Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Multi-Horizon Refinement of EDMDc for Aeroengine Model Predictive Control across Multiple Flight Conditions

Junfeng Shen
Shanghai Jiao Tong University
Huihui Li
Innovation Academy for Microsatellites of Chinese Academy of Sciences
Biao Chen
Shanghai Jiao Tong University
Jing Wang
wangjinger@sjtu.edu.cn

Shanghai Jiao Tong University
Qiang Shen
Shanghai Jiao Tong University

Research Article

Keywords: Aeroengine control, Extended dynamic mode decomposition with control, Model predictive control, Measurement histories, Multi-horizon refinement, Multiple flight conditions

Posted Date: November 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11464272/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Multi-Horizon Refinement of EDMDc for Aeroengine Model Predictive Control across Multiple Flight Conditions

Junfeng Shen[1*], Huihui Li[2], Biao Chen[1], Jing Wang[1*], Qiang Shen[1]

[1*]School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai, 200240, China.
[2*]Key Laboratory for Satellite Digitalization Technology, Innovation Academy for Microsatellites of CAS, Shanghai, China.

*Corresponding author(s). E-mail(s): wangjinger@sjtu.edu.cn;

Abstract

Aeroengine model predictive control (MPC) requires accurate predictions over multiple steps across changing flight conditions. This study develops multi-horizon refinement of extended dynamic mode decomposition with control (EDMDc) using measurement histories, protected fuel commands, and measured altitude and Mach number. The refinement minimizes normalized free-running errors over several horizons while preserving prescribed history shifts, with validation-based model selection and spectral screening. AGTF30 identification compares five dictionaries using 1500 s trajectories across sampled conditions spanning 0–12.192 km and Mach 0–0.8. The selected linear-delay model reduces the normalized 15 s test prefix score from 0.21807 to 0.20978 relative to its ridge initializer, with improvements varying by horizon and output. The refined predictor is integrated into constrained MPC with native protection, validated online updates, and proportional–integral (PI) fallback. Compared with PI under common command limits, the complete controller reduces corrected-fan-speed tracking root mean square error from 97.11 to 64.93 rpm in a 240 s repeated-command experiment and from 298.81 to 256.50 rpm in a 75 s staircase and large acceleration/deceleration experiment. Both improvements are accompanied by greater fuel-command variation and characterize the complete controller on the evaluated profiles.

Keywords: Aeroengine control, Extended dynamic mode decomposition with control, Model predictive control, Measurement histories, Multi-horizon refinement, Multiple flight conditions

1 Introduction

Aeroengine fuel control must achieve accurate speed tracking while respecting limits on rotor speed, turbine temperature, compressor pressure, and fuel delivery. These requirements are coupled through the nonlinear engine dynamics, such that adjustments in fuel flow affect both the speed response and thermodynamic operating conditions. Conventional turbofan control architectures coordinate a speed feedback controller with fuel limiters to balance tracking performance and engine protection [1, 2]. Switching between control loops also affects fuel command continuity, motivating compensators for smooth transitions [3]. Improving transient response within this architecture therefore requires control strategies that account for coupled dynamics and remain compatible with the existing protection system.

Model predictive control (MPC) coordinates control objectives and operating limits by optimizing an input sequence over a finite prediction horizon [4, 5]. For engine fuel regulation, the objective can balance speed tracking and fuel variation while accounting for constraints on predicted outputs and actuation. Engine applications include nonlinear predictive control [6], multiplexed predictive control [7], and scheduling of local linear MPC controllers [8]. Implementation studies have also evaluated turbofan MPC using real time simulation platforms [9], while adaptive formulations have combined subspace identification with direct performance control [10]. These approaches demonstrate the relevance of predictive control to engine regulation and the importance of the model used in the optimization.

The prediction model must represent the nonlinear response while permitting repeated constrained optimization. Koopman operator theory addresses this modeling requirement by describing the evolution of observable functions through a linear operator [11]. Extended dynamic mode decomposition (EDMD) approximates this operator from data using a finite dictionary of observables [12]. The choice of observables determines how nonlinear behavior is represented; related approaches include kernel approximations [13], dictionary learning [14], and neural embeddings with linear evolution [15]. These developments provide several ways to construct coordinates suited to prediction from measured trajectories.

Control inputs require their influence to be represented explicitly. Dynamic mode decomposition with control separates the effects of actuation from the identified dynamics [16], and extensions of Koopman theory provide formulations for systems with inputs [17]. Lifted predictors with linear input dependence can then be incorporated into MPC while retaining a quadratic optimization problem [18]. In turbofan applications, Koopman predictors have been combined with MPC and disturbance estimation to regulate spool speeds and engine pressure ratio [19]. This provides a foundation for using lifted models in engine control and motivates assessing their errors during repeated prediction.

Streaming and online DMD algorithms update identified dynamics as new measurements arrive, with weighted formulations allowing greater emphasis on recent observations [20, 21]. Recursive model updates have also been incorporated directly into Koopman MPC [22]. More recent studies address parametric uncertainty through

meta-adaptive Koopman models [23] and constrained tracking through online Koopman updates [24]. These methods establish online identification as a means of adapting lifted predictors to observed changes in system behavior.

Integrating a learned model into constrained control also requires attention to prediction uncertainty and optimization feasibility [25]. Existing approaches address these requirements through robust learning formulations [26], recursive model updates within robust MPC [27], and uncertainty descriptions for Koopman MPC [28]. For the present engine, the design question is how to refine a history-based EDMDc predictor for repeated propagation across sampled flight conditions and use it in constrained fuel control. One-step fitting evaluates individual transitions, whereas the controller depends on sequences of predicted outputs. This motivates an identification objective that evaluates free-running responses over several horizons. The model input must match the protected command applied to the actuator, and the optimization must remain compatible with the original limit selection logic.

To address these requirements, this study develops multi-horizon refinement of a structured EDMDc predictor for aeroengine MPC across multiple flight conditions. The predictor uses measured observation and protected-input histories with explicit flight-condition forcing. Starting from a regularized one-step estimate, offline refinement adjusts only the learned coefficients using weighted, normalized free-running errors over several horizons. Prescribed history shifts and the constant coordinate remain fixed. Spectral screening and validation-based selection determine the retained model.

The refined predictor is integrated into constrained MPC with empirical output margins, native protection, and proportional–integral (PI) fallback. Recursive candidate fitting, subsequent validation, and asynchronous adoption provide the online implementation. The experiments compare dictionaries, evaluate the change in prediction accuracy from the ridge initializer to the refined model, and examine longer uninterrupted predictions. Two control experiments then compare the complete EDMDc-MPC controller with PI under common command limits and changing flight conditions.

The remainder of this paper is organized as follows. Section 2 introduces the AGTF30 component-level model and the Koopman operator framework. Section 3 presents the engine control problem, structured EDMDc identification with multi-horizon refinement, and integration into MPC with online updates. Section 4 describes the simulation setup and analyzes refinement results, prediction accuracy, control performance, online adaptation, and execution times. Section 5 summarizes the findings and outlines directions for future research.

2 Preliminaries

This section introduces the AGTF30 component-level model, its measured engine and actuator variables, and the Koopman operator framework underlying the lifted prediction model.

2.1 AGTF30 component-level engine model

The Advanced Geared Turbofan 30,000 (AGTF30) is a nonlinear component-level engine simulation developed by NASA using the Toolbox for the Modeling and Analysis

of Thermodynamic Systems (T-MATS) [1, 29]. The engine comprises an inlet, a fan, low and high pressure compressors (LPC and HPC), a combustor, high and low pressure turbines (HPT and LPT), and separate core and bypass nozzles. The HPT drives the HPC, whereas the LPT drives the LPC and the fan, with a reduction gearbox coupling the fan to the low pressure shaft. A variable bleed valve (VBV) diverts part of the LPC discharge flow into the bypass duct, and a variable area fan nozzle (VAFN) adjusts the bypass exit area.

Figure 1 shows these gas-path and mechanical connections. The station labels locate the inlet temperature T_2, compressor-discharge pressure P_{s3}, and inter-turbine temperature T_{45} used in the observation model. The separate core and bypass paths, together with the coupled shafts, provide the physical basis for the component balances below.

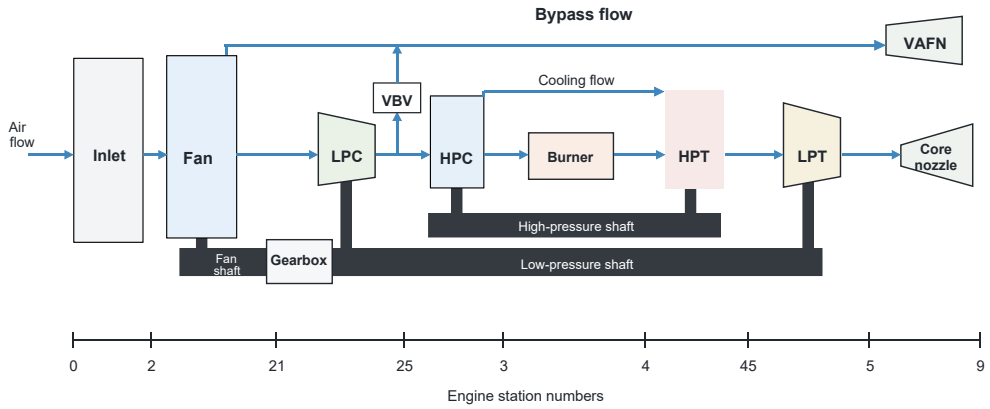


Fig. 1 AGTF30 component arrangement and engine stations.

The fan and compressors are described by characteristic maps and thermodynamic balances. For each component, the corrected mass flow and rotational speed are

$$W_c = W\frac{\sqrt{\vartheta}}{\delta}, \qquad N_c = \frac{N}{\sqrt{\vartheta}}, \qquad \vartheta = \frac{T_{t,\mathrm{in}}}{T_{\mathrm{ref}}}, \quad \delta = \frac{P_{t,\mathrm{in}}}{P_{\mathrm{ref}}}, \tag{1}$$

where W is mass flow, N is rotational speed, and T_t and P_t denote total temperature and total pressure. The reference quantities T_{ref} and P_{ref} define the standard conditions. At a matched operating point, the map relations take the form

$$\begin{aligned} W_c &= a_W \mathcal{W}(N_c/a_N, \mathcal{R}), \\ \pi_c &= 1 + a_\pi\big[\mathcal{P}(N_c/a_N, \mathcal{R}) - 1\big], \\ \eta_c &= a_\eta \mathcal{E}(N_c/a_N, \mathcal{R}), \end{aligned} \tag{2}$$

where $\mathcal{R}$ is the auxiliary map coordinate (Rline), $\mathcal{W}$, $\mathcal{P}$, and $\mathcal{E}$ are the flow, pressure-ratio, and efficiency maps, and a_N, a_W, a_π, a_η are component-specific scaling factors.

The compressor exit conditions satisfy

$$P_{t,\mathrm{out}} = \pi_c P_{t,\mathrm{in}}, \qquad h_{t,\mathrm{out}} = h_{t,\mathrm{in}} + \frac{h_{t,\mathrm{out},s} - h_{t,\mathrm{in}}}{\eta_c}, \tag{3}$$

where h_t denotes specific total enthalpy and the subscript s denotes the isentropic exit state at the prescribed outlet pressure. T-MATS evaluates the thermodynamic properties from temperature and gas composition and accounts for compressor bleed extraction in the mass and power balances.

The combustor adds the actual fuel flow W_f to the incoming gas stream. In the lower-heating-value formulation used by AGTF30, its mass, energy, and pressure relations are

$$\begin{aligned} W_{\mathrm{out}} &= W_{\mathrm{in}} + W_f, \\ W_{\mathrm{out}} h_{t,\mathrm{out}} &= W_{\mathrm{in}} h_{t,\mathrm{in}} + \eta_b W_f H_{\mathrm{LHV}}, \\ P_{t,\mathrm{out}} &= (1 - \zeta_b) P_{t,\mathrm{in}}, \end{aligned} \tag{4}$$

where η_b is combustion efficiency, H_{LHV} is the fuel lower heating value, and ζ_b is the fractional total-pressure loss. Here, W_f corresponds to the actual fuel flow W_{fact}, whose relation to the protected command is governed by the fuel actuator.

The turbines use characteristic maps to determine flow capacity and efficiency as functions of corrected speed and expansion ratio. Their Numerical Propulsion System Simulation (NPSS) maps retain the native conventions $N/\sqrt{T_{t,\mathrm{in}}}$ and $W\sqrt{T_{t,\mathrm{in}}}/P_{t,\mathrm{in}}$, with the associated map scaling factors. For an expansion ratio $\pi_t = P_{t,\mathrm{in}}/P_{t,\mathrm{out}}$, the expansion and power relations can be written as

$$\begin{aligned} P_{t,\mathrm{out}} &= P_{t,\mathrm{in}}/\pi_t, \\ h_{t,\mathrm{exp}} &= h_{t,m} - \eta_t (h_{t,m} - h_{t,s}), \\ \mathcal{P}_t &= W_m (h_{t,m} - h_{t,\mathrm{exp}}), \end{aligned} \tag{5}$$

where W_m and $h_{t,m}$ describe the gas after mixing with cooling flow admitted at the turbine inlet, η_t is turbine efficiency, and $\mathcal{P}_t$ is extracted power. The state $h_{t,\mathrm{exp}}$ precedes any outlet cooling-flow mixing; the final outlet state follows the corresponding mass and enthalpy balances. All quantities in these balances use consistent units, with the native unit conversions retained in the simulator.

The two shaft speeds evolve according to the net turbine and load torques,

$$J_H \dot{\omega}_H = \tau_{H,\mathrm{net}}, \qquad J_L \dot{\omega}_L = \tau_{L,\mathrm{net}}, \qquad \omega_f = \omega_L / i_g, \tag{6}$$

where $\omega_H, \omega_L, \omega_f$ are the high pressure shaft, low pressure shaft, and fan angular velocities. The net torques include turbine drive, compressor and fan loads, and modeled mechanical losses. The equivalent low pressure inertia J_L includes the fan inertia reflected through the gearbox as J_f/i_g^2; the adopted configuration uses $i_g = 3.1$. The corresponding speed signals are N_3, N_2, N_1, respectively, so the fan and low pressure shaft speeds are kinematically coupled.

Inlet and duct pressure losses, bleed and cooling flows, and core and bypass nozzle flow relations complete the gas path. The component-matching solver enforces flow

compatibility at the current shaft speeds, while the torque balances determine their transient evolution. The reported experiments retain the nominal component maps while varying fuel demand and prescribed flight conditions. The resulting physical model supplies the simulation trajectories; Section 3 develops the EDMDc predictor from measured histories and protected fuel commands.

The actual fuel flow W_{fact} responds dynamically to the protected fuel command W_{fdmd} through the fuel actuator and fuel-transport delay. These dynamics are retained in the simulation trajectories used for identification and are included in the actuator–engine–sensor prediction boundary defined in Section 3.1. Histories of measured observations, including actual fuel flow, and previously applied protected commands provide temporal information about this response for the EDMDc predictor.

2.2 Koopman operator

Koopman operator theory describes nonlinear dynamics through the linear evolution of functions of the system state, called observables [11, 30]. To introduce this representation, consider an autonomous system obtained by fixing the control input, operating conditions, and component parameters:

$$\dot{\mathbf{s}} = F(\mathbf{s}), \qquad \mathbf{s}(t) = S^t(\mathbf{s}_0), \tag{7}$$

where $\mathbf{s} \in \mathcal{S} \subseteq \mathbb{R}^{n_s}$ is the state, F is the vector field, and S^t is its flow map. For a scalar observable $\varphi : \mathcal{S} \to \mathbb{C}$, the Koopman operator is defined by composition with the flow,

$$(\mathcal{K}^t\varphi)(\mathbf{s}) = \varphi(S^t(\mathbf{s})). \tag{8}$$

The observable space is chosen to be closed under this composition. Although F may be nonlinear, $\mathcal{K}^t$ is linear in the observable: $\mathcal{K}^t(a\varphi + b\chi) = a\mathcal{K}^t\varphi + b\mathcal{K}^t\chi$ for scalars a, b. The operator generally acts on an infinite-dimensional space, motivating finite-dimensional approximations for control.

For sufficiently smooth observables in its domain, the infinitesimal generator $\mathcal{G}$ describes the continuous-time evolution [11, 31]:

$$(\mathcal{G}\varphi)(\mathbf{s}) = \lim_{\tau \to 0^+} \frac{(\mathcal{K}^\tau\varphi)(\mathbf{s}) - \varphi(\mathbf{s})}{\tau} = \nabla\varphi(\mathbf{s})^\mathsf{T} F(\mathbf{s}). \tag{9}$$

A nonzero eigenfunction χ_j satisfying $\mathcal{G}\chi_j = \lambda_j\chi_j$ evolves exponentially along trajectories. For a selected set of n_e eigenfunctions, define $\mathbf{\Phi} = [\chi_1, \ldots, \chi_{n_e}]^\mathsf{T}$ and $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_{n_e})$. Their evolution is

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{\Phi}(\mathbf{s}(t)) = \Lambda\mathbf{\Phi}(\mathbf{s}(t)), \qquad \mathbf{\Phi}(\mathbf{s}(t)) = e^{\Lambda t}\mathbf{\Phi}(\mathbf{s}_0). \tag{10}$$

If a vector of quantities of interest $\mathbf{q}(\mathbf{s})$ is well approximated by the span of these eigenfunctions, it can be reconstructed through Koopman modes:

$$\mathbf{q}(\mathbf{s}(t)) \approx C_\Phi \mathbf{\Phi}(\mathbf{s}(t)) = \sum_{j=1}^{n_e} \mathbf{c}_j \chi_j(\mathbf{s}_0) e^{\lambda_j t}, \tag{11}$$

where $\mathbf{c}_j$ is the j-th column of C_Φ. The reconstruction is exact only when the selected eigenfunctions span the required observables; such a finite basis is not available for every nonlinear system [32].

Sampling at the controller period T_s gives the discrete-time operator $\mathcal{K} = \mathcal{K}^{T_s}$. The corresponding eigenvalues satisfy

$$\mathcal{K}\chi_j = \mu_j \chi_j, \qquad \mu_j = e^{\lambda_j T_s}, \qquad \mathbf{\Phi}(\mathbf{s}_{k+1}) = e^{\Lambda T_s} \mathbf{\Phi}(\mathbf{s}_k). \tag{12}$$

This relation distinguishes the sampled operator spectrum from the continuous-time generator spectrum.

Exact eigenfunctions are usually unknown. A practical approximation instead uses a finite dictionary $\boldsymbol{\varphi} = [\varphi_1, \ldots, \varphi_{n_\varphi}]^\mathsf{T}$ and a matrix K fitted from sampled trajectories [12]:

$$\boldsymbol{\varphi}(\mathbf{s}_{k+1}) = K\boldsymbol{\varphi}(\mathbf{s}_k) + \mathbf{r}_\varphi(\mathbf{s}_k), \tag{13}$$

where $\mathbf{r}_\varphi$ contains the finite-dictionary and identification errors. An invariant observable subspace admits exact finite-dimensional closure, whereas a general dictionary yields an approximation whose accuracy depends on the observables and sampled data [32, 33]. This distinction motivates assessing prediction accuracy on trajectories beyond those used for fitting.

For control applications, varying inputs must be represented explicitly in the lifted predictor [17, 18]. Here, lifting uses measured output and protected input histories because the internal engine state is unavailable. Sections 3.1 and 3.2 specify these coordinates and construct the EDMDc model. The dictionary supports prediction over the sampled operating conditions without assuming that its functions are exact Koopman eigenfunctions.

3 Multi-horizon EDMDc refinement and predictive control

This section develops multi-horizon EDMDc refinement and its use in aeroengine MPC across multiple flight conditions. Section 3.1 defines the control problem and the measurement histories available for prediction. Section 3.2 constructs the structured EDMDc initializer and refines its learned coefficients using prediction errors over multiple horizons. Section 3.3 formulates MPC with fuel limits and empirical constraint tightening. Section 3.4 integrates online training, candidate validation, and asynchronous model commitment with PI fallback.

3.1 Control problem and measurement histories

We consider the fuel control loop of the AGTF30 engine under prescribed flight conditions [1]. The controller regulates the corrected fan speed through a fuel request $u_{f,k}$ processed by the original limit selector. The resulting protected command $u_k = W_{\mathrm{fdmd},k}$ drives the fuel actuator, while the variable bleed valve and variable area fan nozzle retain their original scheduling laws. The prediction boundary therefore includes actuator, engine, and sensor dynamics, represented by

$$\mathbf{s}_{k+1} = f(\mathbf{s}_k, u_k, \boldsymbol{\xi}_k; \boldsymbol{\theta}), \qquad \mathbf{o}_k = g(\mathbf{s}_k), \tag{14}$$

where $\mathbf{s}_k$ is the internal state, $\boldsymbol{\theta}$ describes component characteristics, and $\boldsymbol{\xi}_k = [h_{\mathrm{alt},k},\, \mathrm{Ma}_k]^{\mathsf{T}}$ contains the measured altitude and Mach number. The flight conditions are prescribed external inputs; fuel remains the only optimized actuator command.

The available feedback is collected in

$$\mathbf{o}_k = \begin{bmatrix} N_{1c,k} & N_{3,k} & T_{45,k} & P_{s3,k} & W_{\mathrm{fact},k} & V_{\mathrm{BV},k} & A_{\mathrm{AFN},k} & T_{2,k} \end{bmatrix}^{\mathsf{T}}. \tag{15}$$

Here, $N_{1c,k}$ and $N_{3,k}$ are the corrected fan speed and high pressure spool speed; $T_{45,k}$, $P_{s3,k}$, and $T_{2,k}$ are temperature, static pressure, and inlet temperature feedback. The actuator quantities $W_{\mathrm{fact},k}$, $V_{\mathrm{BV},k}$, and $A_{\mathrm{AFN},k}$ denote actual fuel flow, bleed valve position, and fan nozzle area, respectively. The actual fuel flow is retained as feedback because its response to the protected command depends on actuator dynamics.

For a prescribed speed reference r_k, define the tracking error and fuel increment as

$$e_k = N_{1c,k} - r_k, \qquad \Delta u_k = u_k - u_{k-1}. \tag{16}$$

The control objective combines accurate tracking with limited fuel variation, expressed by the stage cost

$$\ell_k = q_N \left(\frac{e_k}{s_N} \right)^2 + q_u \left(\frac{\Delta u_k}{s_f} \right)^2, \tag{17}$$

where $q_N, q_u > 0$ are weights and $s_N, s_f > 0$ are normalization scales. Admissible operation is subject to

$$\begin{aligned} & u_{\min} \le u_k \le u_{\max}, && |\Delta u_k| \le \dot{u}_{\max} T_s, \\ & 0 \le N_{1c,k} \le \overline{N}_1 \sqrt{T_{\mathrm{ref}}/T_{2,k}}, && \\ & N_{3,k} \le \overline{N}_3, && T_{45,k} \le \overline{T}_{45}, \qquad P_{s3,k} \le \overline{P}_{s3}. \end{aligned} \tag{18}$$

The fan-speed bound follows from $N_1 = N_{1c}\sqrt{T_2/T_{\mathrm{ref}}}$, with $T_{\mathrm{ref}} = 288.15\,\mathrm{K}$; it therefore varies with inlet temperature. The remaining bounds correspond to the original high-pressure shaft speed, measured temperature, and pressure limits. The controller also checks that its measured histories remain within the training-data domain.

Because the internal state $\mathbf{s}_k$ is not directly available, prediction is based on observation and protected input histories. For a history length L, define

$$\mathbf{x}_k = \left[\mathbf{o}_k^\mathsf{T} \ \cdots \ \mathbf{o}_{k-L+1}^\mathsf{T} \ u_{k-1} \ \cdots \ u_{k-L}\right]^\mathsf{T}. \tag{19}$$

These measurable coordinates supply temporal information associated with fuel actuation, rotor motion, and sensor responses. The use of measurement histories as observables is also established in Koopman approximation [34]. At sampling instant k, $\mathbf{x}_k$ contains the current and past observations together with previously applied protected commands, so its construction uses only information already available to the controller.

The prediction task is to relate $\mathbf{x}_k$ and a planned fuel sequence to future speed, temperature, and pressure responses for evaluating the tracking objective and operating constraints. Following the finite-dimensional Koopman representation introduced in Section 2.2, Section 3.2 constructs an EDMDc predictor on these measurable coordinates with explicit dependence on the protected fuel command.

3.2 Structured EDMDc identification with multi-horizon refinement

Extended dynamic mode decomposition with control (EDMDc) introduces an explicit control input into the EDMD prediction model [12, 17, 18]. Dictionary observables, including the history coordinates and any selected nonlinear features, are collected in a lifted vector, whose evolution is approximated by

$$\begin{aligned} \mathbf{z}_k &= \boldsymbol{\psi}(\mathbf{x}_k), \\ \widehat{\mathbf{z}}_{k+1|k} &= A\mathbf{z}_k + B\widetilde{u}_k, \qquad \widetilde{u}_k = \frac{u_k - \mu_u}{\sigma_u}. \end{aligned} \tag{20}$$

For the present engine control problem, $\mathbf{x}_k$ is the measurement history defined in Section 3.1, and u_k is the protected fuel command. The matrices $A \in \mathbb{R}^{n_z \times n_z}$ and $B \in \mathbb{R}^{n_z \times 1}$ describe the lifted dynamics and input influence, respectively, where n_z is the number of observables. The input mean μ_u and standard deviation σ_u are obtained from training data.

Each candidate dictionary $\boldsymbol{\psi}$ contains the normalized varying history coordinates and a constant. Candidate extensions add squares, quadratic products, radial basis functions, or Fourier functions of the current observations, with preprocessing fitted only on training data. Restricting nonlinear extensions to current observations limits the lifted dimension. The dictionary is selected by validation prediction error; the experiments retain the linear-delay dictionary. The four outputs needed for tracking and constraint evaluation are

$$\mathbf{y}_k = \left[N_{1c,k} \ N_{3,k} \ T_{45,k} \ P_{s3,k}\right]^\mathsf{T}, \qquad \widehat{\mathbf{y}}_{k+1|k} = C\widehat{\mathbf{z}}_{k+1|k}. \tag{21}$$

These outputs are included among the history coordinates, so C recovers their physical units directly from the training means and scales. The constant coordinate accounts for the offsets.

Figure 2 connects the measurement histories, lifted coordinates, and reconstructed outputs. The vertical lifting links describe how measured samples are represented for identification. During a free-running prediction, lifting is performed only at the initial history; subsequent predicted coordinates follow the lower sequence under the applied inputs. Thus, the diagram does not imply repeated correction by future observations. This distinction underlies the rollout evaluation introduced below. Bracketed indices in the diagram denote the same sampling instants as the subscripts in the equations. The diagram shows the fuel-input form; measured altitude and Mach number enter through the additional forcing term in Eq. 31.

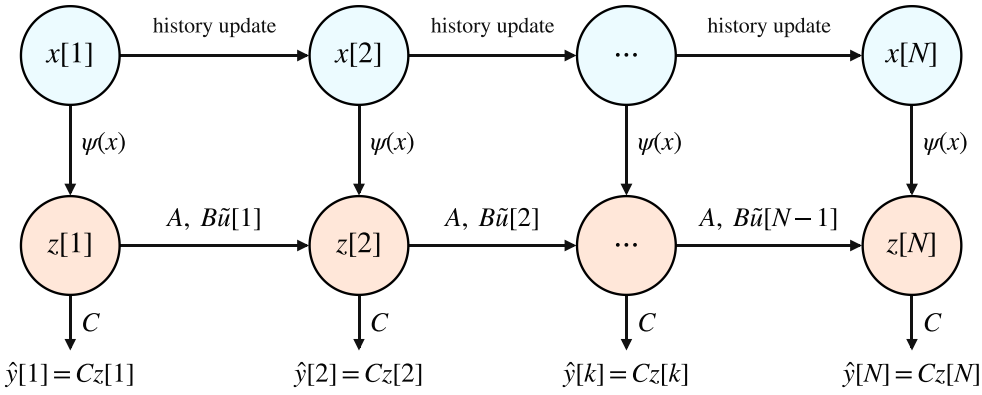


Fig. 2 History-based EDMDc prediction model for aeroengine fuel regulation.

The standard EDMDc estimate is obtained from pairs of consecutive lifted samples. For one uninterrupted training segment, reindex the samples with complete histories as $\mathbf{z}_0, \ldots, \mathbf{z}_M$ and form

$$\begin{aligned} Z_- &= \begin{bmatrix} \mathbf{z}_0 & \mathbf{z}_1 & \cdots & \mathbf{z}_{M-1} \end{bmatrix}, \\ Z_+ &= \begin{bmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \cdots & \mathbf{z}_M \end{bmatrix}, \\ \widetilde{U}_- &= \begin{bmatrix} \widetilde{u}_0 & \widetilde{u}_1 & \cdots & \widetilde{u}_{M-1} \end{bmatrix}. \end{aligned} \tag{22}$$

Each input column corresponds to the transition between the matching columns of Z_- and Z_+. For multiple segments, valid pairs are formed within each segment before concatenating the matrices columnwise; M then denotes the total number of pairs. Complete trajectories are assigned to training, validation, and test sets before histories are formed, and all normalization statistics are fitted on training data alone.

Stacking the lifted coordinates and inputs gives the regression problem

$$Z_+ \approx AZ_- + B\widetilde{U}_- = \Theta\Omega_-, \qquad \Theta = \begin{bmatrix} A & B \end{bmatrix}, \qquad \Omega_- = \begin{bmatrix} Z_- \\ \widetilde{U}_- \end{bmatrix}. \tag{23}$$

Here, $\Omega_- \in \mathbb{R}^{(n_z+1)\times M}$. A regularized EDMDc estimate minimizes the one-step fitting error:

$$\Theta_{\mathrm{EDMDc}} = \arg\min_{\Theta} \left\{ \frac{1}{M} \|Z_+ - \Theta\Omega_-\|_F^2 + \lambda_{\mathrm{reg}} \|\Theta\|_F^2 \right\}. \tag{24}$$

Setting the derivative with respect to Θ to zero gives

$$\Theta_{\mathrm{EDMDc}} = Z_+ \Omega_-^{\mathsf{T}} \left(\Omega_- \Omega_-^{\mathsf{T}} + M\lambda_{\mathrm{reg}} I \right)^{-1}. \tag{25}$$

The factor M follows from averaging the squared residuals, and $\lambda_{\mathrm{reg}} > 0$ regularizes correlated regressors. Numerically, the associated linear system is solved by factorization without explicitly forming the inverse [35].

The history structure supplies known relationships that can be imposed on this estimate [36]. Past observations and inputs shift by one position, the new fuel command enters the most recent input position, and the constant remains one. Their rows in A and B are therefore set directly, including the adjustments required by normalization. Only the rows for new observations and any added nonlinear features retain fitted coefficients; their index set is $\mathcal{L}$. Denote the resulting structured initializer by $\Theta_{\mathrm{ls}} = [A_{\mathrm{ls}}\ B_{\mathrm{ls}}]$, with $\Theta_{\mathrm{ls},\mathcal{L}} = [\Theta_{\mathrm{EDMDc}}]_{\mathcal{L}}$. Since Eq. 24 separates by rows, these learned coefficients are also obtained by fitting only the unknown rows. The prescribed rows remain fixed during subsequent refinement and online updates.

A small one-step residual does not necessarily imply accurate predictions after repeated propagation. Following the use of trajectory errors in turbofan Koopman identification [19], the structured model is further assessed and refined using continuous sequences of predicted outputs. At each valid origin i, initialize once from the measured history and propagate for H steps:

$$\begin{aligned} \widehat{\mathbf{z}}_{i|i} &= \boldsymbol{\psi}(\mathbf{x}_i), \\ \widehat{\mathbf{z}}_{i+h|i} &= A\widehat{\mathbf{z}}_{i+h-1|i} + B\widetilde{u}_{i+h-1}, \\ \widehat{\mathbf{y}}_{i+h|i} &= C\widehat{\mathbf{z}}_{i+h|i}, \qquad h = 1, \ldots, H. \end{aligned} \tag{26}$$

The predictor uses recorded protected commands throughout the interval. Subsequent measured outputs serve only as comparison targets; they do not reset the predicted state or history. Each prediction interval must remain within one uninterrupted trajectory.

The error over the complete predicted response is measured by

$$J_{\mathrm{roll}}(\Theta; \mathcal{O}, H) = \frac{1}{4|\mathcal{O}|H} \sum_{i\in\mathcal{O}} \sum_{h=1}^{H} \sum_{j=1}^{4} \left(\frac{\widehat{y}_{j,i+h|i} - y_{j,i+h}}{s_j} \right)^2, \tag{27}$$

where $\mathcal{O}$ contains valid prediction origins and $s_j > 0$ is the training standard deviation of output j. Normalization makes the four outputs comparable despite their different physical units. Summing over all consecutive steps accounts for error accumulation throughout each prediction interval.

Multi-horizon refinement combines normalized rollout losses over prescribed prediction lengths. Each loss includes all steps up to its horizon, so the objective evaluates

both shorter responses and accumulated errors during longer propagation. These identification horizons determine the training objective and are distinct from the MPC prediction horizon N_p. For prescribed horizons $\mathcal{H}_{\mathrm{id}}$ and weights $w_H > 0$ summing to one, define

$$J_{\mathrm{mh}}(\Theta; \mathcal{O}) = \sum_{H \in \mathcal{H}_{\mathrm{id}}} w_H J_{\mathrm{roll}}(\Theta; \mathcal{O}, H). \tag{28}$$

Starting from Θ_{ls}, the learned rows are refined by solving

$$\begin{aligned} \underset{\Theta_{\mathcal{L}}}{\text{minimize}} \quad & J_{\mathrm{mh}}(\Theta; \mathcal{O}_{\mathrm{tr}}) \\ & + \lambda_{\mathrm{roll}} \left\| \Theta_{\mathcal{L}} - \Theta_{\mathrm{ls},\mathcal{L}} \right\|_F^2 . \end{aligned} \tag{29}$$

Here, $\lambda_{\mathrm{roll}} > 0$ limits departure from the initializer. The remaining matrix rows, dictionary, normalization, and decoder are held fixed. L-BFGS with backtracking, stopping tolerances, and an iteration budget is used for offline refinement [37]. A trial step must produce finite predictions, reduce the training objective, and satisfy a spectral screen on the dynamic block after separating the prescribed constant coordinate. Growing dynamic modes are rejected rather than accepted for later validation.

The initializer and every eligible refinement iterate are compared on separate validation trajectories using $\sqrt{J_{\mathrm{mh}}}$, with common output scales and prediction origins. A candidate is retained only if it improves the best validation score and its score at each horizon is no more than a prescribed factor above the initializer's score. The best retained iterate, rather than the last optimizer iterate, supplies the initial model M_0. After selection, prediction accuracy is reported using normalized prefix scores and endpoint root mean square error (RMSE), expressed in physical units or normalized by the training range as specified in the results. Longer test trajectories are also propagated from a single initialization to assess accumulated error beyond the MPC horizon.

For control, repeated application of Eq. 20 gives

$$\widehat{\mathbf{y}}_{k+h|k} = CA^h \mathbf{z}_k + \sum_{i=0}^{h-1} CA^{h-1-i} B \widetilde{u}_{k+i|k}, \qquad h = 1, \ldots, N_p, \tag{30}$$

where $\widetilde{u}_{k+i|k}$ is the normalized fuel command planned at instant k for instant $k+i$. This uses the same propagation as Eq. 26, with planned inputs replacing recorded inputs. With A, B, C fixed over the horizon, the predicted outputs are affine in the fuel sequence and can be used in the MPC formulation of Section 3.3. Algorithm 1 summarizes identification and model selection.

For varying flight conditions, the fuel-input derivation is extended with the measured external input $\boldsymbol{\xi}_k$:

$$\widehat{\mathbf{z}}_{k+1|k} = A\mathbf{z}_k + B\widetilde{u}_k + E\widetilde{\boldsymbol{\xi}}_k, \qquad E \in \mathbb{R}^{n_z \times 2}. \tag{31}$$

The external inputs are normalized using training means and standard deviations. The preceding regression and rollout equations apply with the augmented regressor $[\mathbf{z}_k^{\mathsf{T}}, \widetilde{u}_k, \widetilde{\boldsymbol{\xi}}_k^{\mathsf{T}}]^{\mathsf{T}}$ and coefficient matrix $\Theta = [A \; B \; E]$. The added columns are fitted and

Algorithm 1 Structured EDMDc identification with multi-horizon refinement

Require: Training, validation, and test trajectories; candidate settings; $\mathcal{H}_{\mathrm{id}}, w_H$
Ensure: Initial predictor M_0

```
for each prescribed history, dictionary, and regularization setting do
    Fit normalization and construct ψ, C from training data
    Form Z_−, Z_+, Ũ_− using Eq. 22
    Obtain the regularized EDMDc estimate using Eq. 25
    Prescribe the history-shift and constant rows to form Θ_ls
    Refine only rows L using Eqs. 26–29
    Reject growing dynamic modes; score each eligible iterate on common validation origins
    Retain the best validation iterate subject to the per-horizon regression guard
end for
Select the best retained configuration and store M_0 with its preprocessing
Evaluate test prediction errors, including longer uninterrupted rollouts
```

refined only in the learned rows. Offline validation uses the recorded external-input sequence. During control, the current measured altitude and Mach number are held constant over the prediction horizon; future flight-condition samples are not supplied to the optimizer.

3.3 Model predictive control formulation

The controller uses the discrete EDMDc model from Section 3.2 to predict engine responses to planned fuel commands, following the Koopman-MPC formulation [18, 19]. Figure 3 shows how prediction, constraint information, and model updates enter the feedback loop. Measured histories initialize the predictor, which supplies future outputs for the tracking objective and constraint evaluation. The constraint-update block supplies the current executable fuel bounds and empirical output margins. The diagram labels the fuel request as u_f, corresponding to $u_{f,k}$; the original limit selector produces the protected command u_k used by the predictor. The measured-feedback connection, labeled y schematically, represents the full observation vector $\mathbf{o}_k$ in Eq. 15; the predictor output $\mathbf{y}_k$ is its four-output subset. The prediction structure, normalization, and decoder remain fixed while eligible model parameters may change after validation; the corresponding online procedure is detailed in Section 3.4.

To express the commands in physical fuel units, Eq. 31 is written as

$$\begin{aligned} \widehat{\mathbf{z}}_{k+i+1|k} &= A\widehat{\mathbf{z}}_{k+i|k} + B_u u_{k+i|k} + \mathbf{b}_{u,k}, \\ \widehat{\mathbf{y}}_{k+i+1|k} &= C\widehat{\mathbf{z}}_{k+i+1|k}, \end{aligned} \tag{32}$$

where $B_u = B/\sigma_u$ and $\mathbf{b}_{u,k} = -\mu_u B_u + E\widetilde{\boldsymbol{\xi}}_k$. The known offset therefore combines input normalization with the current flight conditions, held constant within the horizon. Predictions start from $\widehat{\mathbf{z}}_{k|k} = \boldsymbol{\psi}(\mathbf{x}_k)$, using the measurement and protected fuel histories available at instant k. The active model is held fixed throughout each prediction horizon.

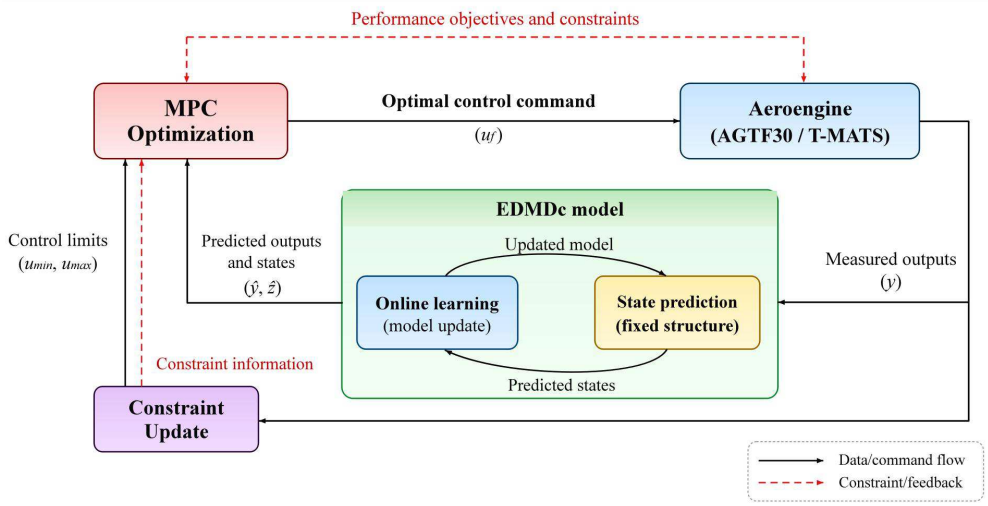


Fig. 3 Functional architecture of the EDMDc-MPC controller.

Successive substitution in Eq. 32 gives the first two output predictions:

$$\begin{aligned}\widehat{\mathbf{y}}_{k+1|k} &= CA\mathbf{z}_k + CB_u u_{k|k} + C\mathbf{b}_{u,k},\\ \widehat{\mathbf{y}}_{k+2|k} &= CA^2\mathbf{z}_k + CAB_u u_{k|k} + CB_u u_{k+1|k} + C(A+I)\mathbf{b}_{u,k}.\end{aligned} \tag{33}$$

These expressions show how each planned command enters the predictions and how its effect propagates through powers of A. Repeating the substitution gives all outputs over the prediction horizon N_p.

The controller optimizes $N_c \leq N_p$ fuel commands, collected in

$$\mathbf{U}_k = \begin{bmatrix} u_{k|k} & u_{k+1|k} & \cdots & u_{k+N_c-1|k}\end{bmatrix}^{\mathsf{T}}. \tag{34}$$

Beyond the control horizon, the last planned command is held constant, so $u_{k+i|k} = u_{k+N_c-1|k}$ for $i = N_c, \ldots, N_p - 1$.

Collecting the output predictions in the same way gives

$$\begin{aligned}\mathbf{Y}_k &= \begin{bmatrix}\widehat{\mathbf{y}}_{k+1|k}^{\mathsf{T}} & \cdots & \widehat{\mathbf{y}}_{k+N_p|k}^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}\\ &= \mathbf{f}_k + \mathcal{M}\mathbf{U}_k.\end{aligned} \tag{35}$$

The vector $\mathbf{f}_k$ collects the terms determined by the current history, flight conditions, and normalization offset. The matrix $\mathcal{M}$ describes the effects of the planned commands:

$$\mathcal{M} = \begin{bmatrix} CB_u & 0 & \cdots & 0 \\ CAB_u & CB_u & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N_c-1}B_u & CA^{N_c-2}B_u & \cdots & CB_u \\ CA^{N_c}B_u & CA^{N_c-1}B_u & \cdots & S_{N_c+1} \\ \vdots & \vdots & & \vdots \\ CA^{N_p-1}B_u & CA^{N_p-2}B_u & \cdots & S_{N_p} \end{bmatrix}. \tag{36}$$

The hth four-entry block of $\mathbf{f}_k$ and the accumulated last-column blocks are

$$\begin{aligned} \mathbf{f}_{k,h} &= CA^h\mathbf{z}_k + \sum_{\ell=0}^{h-1} CA^\ell \mathbf{b}_{u,k}, \quad h = 1, \ldots, N_p, \\ S_h &= \sum_{\ell=0}^{h-N_c} CA^\ell B_u, \qquad\qquad h = N_c, \ldots, N_p. \end{aligned} \tag{37}$$

The lower triangular structure reflects that a command cannot affect earlier outputs. Beyond N_c, the blocks S_h collect repeated applications of the final command. The resulting matrix has $4N_p$ rows and N_c columns.

The fuel sequence is selected by minimizing a finite horizon cost that penalizes predicted speed tracking error and fuel increments:

$$\begin{aligned} J_k(\mathbf{U}_k) = &\sum_{h=1}^{N_p} q_N \left(\frac{\widehat{N}_{1c,k+h|k} - r_{k+h|k}}{s_N} \right)^2 \\ &+ \sum_{i=0}^{N_c-1} q_u \left(\frac{\Delta u_{k+i|k}}{s_f} \right)^2. \end{aligned} \tag{38}$$

Here, $\widehat{N}_{1c,k+h|k}$ is the predicted corrected fan speed, and $r_{k+h|k}$ is the reference available at instant k; the implementation holds it at r_k throughout the horizon. The weights and normalization scales follow Eq. 17. The planned increment is $\Delta u_{k+i|k} = u_{k+i|k} - u_{k+i-1|k}$, with $u_{k-1|k} = u_{k-1}$ taken from the previous protected fuel command.

Since N_{1c} is the first predicted output, its predictions are obtained by taking the first row of each four-output block in Eq. 35. Denote the corresponding vector and matrix by $\mathbf{f}_{N,k}$ and $\mathcal{M}_N$, and stack the references as $\mathbf{r}_k = [r_{k+1|k}, \ldots, r_{k+N_p|k}]^\mathsf{T}$. The speed tracking error is then $\mathbf{f}_{N,k} + \mathcal{M}_N\mathbf{U}_k - \mathbf{r}_k$. Similarly, subtracting successive fuel

commands gives

$$\Delta \mathbf{U}_k = D_u \mathbf{U}_k - \mathbf{d}_k,$$
$$D_u = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -1 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & 0 \\ 0 & \cdots & -1 & 1 \end{bmatrix}, \qquad \mathbf{d}_k = \begin{bmatrix} u_{k-1} & 0 & \cdots & 0 \end{bmatrix}^{\mathsf{T}}. \tag{39}$$

With $\overline{Q} = (q_N/s_N^2)I_{N_p}$, $\overline{R} = (q_u/s_f^2)I_{N_c}$, and $\|\mathbf{v}\|_W^2 = \mathbf{v}^{\mathsf{T}} W \mathbf{v}$, Eq. 38 becomes

$$J_k(\mathbf{U}_k) = \|\mathbf{f}_{N,k} + \mathcal{M}_N \mathbf{U}_k - \mathbf{r}_k\|_{\overline{Q}}^2 + \|D_u \mathbf{U}_k - \mathbf{d}_k\|_{\overline{R}}^2. \tag{40}$$

Fuel magnitude and rate limits are imposed on each planned command. Constraint tightening reserves part of the admissible output range for prediction uncertainty, as in robust MPC formulations [38]. Here, the predicted output limits are tightened using empirical error margins to account for discrepancies between model predictions and measured responses:

$$\begin{aligned} u_{\min} &\le u_{k+i|k} \le u_{\max}, \\ |\Delta u_{k+i|k}| &\le \dot{u}_{\max} T_s, \\ \underline{\mathbf{y}} + \mathbf{m}_{k,h} &\le \widehat{\mathbf{y}}_{k+h|k} \le \overline{\mathbf{y}}_{k,h} - \mathbf{m}_{k,h}, \end{aligned} \tag{41}$$

where $i = 0, \ldots, N_c - 1$, $h = 1, \ldots, N_p$, and T_s is the sampling interval. The vectors $\underline{\mathbf{y}}$ and $\overline{\mathbf{y}}_{k,h}$ collect the output bounds defined in Eq. 18, with inequalities applied componentwise. Each nonnegative margin $\mathbf{m}_{k,h}$ is expressed in the corresponding output units and calibrated from training and subsequently observed prediction residuals.

For the corrected fan-speed upper bound, the implementation uses

$$\overline{N}_{1c,k,h} = \min\left\{ N_{1c,\max}^{\mathrm{tr}}, \overline{N}_1 \sqrt{\frac{T_{\mathrm{ref}}}{T_{2,k} + \Delta T_{2,h}^{\mathrm{tr}}}} \right\}. \tag{42}$$

Here, $N_{1c,\max}^{\mathrm{tr}}$ is the training-domain upper bound and $\Delta T_{2,h}^{\mathrm{tr}}$ is an inflated maximum positive inlet-temperature change observed in valid training windows at horizon h. This empirical guard uses current measurements and training data, not future measured temperature, and is not a certified future bound. The first fuel move additionally lies inside the current native protection interval $[u_{\min,k}^{\mathrm{nat}}, u_{\max,k}^{\mathrm{nat}}]$. These two scalar inequalities are included in the quadratic program (QP) below; native fuel limits at future steps are not predicted.

For each model version, training prediction residuals initialize the margins. Before candidate adoption, the completed validation trajectory can enlarge them:

$$\begin{aligned} m_{j,h}^0 &= \gamma \max_{i \in \mathcal{O}_{\mathrm{tr}}} \left| \widehat{y}_{j,i+h|i} - y_{j,i+h} \right|, \\ m_{j,h}^v &= \max\left\{ m_{j,h}^0, \left| \widehat{y}_{j,i_v+h|i_v}^v - y_{j,i_v+h} \right| \right\}. \end{aligned} \tag{43}$$

Here, $\mathcal{O}_{\mathrm{tr}}$ contains valid training origins, $\gamma = 1.1$ is the training-error inflation factor, and i_v is the completed validation origin of candidate version v. The second expression applies for $h \le \min(N_v, N_p)$; remaining horizons retain $m^0_{j,h}$. Residuals use recorded protected commands without intermediate state resets. The initial model uses $m^0_{j,h}$; after adoption, $m_{j,k,h} = m^v_{j,h}$ until the next model change. These margins summarize observed errors rather than certified future error bounds.

For the optimization, stack the tightened bounds $\underline{\mathbf{y}} + \mathbf{m}_{k,h}$ and $\overline{\mathbf{y}}_{k,h} - \mathbf{m}_{k,h}$, for $h = 1, \dots, N_p$, into $\underline{\mathbf{Y}}_k$ and $\overline{\mathbf{Y}}_k$, respectively. Substituting Eqs. 35 and 39 into the constraints yields

$$G_{\mathrm{qp}}\mathbf{U}_k \le \mathbf{b}^{\mathrm{qp}}_k, \qquad G_{\mathrm{qp}} = \begin{bmatrix} I_{N_c} \\ -I_{N_c} \\ D_u \\ -D_u \\ \mathcal{M} \\ -\mathcal{M} \\ \mathbf{e}_1^{\mathsf{T}} \\ -\mathbf{e}_1^{\mathsf{T}} \end{bmatrix}, \qquad \mathbf{b}^{\mathrm{qp}}_k = \begin{bmatrix} u_{\max}\mathbf{1} \\ -u_{\min}\mathbf{1} \\ \delta_u\mathbf{1} + \mathbf{d}_k \\ \delta_u\mathbf{1} - \mathbf{d}_k \\ \overline{\mathbf{Y}}_k - \mathbf{f}_k \\ \mathbf{f}_k - \underline{\mathbf{Y}}_k \\ u^{\mathrm{nat}}_{\max,k} \\ -u^{\mathrm{nat}}_{\min,k} \end{bmatrix}, \tag{44}$$

where $\delta_u = \dot{u}_{\max} T_s$ and $\mathbf{1} \in \mathbb{R}^{N_c}$ is the vector of ones. Here, $\mathbf{e}_1 = [1, 0, \dots, 0]^{\mathsf{T}} \in \mathbb{R}^{N_c}$ selects the first command. The first two block rows impose fuel magnitude limits, the next two impose rate limits, and the following two impose tightened output bounds. The final two rows impose the current native protection interval on the first command. Rows corresponding to absent upper or lower output bounds are omitted.

Expanding Eq. 40 and dropping terms independent of $\mathbf{U}_k$ gives the quadratic program

$$\begin{aligned} \underset{\mathbf{U}_k}{\text{minimize}} \quad & \frac{1}{2}\mathbf{U}_k^{\mathsf{T}} H \mathbf{U}_k + \mathbf{c}_k^{\mathsf{T}} \mathbf{U}_k, \\ \text{subject to} \quad & G_{\mathrm{qp}}\mathbf{U}_k \le \mathbf{b}^{\mathrm{qp}}_k, \end{aligned} \tag{45}$$

with

$$\begin{aligned} H &= 2\left(\mathcal{M}_N^{\mathsf{T}} \overline{Q} \mathcal{M}_N + D_u^{\mathsf{T}} \overline{R} D_u\right), \\ \mathbf{c}_k &= 2\left[\mathcal{M}_N^{\mathsf{T}} \overline{Q}(\mathbf{f}_{N,k} - \mathbf{r}_k) - D_u^{\mathsf{T}} \overline{R} \mathbf{d}_k\right]. \end{aligned} \tag{46}$$

The current history, reference, previous protected fuel command, and empirical margins therefore enter the QP through $\mathbf{c}_k$ and $\mathbf{b}^{\mathrm{qp}}_k$. Because $q_u > 0$ and D_u is nonsingular, the fuel increment term makes H positive definite. The QP is strictly convex and has a unique optimizer whenever it is feasible. Prediction and QP matrices are rebuilt when a new model is accepted, as described in Section 3.4.

If the solution meets the prescribed optimality tolerances and fuel and predicted output constraints, its first command is sent through the original limit selector; otherwise, the supervisor activates PI fallback. The optimization repeats at the next sample using updated observations and the actual protected fuel history [39].

3.4 Online learning and asynchronous control algorithm

The EDMDc predictor and MPC formulation are integrated into an algorithm with separate learning and control tasks. Online learning adjusts selected model parameters within a subspace supported by the initial training data, while keeping the lifting map, normalization parameters, and output decoder fixed. The learning task fits candidates from eligible measured transitions and evaluates each frozen candidate using subsequently acquired observations. The control task uses the active predictor and checks completed candidates through constrained optimization with current feedback. Accepted model parameters, prediction caches, and empirical error states are committed together to maintain consistency between prediction and control. The original PI controller provides fallback when the control solution does not satisfy the acceptance conditions.

The exchange between prediction and learning in Fig. 3 represents the comparison of predicted and subsequently measured responses during candidate assessment. Parameter fitting uses measured transitions; model predictions are not substituted for the observed training targets.

Online learning estimates a structured correction to the initial model from measured transitions. Let $Q \in \mathbb{R}^{(n_z+3)\times r}$ be an orthonormal basis constructed from the training regressors, where n_z is the lifted dimension and r is the correction dimension. The trainable parameters are restricted to

$$\Theta_{\mathcal{L}} = \Theta_{0,\mathcal{L}} + DQ^{\mathsf{T}}, \qquad D \in \mathbb{R}^{|\mathcal{L}|\times r}. \tag{47}$$

For an eligible transition observed at instant k, the fitting variables are

$$\begin{aligned} \boldsymbol{\omega}_{k-1} &= \begin{bmatrix} \mathbf{z}_{k-1} \\ \widetilde{u}_{k-1} \\ \widetilde{\boldsymbol{\xi}}_{k-1} \end{bmatrix}, \\ \boldsymbol{\phi}_k &= Q^{\mathsf{T}}\boldsymbol{\omega}_{k-1}, \\ \mathbf{b}_k &= \mathbf{z}_{k,\mathcal{L}} - \Theta_{0,\mathcal{L}}\boldsymbol{\omega}_{k-1}. \end{aligned} \tag{48}$$

The inputs are the previous protected fuel command and measured flight conditions, so fitting uses only information available at instant k. Reindexing the fitted transitions by n, the correction minimizes

$$D_n = \arg\min_{D} \left\{ \lambda_{\text{prior}} \|D\|_F^2 + \sum_{\ell=1}^{n} \lambda_f^{n-\ell} \left\| \mathbf{b}_\ell - D\boldsymbol{\phi}_\ell \right\|_2^2 \right\}, \tag{49}$$

where $0 < \lambda_f \leq 1$ is the forgetting factor and $\lambda_{\text{prior}} > 0$ is a persistent regularization weight. Exponential weighting emphasizes recent fitted transitions [40]. A square root QR update replenishes the prior contribution as old data are discounted, preventing regularization in weakly excited directions from vanishing [35]. The regularized regressor information matrix is initialized as $H_0 = \lambda_{\text{prior}} I_r$. Only fitted transitions advance n;

samples reserved for candidate assessment do not enter the loss. The remaining model rows retain the prescribed history shifts and constant coordinate.

The one-step objective in Eq. 49 provides an efficient recursive candidate generator; it does not minimize the offline rollout objective in Eq. 29. Candidate acceptance instead depends on free-running prediction errors. After fitting is frozen, the candidate and active models are propagated from common measured histories using the rollout recursion in Eq. 26, augmented by the flight-condition forcing in Eq. 31, and identical subsequently recorded protected fuel commands, altitude, and Mach number. All scored transitions lie after the fitting cutoff, and a score is formed only when the complete validation interval is available. Neither model receives intermediate measurement corrections within a prediction interval. For the horizon set $\mathcal{H}$, $E^M_{j,h}$ is the RMSE over all prediction steps from 1 through h, normalized by s_j, where M denotes the frozen model. The aggregate score is $S_M = \sqrt{\frac{1}{4}\sum_{j=1}^{4}(E^M_{j,H_{\max}})^2}$, with $H_{\max} = \max \mathcal{H}$. Acceptance requires aggregate improvement and bounded per-output degradation. The implementation also requires improvement in the root mean square of the N_{1c} prefix errors across $\mathcal{H}$, with a separate regression guard at each horizon. All recorded inputs arrive causally during this assessment.

Algorithm 2 summarizes the fitting, subsequent validation, and control tasks.

Algorithm 2 Online training and control of EDMDc-MPC

Require: $M_0, Q, H_0, \lambda_f, N_f, N_v$; MPC settings and acceptance thresholds

$(M_v, v, n) \leftarrow (M_0, 0, 0)$; initialize the estimator with H_0

$\mathcal{J}_n$: objective in Eq. 49

Background learning (asynchronous)

for each learning cycle **do**

  Fit N_f eligible transitions by square root QR recursion; $n \leftarrow n + N_f$

  $D_c \leftarrow \arg\min_D \mathcal{J}_n(D)$

  Form a correction to $[A\ B\ E]_{\mathcal{L}}$ using Eq. 47

  Backtrack the dynamic correction; retain fixed rows and screen stability and relative change

  Keep the supported forcing-column update when it passes the same screens

  Hold D_c and fitting statistics fixed; collect N_v subsequent validation pairs

  Propagate M_c and M_v from common histories using Eq. 31

  Use the same recorded protected commands and flight conditions; do not reset states

  Compute normalized prefix errors $E^M_{j,h}$, aggregate scores S_M, and N_{1c} scores

  if aggregate and N_{1c} improvements and all component and horizon guards pass **then**

    Prepare prediction matrices and margins using Eqs. 35 and 43

    Queue the candidate package with its source version and data time

  end if

end for

Control task

for each sampling instant k **do**

  $\mathbf{z}_k \leftarrow \boldsymbol{\psi}(\mathbf{x}_k)$; use the active model's stored margins

  $(M_{\text{trial}}, v_{\text{trial}}) \leftarrow (M_v, v), \quad \mathbf{U}^\star \leftarrow \emptyset$

  if MPC is permitted and a consistent candidate has valid source version and causal data order **then**

    Solve Eq. 45 with M_c and current feedback

    if the candidate solution qualifies **then**

      $(M_{\text{trial}}, v_{\text{trial}}) \leftarrow (M_c, v_c), \quad \mathbf{U}^\star \leftarrow \mathbf{U}^\star_c$

    end if

  end if

  if $\mathbf{U}^\star = \emptyset$ and MPC is permitted **then**

    Solve Eq. 45 with M_v; retain $\mathbf{U}^\star$ only if qualified

  end if

  if $\mathbf{U}^\star \neq \emptyset$ and any enabled wall-clock deadline check passes **then**

    Commit the trial model, prediction matrices, margins, and version together

    $v \leftarrow v_{\text{trial}}, \quad u_{f,k} \leftarrow [\mathbf{U}^\star]_1$

  else

    Retain M_v; obtain $u_{f,k}$ from guarded PI control

  end if

  Apply $u_{f,k}$ through the original limit selector

  Queue the measured transition with its time and eligibility markers

end for

4 Simulation experiments and results

The experiments evaluate multi-horizon refinement across sampled flight conditions and the use of the resulting predictor in closed-loop control. Section 4.1 defines the simulation and comparison protocol. Section 4.2 presents the identification data, dictionary comparison, refinement results, and longer free-running predictions. Section 4.3 examines tracking and fuel actions for two command profiles under continuously varying flight conditions, followed by model-update diagnostics and measured execution times.

4.1 Simulation setup and evaluation protocol

The AGTF30 engine model [1] and its T-MATS dependencies [29] are simulated in MATLAB R2025a and Simulink normal mode on an AMD Ryzen 9 7945HX CPU with 32 GB RAM. The in-house primal–dual predictor–corrector QP solver uses at most 60 iterations, with tolerances of 10^{-8} for scaled primal feasibility, dual residuals, and complementarity. The sampling period is $T_s = 0.015\,\mathrm{s}$, and each trajectory begins with a 9.99 s warm-up. The original limit selector, fuel actuator dynamics, PI gains, and variable bleed valve and fan nozzle schedules are retained. Only the protected fuel command is optimized. Reported fuel, altitude, temperature, and pressure use kg/s, km, K, and kPa, respectively; the simulator retains its native NASA units, with conversions applied to reported values. The corrected fan speed N_{1c} is the tracking output; N_3 is a coupled high-pressure shaft response with an upper limit and has no independent tracking reference.

The identification dataset contains eight training, three validation, and two test trajectories, each with a 1500 s collection interval after warm-up. The collected altitude and Mach number extend from 0 to 12.192 km and from 0 to 0.8, respectively. These ranges describe the sampled operating profiles; they do not imply uniform coverage of the full flight envelope. Complete trajectories are assigned to the three splits before history windows are formed. Model normalization, input bounds, and operating-domain scales are obtained from training data. Fitting pairs require complete histories and a valid next sample within one uninterrupted active training segment. Candidate origins are taken every five samples, then 12000 pairs are selected at evenly spaced indices of the pooled list; M denotes this fitting subset. Table 1 reports the configuration shared by both closed-loop experiments. Here, ϵ_S and ϵ_N denote the minimum relative reductions in the aggregate and N_{1c} candidate-assessment scores. The ratios β_y and β_N bound candidate-to-active errors for each output and for N_{1c} at each validation horizon, respectively. Each candidate is assessed over $N_v = 100$ subsequent samples (1.5 s).

Two controllers are compared: PI with common command limits and EDMDc-MPC. Here, EDMDc-MPC denotes the complete framework in Section 3, including online candidate estimation, validation, and adoption. Within each experiment, the controllers use the same initial engine state, reference-generation law, prescribed flight conditions, and command-amplitude and rate limits. Both experiments use the same initial EDMDc predictor and controller settings. The MPC prediction and control horizons correspond to 1.5 and 0.75 s, respectively. The current native fuel interval constrains the first optimized command. Native protection remains active after command selection, so its

Table 1 Identification and controller settings shared by the two control experiments.

Parameter	Value
T_s	$0.015\,\mathrm{s}$
(L, n_z)	$(40, 361)$
$\boldsymbol{\psi}$	Linear delay coordinates and constant
$(M, \lambda_{\mathrm{reg}})$	$(12000, 10^{-4})$
$\mathcal{H}_{\mathrm{id}}$	$\{50, 200, 1000\}$
$(w_H, \lambda_{\mathrm{roll}})$	$(1/3, 10^{-3})$
(N_p, N_c)	$(100, 50)$
(q_N, q_u)	$(1, 0.1)$
s_N	$413.1372\,\mathrm{rpm}$
s_f	$0.365300\,\mathrm{kg\,s^{-1}}$
$(u_{\min}, u_{\max})$	$(0.034779, 0.765378)\,\mathrm{kg\,s^{-1}}$
$\dot{u}_{\max}$	$1.020583\,\mathrm{kg\,s^{-2}}$
δ_u	$0.015309\,\mathrm{kg\,s^{-1}}$
$(\overline{N}_1, \overline{N}_3)$	$(2300, 22500)\,\mathrm{rpm}$
$N_{1c,\max}^{\mathrm{tr}}$	$2482.673\,\mathrm{rpm}$
$(\overline{T}_{45}, \overline{P}_{s3})$	$(1341.111\,\mathrm{K}, 5446.858\,\mathrm{kPa})$
γ	1.1
$(\lvert\mathcal{L}\rvert, r)$	$(8, 15)$
$(\lambda_f, \lambda_{\mathrm{prior}})$	$(0.995, 10)$
$(N_f, N_v, \mathcal{H})$	$(100, 100, \{10, 25, 50, 100\})$
(ϵ_S, ϵ_N)	$(2\%, 2\%)$
(β_y, β_N)	$(1.10, 1.02)$

numerical bounds depend on each controller's own engine trajectory. Neither experiment includes scheduled forced-fallback windows.

The repeated-command experiment lasts 240 s after warm-up. Four consecutive 60 s command cycles use power lever angle (PLA) targets of 50, 64, 72, 60, and 50 degrees, beginning at cycle-relative times 2.01, 12, 24, 36, and 48 s with 0.45 s ramps. The initial PLA is 42 degrees; subsequent cycles start from the preceding final level.

The second experiment lasts 75 s and combines ascending steps, descending steps, and large acceleration and deceleration commands in one continuous run. Starting from 42 degrees, PLA steps through 50, 58, 66, and 74 degrees at 5.010, 10.005, 15.000, and 19.995 s, then decreases through 66, 58, 50, and 42 degrees at 24.990, 30.000, 35.010, and 40.005 s. Each step transition lasts one sampling period. At 45 and 60 s, 1.995 s ramps command acceleration to 78 degrees and deceleration to 42 degrees, respectively. Altitude and Mach number vary linearly between the corresponding knots in Table 2. The original reference-generation law converts PLA and measured operating conditions into the N_{1c} reference in both experiments. These are prescribed engine boundary conditions rather than simulated aircraft flight-dynamics trajectories.

All tracking and fuel-variation metrics use the complete post-warm-up intervals, including handover and protection activity: 16001 samples for the repeated-command experiment and 5001 samples for the staircase experiment. With $e_k = N_{1c,k} - r_k$,

Table 2 Flight-condition knots for the two control experiments. Both time columns are measured after warm-up.

Repeated-command time (s)	Staircase-command time (s)	Altitude (km)	Mach number
0	0	6.096	0.50
60	18.75	3.048	0.30
180	56.25	9.144	0.65
240	75	6.096	0.50

$k = 0, \ldots, K$, and $K = 16000$ or 5000, respectively, the metrics are RMSE, peak absolute error, integral of absolute error (IAE), and fuel-command total variation (TV):

$$\begin{aligned} \mathrm{RMSE} &= \sqrt{\frac{1}{K+1}\sum_{k=0}^{K} e_k^2}, & E_{\max} &= \max_{0\le k\le K} |e_k|, \\ \mathrm{IAE} &= T_s\left(\frac{|e_0|+|e_K|}{2} + \sum_{k=1}^{K-1}|e_k|\right), & \mathrm{TV}_u &= \sum_{k=1}^{K}|u_k - u_{k-1}|. \end{aligned} \tag{50}$$

IAE uses trapezoidal integration. Fuel TV measures cumulative command movement, not fuel consumption. Each controller contributes one trajectory per experiment; the four repeated command cycles are not independent experimental replicates.

The engine-solver checks flag one PI sample at 189.825 s in the repeated-command experiment and one EDMDc-MPC sample at 4.065 s in the staircase experiment. The latter experiment also contains 90 PI samples above the T_{45} upper limit of 1341.111 K, with a maximum exceedance of 1.446 K. All flagged samples remain in the plots and metrics. These records limit numerical and operating-limit qualification of the comparisons. Control-cycle wall times are logged, but wall-clock deadline rejection is disabled in these offline simulations.

4.2 Prediction validation and multi-horizon refinement results

Figure 4 shows the collected flight conditions. The training set includes six fixed conditions and separate climb and descent profiles. Validation uses fixed conditions at (2.4384 km, 0.25), (6.7056 km, 0.55), and (11.2776 km, 0.78). The two test conditions are (5.4864 km, 0.45) and (10.0584 km, 0.70). They are distinct from the fixed training points. Samples failing the existing numerical or physical checks are excluded from fitting and window-based scoring; history and prediction windows cannot cross an excluded interval.

The collection schedule contains three successive 500 s stages. Random-amplitude PLA changes first excite transitions between operating levels. A frequency sweep from 0.005 to 0.08 Hz then provides responses at different time scales. Staircase increases and decreases followed by large acceleration and deceleration commands complete the interval. The scheduled PLA lies between 40 and 80 degrees, and an additional bounded fuel probe is applied through the existing protection chain. The recorded

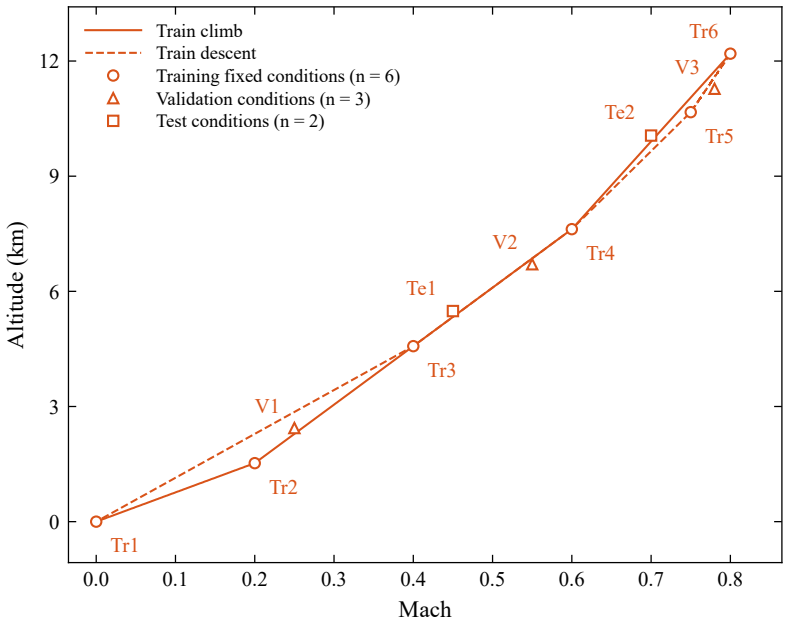


Fig. 4 Recorded altitude–Mach conditions of the identification dataset.

protected fuel command, rather than the requested perturbation before protection, is the identification input.

Figure 5 presents a representative training trajectory at 4.572 km and Mach 0.40. It shows how the fuel input excites the corrected fan speed, high-pressure shaft speed, and measured temperature over the full collection interval. The horizontal axis retains the recorded 0–1500 s duration after warm-up. For display, each signal is mapped using $(x - x_{\min})/(x_{\max} - x_{\min})$, where the extrema come only from valid active training samples. This display scaling is separate from the mean and standard-deviation normalization used for identification. The example is one trajectory within the wider training dataset, not the entire identification dataset. Of its 100001 active samples, 98539 pass the existing quality checks.

Five dictionaries are compared using the same history length and data splits: linear delay coordinates, current-observation squares, current-observation quadratic terms, radial basis functions, and Fourier functions. Their lifted dimensions are 361, 369, 397, 393, and 393, respectively. Each contains the delay coordinates and a constant. Squares add eight squared current observations; quadratic features include all 36 unique second-order products. Gaussian radial basis functions (RBFs) use 32 centers selected by deterministic farthest-point sampling of normalized training observations. The retained width is twice the median nearest-center distance, selected by validation from factors $\{0.5, 1, 2\}$. Fourier features use coordinatewise sine and cosine terms with periods twice the training ranges; validation retains harmonics 1 and 2 from one- and two-harmonic candidates. Ridge initialization is followed by multi-horizon refinement using equally weighted horizons of 50, 200, and 1000 steps (0.75, 3, and 15 s). The validation score

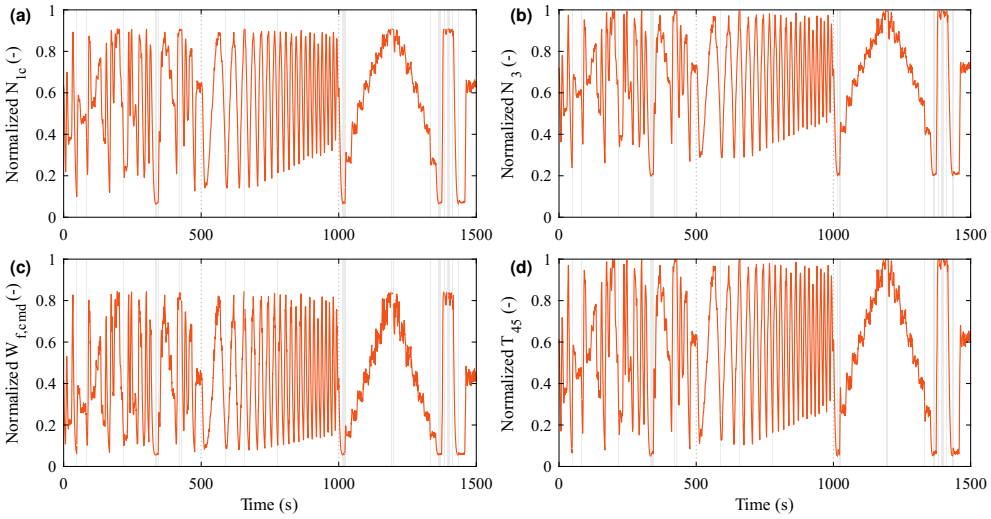


Fig. 5 Normalized input and output signals from a representative AGTF30 training trajectory at 4.572 km and Mach 0.40.

is the square root of the weighted mean normalized rollout loss in Eq. 28. Stable candidates are assessed after each accepted optimizer step, and the best validation iterate is retained subject to a per-horizon regression limit. The test trajectories do not select the retained iterate.

Validation selects the linear-delay model after multi-horizon refinement. Relative to its ridge initializer, the normalized multi-horizon validation score decreases from 0.13715 to 0.13379. Adding nonlinear features does not improve the best validation score in this dataset. The retained iterate is iteration 6 of a 40-step budget, rather than the last optimizer iterate. Ten unstable trial steps are rejected during this refinement. The retained dynamic spectral radius is 0.9999632; the full lifted matrix also contains the prescribed constant-coordinate eigenvalue at one. This numerical screen excludes growing dynamic modes within its tolerance and does not establish global nonlinear stability.

Table 3 compares the retained refined models on 128 common valid test origins at the 1.5 s MPC horizon and a longer 15 s horizon. Each forecast is initialized from the measured history and then propagated using recorded inputs without intermediate measurement corrections. For output j, the reported percentage is $\mathrm{NRMSE}_j(h) = 100\,\mathrm{RMSE}_j(h)/R_j^{\mathrm{tr}}$, where $R_j^{\mathrm{tr}} = y_{j,\max}^{\mathrm{tr}} - y_{j,\min}^{\mathrm{tr}}$ is its valid training range and $\mathrm{RMSE}_j(h)$ is the error at forecast step h across the common origins. The same training range is used for all dictionaries and horizons. This display normalization matches the trajectory figures and is separate from the standard-deviation scaling in the identification loss.

The lowest short-horizon error depends on the output: Fourier features give the lowest N_{1c} error, radial basis functions the lowest N_3 error, quadratic features the lowest P_{s3} error, and linear delay coordinates the lowest T_{45} error. At 15 s, the linear-delay model gives the lowest error for all four outputs. Thus, nonlinear features can

Table 3 Test endpoint prediction errors for the validation-retained refined models. All entries are training-range normalized RMSE (%); bold values are the lowest in each output column at the stated horizon.

Dictionary	N_{1c}	N_3	T_{45}	P_{s3}
Prediction horizon: 1.5 s (100 steps)				
Linear delay	1.312	1.159	**2.762**	0.665
Current squares	1.247	1.164	2.921	0.639
Current quadratic	1.168	1.121	2.907	**0.615**
Radial basis	1.173	**1.111**	2.767	0.637
Fourier	**0.954**	1.135	3.028	0.622
Prediction horizon: 15 s (1000 steps)				
Linear delay	**6.900**	**5.874**	**6.813**	**1.588**
Current squares	7.369	6.554	6.846	1.650
Current quadratic	7.006	5.965	7.011	1.668
Radial basis	7.058	5.954	6.874	1.659
Fourier	7.125	6.233	7.098	1.653

All models use the same 128 valid test origins and recorded inputs, with no intermediate measurement corrections. Errors are evaluated at the forecast endpoint; they are not full-trajectory errors over 1500 s.

improve individual short-horizon predictions, but do not consistently improve longer propagation in this dataset. These test errors characterize the models selected by validation and are not used to change the selected dictionary or refinement iterate.

Figures 6 and 7 assess the selected frozen model over the complete 1500 s test intervals. Each prediction is initialized once using the warm-up history and then propagated for 100000 consecutive steps with the recorded protected fuel commands, altitude, and Mach number. Subsequent measured outputs do not correct the state, and there are no state resets, parameter updates, or intermediate re-lifting operations. Prediction continues across quality-flagged intervals; only valid forecast–reference pairs contribute to the error statistics. The initialization sample is excluded from scoring. In these figures, the superscript n denotes the display scaling $y_j^{\mathrm{n}} = (y_j - y_{j,\min}^{\mathrm{tr}})/R_j^{\mathrm{tr}}$, applied identically to measured and predicted outputs.

The model reproduces the broad response patterns but exhibits substantial accumulated error over this long interval. The N_{1c} RMSE is 84.53 rpm at the first test condition and 168.06 rpm at the second; the corresponding N_3 values are 301.89 and 625.95 rpm. All predicted samples remain finite. Finite propagation should not be interpreted as accurate reconstruction of every transient or validation of the physical reference samples excluded by the quality checks.

Figure 8 evaluates the change in prediction accuracy produced by multi-horizon refinement and validation-based selection. The ridge initializer and refined model use the same linear-delay dictionary, preprocessing, 128 common valid test origins, and recorded inputs, with the training-range normalization defined for Table 3. Each forecast is initialized separately and propagated without resets within its 15 s window. At the 1.5 s MPC horizon, the refined-model normalized errors for N_{1c}, N_3, T_{45}, and

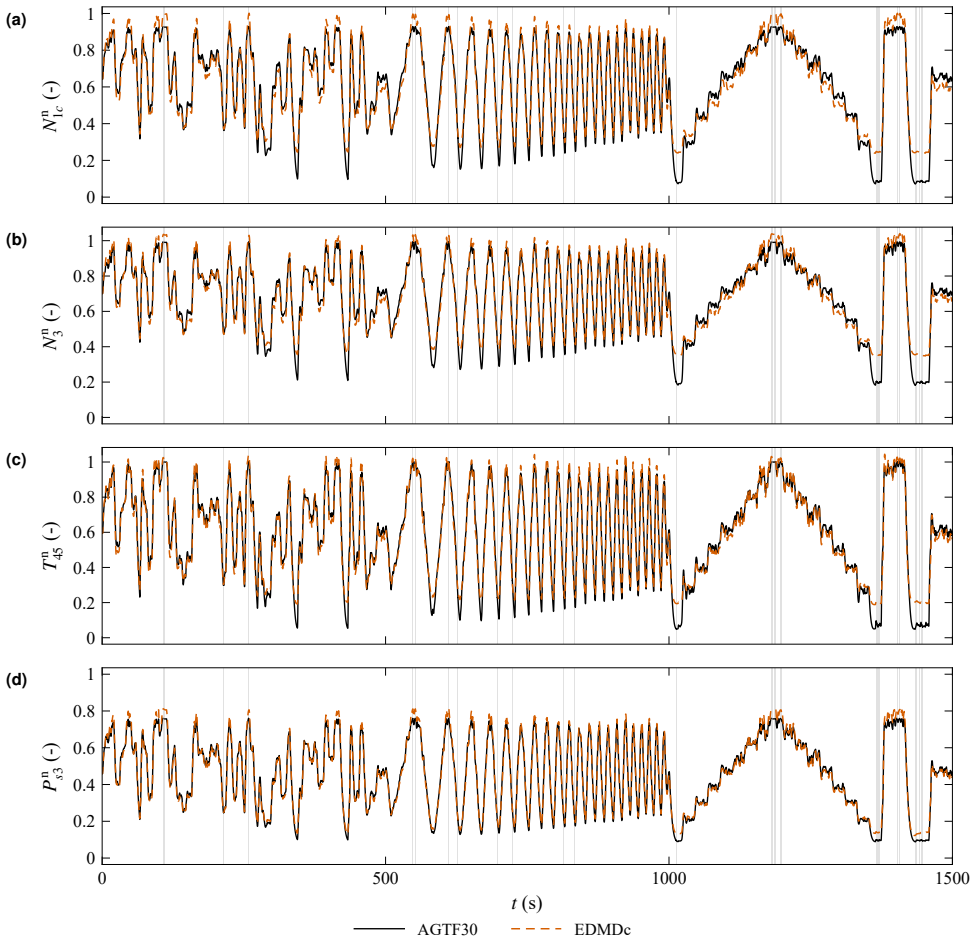


Fig. 6 Single-initialization prediction by the selected refined linear-delay model at 5.4864 km and Mach 0.45.

P_{s3} are 1.312%, 1.159%, 2.762%, and 0.665%, respectively. Errors generally increase over the early horizons but need not grow monotonically: each point measures error at that forecast endpoint, not cumulative error. Multi-horizon refinement reduces many longer-horizon errors, although some temperature errors remain higher. Separately, the normalized test prefix score decreases from 0.21807 to 0.20978 at 15 s and increases slightly from 0.013665 to 0.013730 at 0.75 s. The observed benefit therefore depends on the prediction horizon and output. This comparison evaluates the refined model against its ridge initializer; a comparison with single-horizon refinement would be needed to isolate the effect of combining several training horizons.

The test trajectories were inspected during method development. Their separation from fitting and final validation selection is maintained, but these results should not be

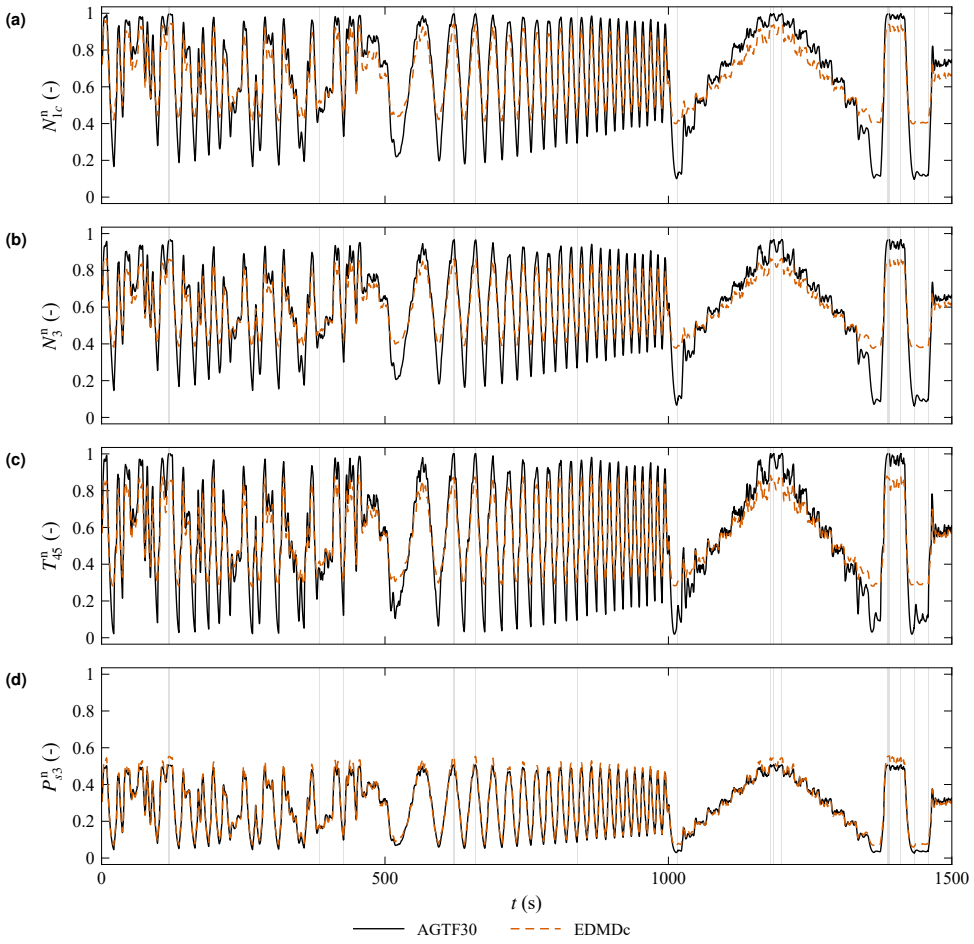


Fig. 7 Single-initialization prediction at 10.0584 km and Mach 0.70 using the same frozen model and propagation protocol as Fig. 6.

described as a new blind evaluation. The observed prediction errors and their growth with horizon motivate assessing the predictor within the complete receding-horizon control loop.

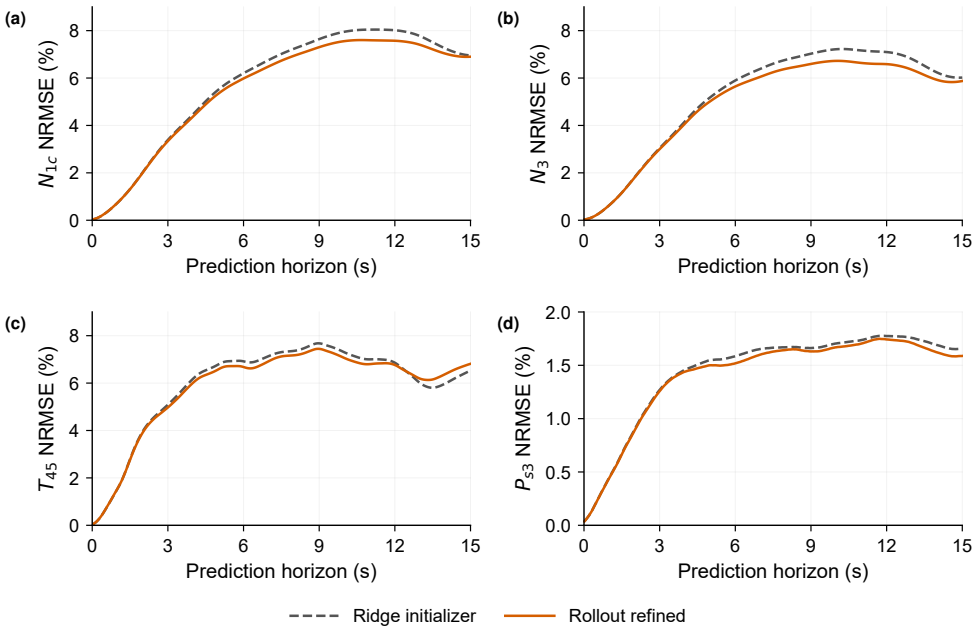


Fig. 8 Endpoint prediction errors before and after multi-horizon refinement of the linear-delay EDMDc model. The ridge initializer and refined model are evaluated on 128 common valid test origins; errors are training-range normalized RMSE.

4.3 Closed-loop MPC performance across flight conditions

Figure 9 compares PI and EDMDc-MPC during the 240 s repeated-command experiment. EDMDc-MPC reduces the aggregate N_{1c} tracking error over the repeated command transitions and varying flight conditions. The N_3 panel records the associated high-pressure shaft response; it does not represent a second independently tracked output.

Table 4 Descriptive tracking and fuel-command metrics over each complete control experiment.

Time	Controller	RMSE (rpm)	IAE (rpm s)	Peak error (rpm)	Fuel TV (kg s^{-1})
240 s	PI	97.106	10668.40	445.186	2.954
	EDMDc-MPC	64.926	4924.83	408.714	8.026
75 s	PI	298.805	13103.28	1030.618	1.962
	EDMDc-MPC	256.499	9901.80	954.813	4.658

One trajectory per controller and experiment; all 16001 and 5001 post-warm-up samples are retained, respectively, including numerical and temperature-limit flags. Fuel TV excludes warm-up.

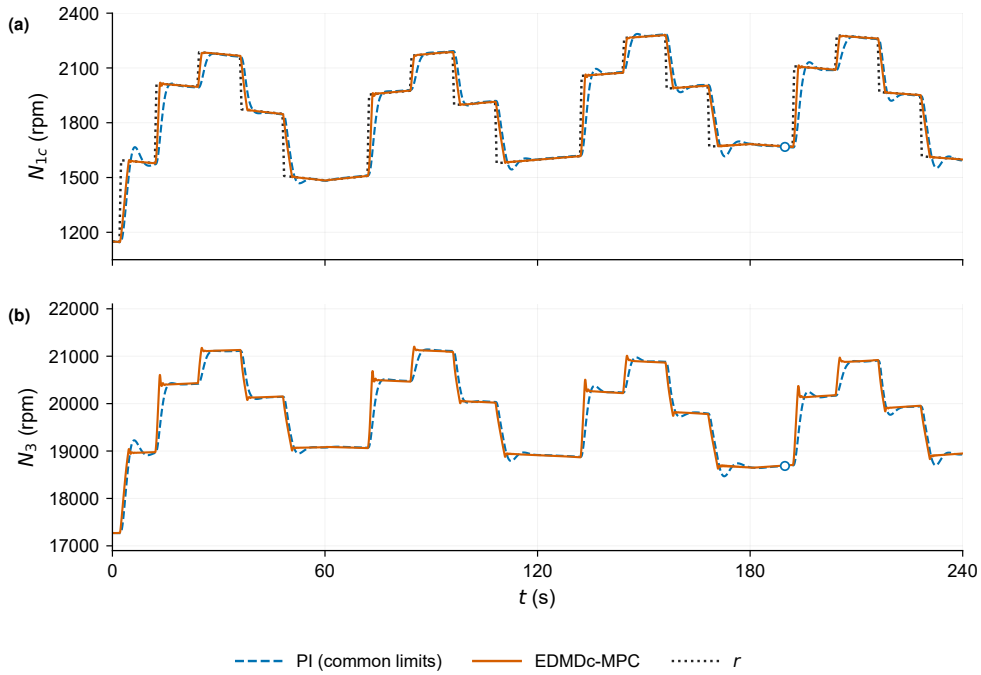


Fig. 9 Closed-loop rotor responses of PI and EDMDc-MPC during the complete 240 s repeated-command experiment, with t measured after warm-up.

Table 4 quantifies the tracking–actuation tradeoff. In the repeated-command experiment, EDMDc-MPC reduces RMSE from 97.106 to 64.926 rpm and IAE from 10668.40 to 4924.83 rpm s relative to PI. Peak error decreases from 445.186 to 408.714 rpm. These tracking improvements are accompanied by greater fuel-command variation: fuel TV increases from 2.954 to 8.026 kg s^{-1}. EDMDc-MPC directly applies an MPC command at 99.881% of the evaluated samples; the remaining samples are associated with PI fallback or handover. No QP failures are recorded. The comparison evaluates the complete control architecture and does not isolate the contribution of offline multi-horizon refinement or subsequent online model updates.

Figure 10 compares fuel commands with their native protection bounds. The applied commands also satisfy the common command-amplitude limits and the rate bound referenced to the previous protected command. EDMDc-MPC reaches the common rate limit of 1.0206 kg s^{-2}, while PI reaches approximately 0.2161 kg s^{-2}. This difference is consistent with the higher MPC fuel TV. Actual fuel flow retains its actuator dynamics and is distinct from the commanded input.

Figure 11 presents the ascending and descending staircase followed by large acceleration and deceleration. For this 75 s experiment, EDMDc-MPC reduces N_{1c} RMSE from 298.805 to 256.499 rpm and IAE from 13103.28 to 9901.80 rpm s, while peak error

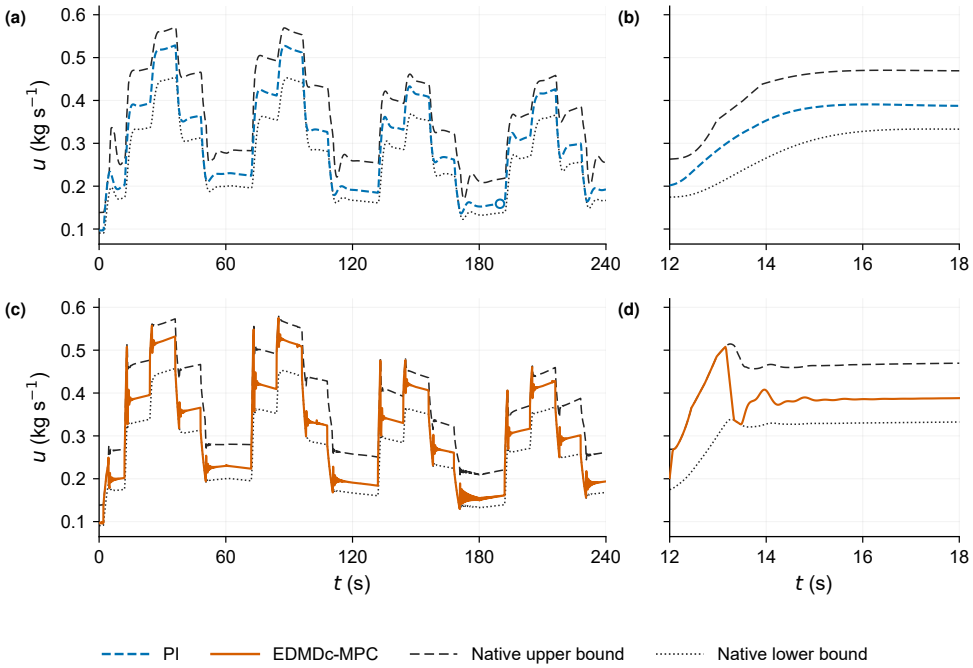


Fig. 10 Protected fuel commands u and native protection bounds during the 240 s repeated-command experiment. Panels (a,b) show PI and (c,d) show EDMDc-MPC.

decreases from 1030.618 to 954.813 rpm. Both controllers exhibit appreciable transient tracking errors during the larger reference changes. The lower panel shows the accompanying N_3 response over the same run.

Figure 12 shows the fuel commands from the same 75 s runs, including an enlargement around the large acceleration at 45 s. Recorded commands satisfy the native protection bounds and the common amplitude and rate limits. Fuel TV rises from 1.962 for PI to 4.658 kg s^{-1} for EDMDc-MPC, which again associates lower tracking error with greater control movement. EDMDc-MPC directly applies MPC commands at 99.620% of the evaluated samples, with no recorded QP failures. Input-bound compliance does not by itself imply compliance with every engine-output limit, as indicated by the PI temperature-limit exceedances reported in Section 4.1.

Model-update diagnostics further characterize the controller during the 240 s repeated-command experiment. EDMDc-MPC completes 79 candidate assessments and adopts 16 updates, with every assessment origin following its fitting cutoff. A separate prediction check uses 160 common valid origins at each of 10, 25, 50, and 100 steps. At each origin, the initial and then-active models are frozen, initialized from identical measurement histories, and propagated with the same recorded protected fuel commands and flight conditions, without intermediate output feedback. These checks do not enter candidate selection. At 100 steps (1.5 s), the initial and active models

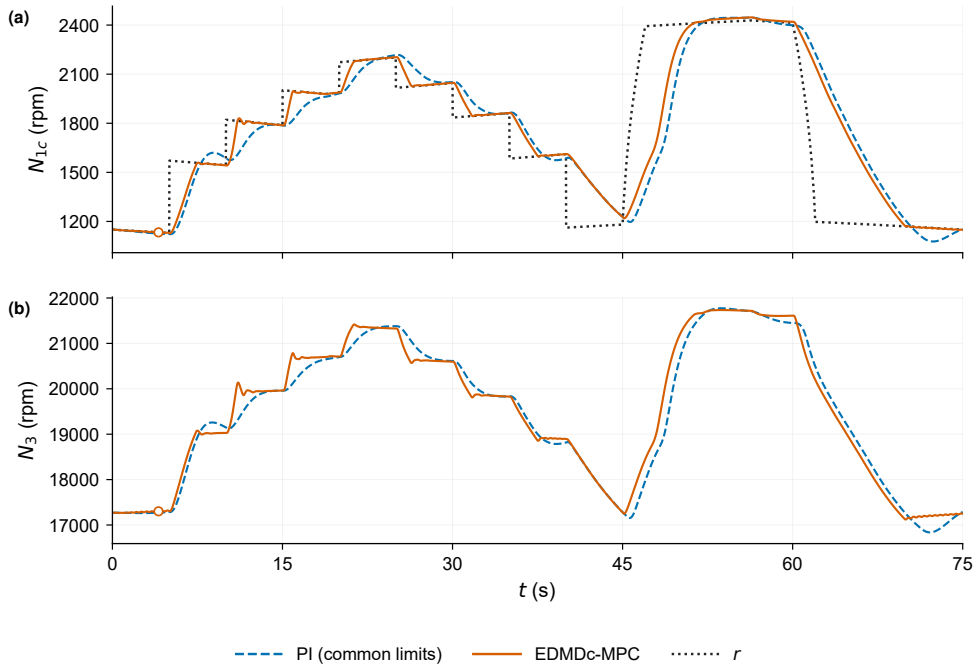


Fig. 11 Closed-loop rotor responses of PI and EDMDc-MPC during the 75 s staircase and large acceleration/deceleration experiment.

give N_{1c} RMSE values of 16.77 and 16.97 rpm, respectively; active-model errors are also slightly higher at the other horizons. Thus, these diagnostics do not establish an aggregate N_{1c} prediction benefit from online updates. The tracking improvements reported above characterize the complete EDMDc-MPC controller rather than the isolated contribution of its updates.

Computational cost is evaluated over the same 16001 logged control cycles per controller. The PI median, 95th-percentile, and maximum wall times are 0.033, 0.056, and 0.280 ms, respectively; the corresponding EDMDc-MPC values are 4.522, 6.850, and 27.591 ms. One EDMDc-MPC cycle exceeds the 15 ms sampling period. These measurements cover the control callback and exclude the total processor demand of asynchronous background learning. With wall-clock deadline rejection disabled, they indicate typical computational feasibility in this simulation environment but do not establish hard real-time compliance. The comparisons remain limited to the sampled profiles, given the retained numerical flags, reuse of test trajectories during development, and absence of independent repeated runs.

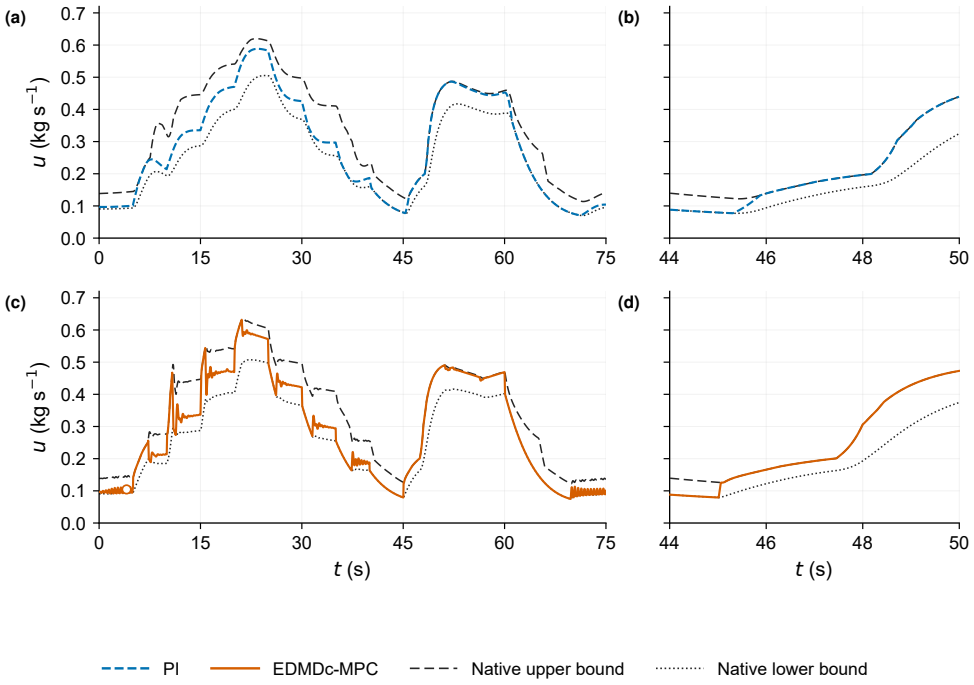


Fig. 12 Protected fuel commands u and native protection bounds during the 75 s staircase and large acceleration/deceleration experiment.

5 Conclusions

This study developed multi-horizon refinement of EDMDc for aeroengine MPC across multiple flight conditions. The identification procedure refines a regularized one-step estimate using normalized free-running errors over several horizons, while preserving the prescribed history shifts and constant coordinate. Validation-based model selection and spectral screening determine the retained predictor. Measurement histories, protected fuel commands, and explicit flight-condition forcing provide the coordinates and inputs used for prediction.

The linear-delay model was selected from five dictionaries. Multi-horizon refinement reduced its normalized validation score from 0.13715 to 0.13379 and its normalized test prefix score at 15 s from 0.21807 to 0.20978. The short-horizon prefix score increased slightly, and some temperature errors remained higher, so the improvements depend on the horizon and output. Single-initialization tests over 1500 s also exposed accumulated prediction errors beyond the MPC horizon.

The refined predictor was integrated into constrained MPC with validated online updates, native engine protection, and PI fallback. Relative to PI with common command limits, the complete controller reduced corrected-fan-speed RMSE by 33.1% in the 240 s repeated-command experiment and 14.2% in the 75 s staircase and large

acceleration/deceleration experiment. Both experiments also reduced IAE with greater fuel-command variation. These results characterize the complete controller over the sampled profiles; they do not isolate the control benefit of offline refinement or online updates.

The prediction comparison establishes the changes relative to the ridge initializer. Isolating the value of combining horizons requires a comparison with single-horizon refinement under matched settings. Section 4.3 details the numerical, timing, and online-update limitations. Future work should include controlled comparisons of horizon selection, offline refinement, and online updating, together with fresh trajectories and independent runs on a target control platform to assess generalization and real-time execution.

References

[1] Chapman, J.W., Litt, J.S.: Control design for an advanced geared turbofan engine. In: 53rd AIAA/SAE/ASEE Joint Propulsion Conference. American Institute of Aeronautics and Astronautics, Reston, VA (2017). https://doi.org/10.2514/6.2017-4820 . AIAA Paper 2017-4820

[2] Csank, J., May, R., Litt, J., Guo, T.-H.: Control design for a generic commercial aircraft engine. In: 46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit. American Institute of Aeronautics and Astronautics, Reston, VA (2010). https://doi.org/10.2514/6.2010-6629 . AIAA Paper 2010-6629

[3] Liu, X., Luo, C., Xiong, L.: Compensators design for bumpless switching in aero-engine multi-loop control system. Asian Journal of Control **24**(5), 2665–2678 (2022) https://doi.org/10.1002/asjc.2677

[4] Qin, S.J., Badgwell, T.A.: A survey of industrial model predictive control technology. Control Engineering Practice **11**(7), 733–764 (2003) https://doi.org/10.1016/s0967-0661(02)00186-7

[5] Mayne, D.Q.: Model predictive control: Recent developments and future promise. Automatica **50**(12), 2967–2986 (2014) https://doi.org/10.1016/j.automatica.2014.10.128

[6] Mu, J., Rees, D., Liu, G.P.: Advanced controller design for aircraft gas turbine engines. Control Engineering Practice **13**(8), 1001–1015 (2005) https://doi.org/10.1016/j.conengprac.2004.11.001

[7] Richter, H., Singaraju, A.V., Litt, J.S.: Multiplexed predictive control of a large commercial turbofan engine. Journal of Guidance, Control, and Dynamics **31**(2), 273–281 (2008) https://doi.org/10.2514/1.30591

[8] Du, X., Sun, X.-M., Wang, Z.-M., Dai, A.-N.: A scheduling scheme of linear model predictive controllers for turbofan engines. IEEE Access **5**, 24533–24541 (2017) https://doi.org/10.1109/access.2017.2764076

[9] Montazeri-Gh, M., Rasti, A., Jafari, A., Ehteshami, M.: Design and implementation of MPC for turbofan engine control system. Aerospace Science and Technology **92**, 99–113 (2019) https://doi.org/10.1016/j.ast.2019.05.061

[10] Chen, Q., Sheng, H., Zhang, T.: A novel direct performance adaptive control of aero-engine using subspace-based improved model predictive control. Aerospace Science and Technology **128**, 107760 (2022) https://doi.org/10.1016/j.ast.2022.107760

[11] Brunton, S.L., Budišić, M., Kaiser, E., Kutz, J.N.: Modern Koopman theory for dynamical systems. SIAM Review **64**(2), 229–340 (2022) https://doi.org/10.1137/21m1401243

[12] Williams, M.O., Kevrekidis, I.G., Rowley, C.W.: A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition. Journal of Nonlinear Science **25**(6), 1307–1346 (2015) https://doi.org/10.1007/s00332-015-9258-5

[13] Williams, M.O., Rowley, C.W., Kevrekidis, I.G.: A kernel-based method for data-driven Koopman spectral analysis. Journal of Computational Dynamics **2**(2), 247–265 (2015) https://doi.org/10.3934/jcd.2015005

[14] Li, Q., Dietrich, F., Bollt, E.M., Kevrekidis, I.G.: Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator. Chaos: An Interdisciplinary Journal of Nonlinear Science **27**(10), 103111 (2017) https://doi.org/10.1063/1.4993854

[15] Lusch, B., Kutz, J.N., Brunton, S.L.: Deep learning for universal linear embeddings of nonlinear dynamics. Nature Communications **9**(1), 4950 (2018) https://doi.org/10.1038/s41467-018-07210-0

[16] Proctor, J.L., Brunton, S.L., Kutz, J.N.: Dynamic mode decomposition with control. SIAM Journal on Applied Dynamical Systems **15**(1), 142–161 (2016) https://doi.org/10.1137/15m1013857

[17] Proctor, J.L., Brunton, S.L., Kutz, J.N.: Generalizing Koopman theory to allow for inputs and control. SIAM Journal on Applied Dynamical Systems **17**(1), 909–930 (2018) https://doi.org/10.1137/16m1062296

[18] Korda, M., Mezić, I.: Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. Automatica **93**, 149–160 (2018) https://doi.org/10.1016/j.automatica.2018.03.046

[19] Grasev, D.: Koopman-based nonlinear identification and model predictive control of a turbofan engine. IET Control Theory & Applications **20**(1) (2026) https://doi.org/10.1049/cth2.70169 . Article e70169

[20] Hemati, M.S., Williams, M.O., Rowley, C.W.: Dynamic mode decomposition for large and streaming datasets. Physics of Fluids **26**(11), 111701 (2014) https:

//doi.org/10.1063/1.4901016

[21] Zhang, H., Rowley, C.W., Deem, E.A., Cattafesta, L.N.: Online dynamic mode decomposition for time-varying systems. SIAM Journal on Applied Dynamical Systems **18**(3), 1586–1609 (2019) https://doi.org/10.1137/18m1192329

[22] Calderon, H.M., Schulz, E., Oehlschlagel, T., Werner, H.: Koopman operator-based model predictive control with recursive online update. In: 2021 European Control Conference (ECC), pp. 1543–1549. IEEE, Piscataway, NJ (2021). https://doi.org/10.23919/ecc54610.2021.9655220

[23] Han, M., Wong, K., Law, A.W.-K., Yin, X.: MAKO: Meta-adaptive Koopman operators for learning-based model predictive control of parametrically uncertain nonlinear systems. Automatica **186**, 112827 (2026) https://doi.org/10.1016/j.automatica.2026.112827

[24] Yan, H., Yan, H., Peng, Y., Liu, Q., Lv, Y., Rao, K.: Robust tracking MPC for nonlinear constrained systems via online-updated Koopman operators. Automatica **190**, 113069 (2026) https://doi.org/10.1016/j.automatica.2026.113069

[25] Hewing, L., Wabersich, K.P., Menner, M., Zeilinger, M.N.: Learning-based model predictive control: Toward safe learning in control. Annual Review of Control, Robotics, and Autonomous Systems **3**(1), 269–296 (2020) https://doi.org/10.1146/annurev-control-090419-075625

[26] Aswani, A., Gonzalez, H., Sastry, S.S., Tomlin, C.: Provably safe and robust learning-based model predictive control. Automatica **49**(5), 1216–1226 (2013) https://doi.org/10.1016/j.automatica.2013.02.003

[27] Lorenzen, M., Cannon, M., Allgöwer, F.: Robust MPC with recursive model update. Automatica **103**, 461–471 (2019) https://doi.org/10.1016/j.automatica.2019.02.023

[28] Wang, M., Lou, X., Cui, B.: Learning-based robust model predictive control with data-driven Koopman operators. International Journal of Machine Learning and Cybernetics **14**(9), 3295–3321 (2023) https://doi.org/10.1007/s13042-023-01834-5

[29] Chapman, J.W., Lavelle, T.M., May, R., Litt, J.S., Guo, T.-H.: Propulsion system simulation using the toolbox for the modeling and analysis of thermodynamic systems (T-MATS). In: 50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference. American Institute of Aeronautics and Astronautics, Reston, VA (2014). https://doi.org/10.2514/6.2014-3929 . AIAA Paper 2014-3929

[30] Mezić, I.: Spectral properties of dynamical systems, model reduction and decompositions. Nonlinear Dynamics **41**(1–3), 309–325 (2005) https://doi.org/10.1007/s11071-005-2824-x

[31] Budišić, M., Mohr, R., Mezić, I.: Applied Koopmanism. Chaos: An Interdisciplinary Journal of Nonlinear Science **22**(4), 047510 (2012) https://doi.org/10.1063/1.4772195

[32] Brunton, S.L., Brunton, B.W., Proctor, J.L., Kutz, J.N.: Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. PLOS ONE **11**(2) (2016) https://doi.org/10.1371/journal.pone.0150171 . Article e0150171

[33] Korda, M., Mezić, I.: On convergence of extended dynamic mode decomposition to the Koopman operator. Journal of Nonlinear Science **28**(2), 687–710 (2018) https://doi.org/10.1007/s00332-017-9423-0

[34] Arbabi, H., Mezić, I.: Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator. SIAM Journal on Applied Dynamical Systems **16**(4), 2096–2126 (2017) https://doi.org/10.1137/17m1125236

[35] Björck, Å.: Numerical Methods for Least Squares Problems. Society for Industrial and Applied Mathematics, Philadelphia, PA (1996). https://doi.org/10.1137/1.9781611971484

[36] Baddoo, P.J., Herrmann, B., McKeon, B.J., Kutz, J.N., Brunton, S.L.: Physics-informed dynamic mode decomposition. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **479**(2271), 20220576 (2023) https://doi.org/10.1098/rspa.2022.0576

[37] Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Mathematical Programming **45**, 503–528 (1989) https://doi.org/10.1007/BF01589116

[38] Langson, W., Chryssochoos, I., Raković, S.V., Mayne, D.Q.: Robust model predictive control using tubes. Automatica **40**(1), 125–133 (2004) https://doi.org/10.1016/j.automatica.2003.08.009

[39] Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica **36**(6), 789–814 (2000) https://doi.org/10.1016/s0005-1098(99)00214-9

[40] Bruce, A.L., Goel, A., Bernstein, D.S.: Convergence and consistency of recursive least squares with variable-rate forgetting. Automatica **119**, 109052 (2020) https://doi.org/10.1016/j.automatica.2020.109052