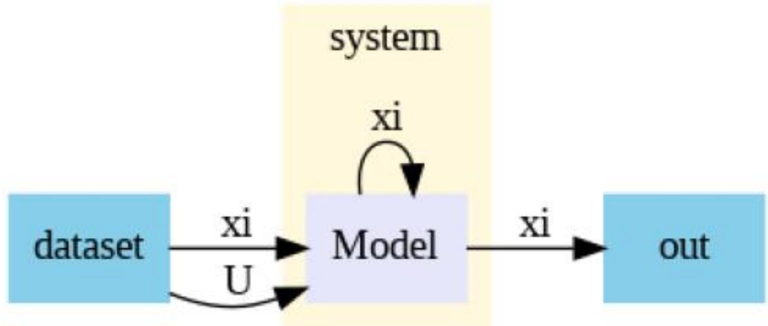

Figure 7. ML architecture for the DPC NODE model.

The performance of the NODE is shown in Figure 8. The model achieved an RMSE of 0.376951 and R^2 value of 0.966597 on the DPC test dataset. The learned dynamics closely approximate the ground truth and are sufficient for downstream control tasks. The smooth trajectory alignment across time steps indicates good generalization of the NODE to unseen data.

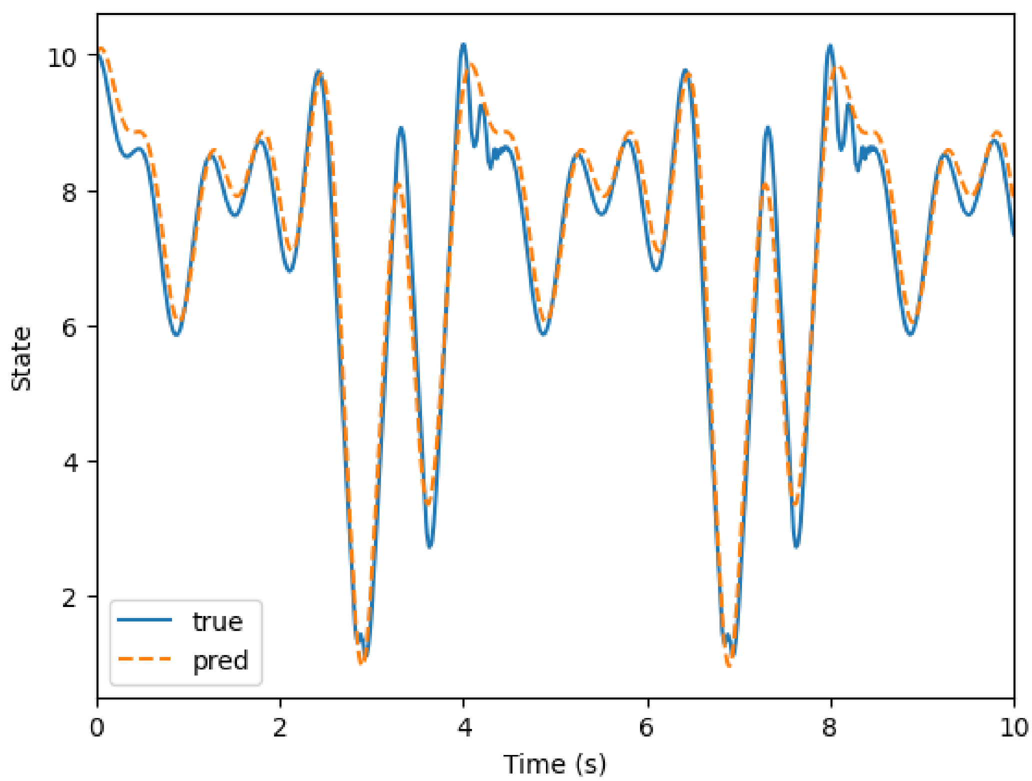

Figure 8. Learned NODE output (orange) compared with ground truth (blue) on the DPC test data.

The closed-loop control policy (Figure 9) was realized as an MLP with 2 inputs (the current system state and a time-dependent reference signal), 1 output (the control input), and 3 hidden layers of 20 neurons each. Figure 10 shows the closed-loop system's step response demonstrates significant performance gains over the open-loop configuration. Specifically, these include the following:

- Rise time improved from 0.2300 to 0.0100 s,
- Overshoot decreased from 17.9% to 1.16%,
- Settling time dropped from 0.44 to 0.02 s, and
- Steady-state error reduced from 0.018 to 0.001.

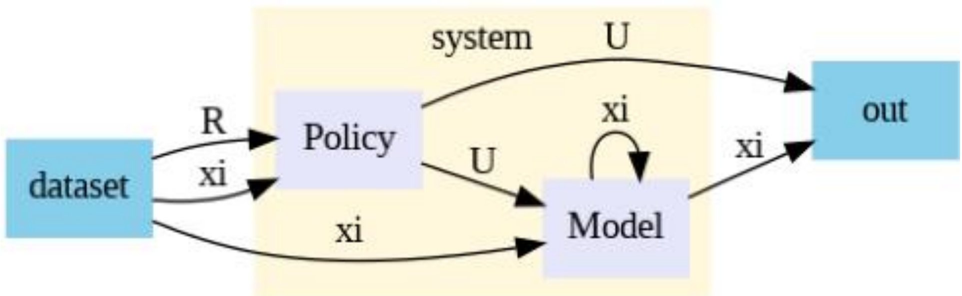

Figure 9. Closed-loop DPC block diagram with control policy integrated with NODE model.

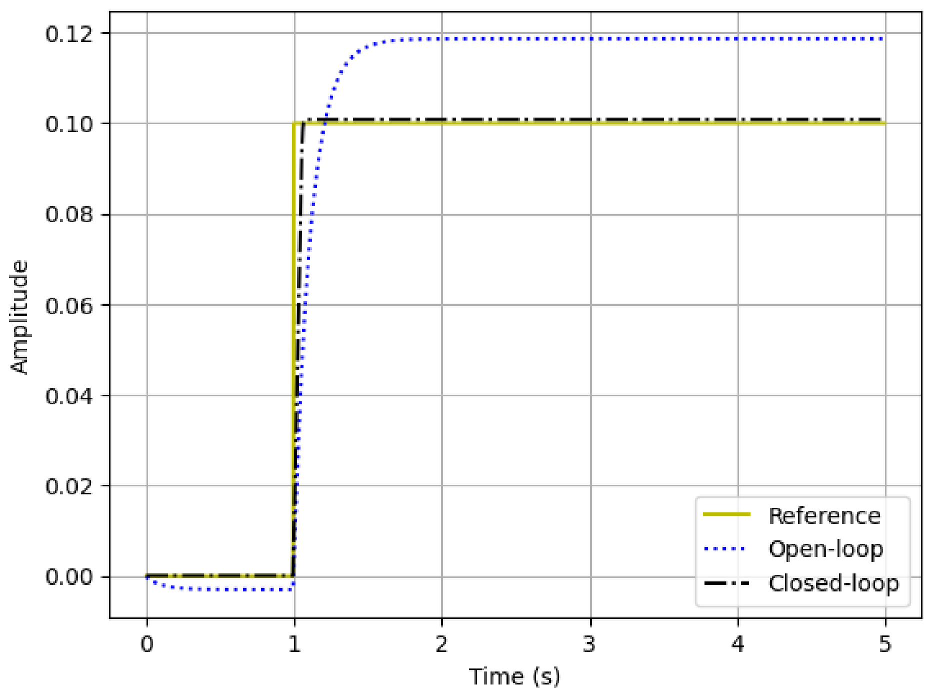

Figure 10. Step response of open-loop NODE model and closed-loop DPC.

These metrics illustrate the DPC framework's ability to achieve rapid, precise reference tracking with minimal transients. The substantial reductions in rise time, overshoot, and settling time highlight the effectiveness of jointly optimizing the model and control policy in a differentiable framework. These results suggest that the learned control law successfully captures key features of the system dynamics and uses them to produce high-performance closed-loop behavior, even in the absence of explicit analytical modeling.

4. Conclusion

This report presented two data-driven modeling and control strategies for HV applications: one grounded in traditional PID control architecture, and the other based on a DPC framework. The modeling phase first involved constructing an NSS model to capture the dynamics relevant to the control surface actuation system. In parallel, a NODE model was developed as a high-fidelity, differentiable simulator. Both models enabled simulation, analysis, and controller tuning in the absence of an analytical plant model, serving as a surrogate for real-time testing and system identification. The PID controller was optimized using classic tuning algorithms and applied to the NSS model. In contrast, the DPC approach used a neural policy and trained end-to-end via the differentiable NODE model to minimize a custom objective.

While the DPC ultimately achieved slightly faster rise and settling times in closed-loop simulations, the overall performance of both controllers was comparable in this specific case. The negligible overshoot, fast response, and stable behavior of the PID controller suggest that the underlying system was well-behaved and exhibited low levels of noise—conditions under which PID control tends to perform reliably. However, the DPC framework retains significant advantages in terms of flexibility and generalizability. The NODE model and neural control policy were each trained in only a few minutes, compared with the approximately 2.5 h needed to train the NSS. This efficiency, combined with the DPC's ability to incorporate complex nonlinearities directly into its learning architecture, makes it a compelling approach for high-velocity control scenarios characterized by high uncertainty, strong coupling, and rapidly varying dynamics.

Future work will embed both control strategies into the real-time processor of the RCP platform, enabling closed-loop evaluation on physical hardware. This next phase will test real-world performance, evaluate robustness to noise and delays, and offer practical insights into deployment of learning-based control systems in high-stakes aerospace environments.

5. References

1. National Research Council. Hypersonic technology for military application. National Academies Press; 1989.

2. Bose D et al. Uncertainty assessment of hypersonic aerothermodynamics prediction capability. National Aeronautical and Space Administration (US); 2011. NASA Technical Memorandum NASA/TM-2011-217082.

3. Drgoňa J, Kiš K, Tuor A, Vrabie D, Klaučo M. Differentiable predictive control: deep learning alternative to explicit model predictive control for unknown nonlinear systems. J Process Control. 2022;116:80–92.

4. Djanal-Mann DJ, Murugan M. Rapid control prototyping of hypersonic morphing vehicle control surfaces. Joint Army–Navy–NASA–Air Force (JANNAF); 2024.

5. Djanal-Mann DJ, Murugan M. Data-driven PID control experimentation in MATLAB for hypersonic vehicle control surface actuation. Joint Army–Navy–NASA–Air Force (JANNAF); 2024.

6. Djanal-Mann DJ, Murugan M. Application of PID control to hypersonic vehicle control surfaces. AIAA SciTech Forum; 2025 Jan 6–10; Orlando, FL. American Institute of Aeronautics and Astronautics (AIAA); 2025. https://doi.org/10.2514/6.2025-2632

7. Kingma DP, Ba JL. Adam: a method for stochastic optimization. arXiv; 2017 Jan 30. arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980

8. Dorf RC, Bishop RH. Modern control systems. 12th ed. Pearson; 2010.

9. Ogata K. Modern control engineering. 4th ed. Prentice Hall; 2001.

10. Nise NS. Control systems engineering. 6th ed. Wiley; 2011.

List of Symbols, Abbreviations, and Acronyms

ARL	U.S. Army Combat Capabilities Development Command Army Research Laboratory
DoD	Department of Defense
DPC	differentiable predictive controller
HV	high-velocity vehicle
ML	machine learning
MLP	multilayer perceptron
MSE	mean squared error
NODE	neural ordinary differential equation
NRL	Naval Research Laboratory
NSS	neural state-space
PI	proportional–integral
PID	proportional–integral–derivative
RCP	rapid control prototyping
RMSE	root MSE

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLB CI TECH LIB