

POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Learning-Based Model Predictive Control for Active Noise Control Systems

Tesi di Laurea Magistrale in
Mechanical Engineering - Ingegneria Meccanica

Author: **Mahshad Rastegarmoghaddam**

Student ID: 247309
Advisor: Prof. Hamid Reza Karimi
Co-advisors: Mohsen Jalaeian-Farimani, Chao Liang
Academic Year: 2025-26

Abstract

Active noise control is effective at low frequencies, where passive treatments are often impractical. However, its performance can degrade under non-stationary disturbances, nonlinear acoustic effects, and uncertainty in the plant dynamics. Adaptive algorithms such as FxNLMS require convergence and can be sensitive to step-size selection, secondary-path uncertainty, and changes in the disturbance. Model-based methods, instead, depend on identified acoustic models that may vary in realistic applications. In particular, model predictive control (MPC) relies on accurate primary- and secondary-path models to predict the future residual error. This thesis addresses these limitations by developing a Hybrid MPC–LSTM framework to compensate for nonlinear and history-dependent model mismatch.

The proposed method retains a linear state-space MPC controller and uses a causal long short-term memory (LSTM) network to predict the lumped model-mismatch sequence over the MPC horizon. The complete predicted sequence is used to correct the lifted MPC prediction model, while its first element is used to compensate the state observer. During closed-loop operation, the LSTM predictor is used with fixed offline-trained parameters. Six recurrent, convolutional, and attention-based sequence architectures were benchmarked under a common protocol, with the LSTM providing the best trade-off between prediction accuracy and computational cost. The boundedness of the LSTM mismatch estimate and input-to-state stability of the compensated observer error were also established under the stated assumptions.

The proposed controller was evaluated in a nonlinear mismatch simulation and in a measurement-informed study based on acoustic-path models identified from Kundt's tube measurements. For traffic, airplane, knock, and white-noise disturbances, the Hybrid MPC–LSTM controller achieved higher attenuation than FxNLMS and MPC without mismatch preview in both studies, and it also outperformed pure LSTM control in the measurement-informed evaluation. Relative to MPC without mismatch preview, the average noise-reduction improvement was approximately 14.35 dB in nonlinear simulation and 8.81 dB in the measurement-informed evaluation. These results show that, in this setting, learning is more effective when used as a model-correction module inside MPC

rather than as a direct replacement for MPC.

Keywords: active noise control; model predictive control; unmodelled dynamics; deep learning; long short-term memory.

Abstract in lingua italiana

Il controllo attivo del rumore è efficace alle basse frequenze, per le quali i trattamenti passivi risultano spesso poco pratici. Le sue prestazioni possono tuttavia degradare in presenza di disturbi non stazionari, effetti acustici nonlineari e incertezza nella dinamica dell'impianto. Gli algoritmi adattativi, come FxNLMS, richiedono una fase di convergenza e possono essere sensibili alla scelta del passo di adattamento, all'incertezza sul cammino secondario e alle variazioni del disturbo. I metodi basati su modello dipendono invece da modelli acustici identificati, che possono variare nelle applicazioni reali. In particolare, il controllo predittivo basato su modello (MPC) richiede modelli accurati dei cammini primario e secondario per predire l'errore residuo futuro. Questa tesi affronta tali limitazioni sviluppando un framework Hybrid MPC–LSTM per compensare il mismatch di modello nonlineare e dipendente dalla storia del sistema.

Il metodo proposto mantiene un controllore MPC lineare in spazio di stato e utilizza una rete causale long short-term memory (LSTM) per predire la sequenza del mismatch di modello aggregato sull'orizzonte dell'MPC. L'intera sequenza predetta viene utilizzata per correggere il modello di predizione esteso dell'MPC, mentre il primo elemento viene impiegato per compensare l'osservatore di stato. Durante il funzionamento in anello chiuso, il predittore LSTM viene utilizzato con parametri fissi addestrati offline. Sei architetture sequenziali ricorrenti, convoluzionali e basate sull'attenzione sono state confrontate mediante un protocollo comune; la rete LSTM ha fornito il miglior compromesso tra accuratezza di predizione e costo computazionale. La limitatezza della stima LSTM del mismatch e la stabilità input-to-state dell'errore dell'osservatore compensato sono state inoltre dimostrate sotto le ipotesi dichiarate.

Il controllore proposto è stato valutato mediante una simulazione con mismatch nonlineare e una valutazione basata su dati di misura, utilizzando modelli acustici identificati da un tubo di Kundt. Per disturbi di traffico, aeroplano, knock e rumore bianco, il controllore Hybrid MPC–LSTM ha ottenuto una maggiore attenuazione rispetto a FxNLMS e all'MPC senza mismatch preview in entrambi gli studi; nella valutazione basata su dati di misura ha inoltre superato il controllore pure LSTM. Rispetto all'MPC senza mismatch preview, il miglioramento medio della riduzione del rumore è stato pari a circa 14.35 dB

nella simulazione nonlineare e a 8.81 dB nella valutazione basata su dati di misura. Questi risultati mostrano che, nel contesto considerato, l'apprendimento è più efficace quando viene impiegato per compensare il mismatch di modello all'interno dell'MPC anziché per sostituire il controllore predittivo.

Parole chiave: controllo attivo del rumore; controllo predittivo basato su modello; dinamiche non modellate; deep learning; long short-term memory.

Contents

Abstract **i**

Abstract in lingua italiana **iii**

Contents **v**

Introduction **1**

1 Review of the state of the art **5**
1.1 Classical Adaptive ANC Algorithms 5
1.2 Model-Based and Predictive ANC 6
1.3 Nonlinear and Robust ANC 7
1.4 Deep Learning in ANC 7
1.5 Causal Sequence Prediction Models 8
1.6 Identified Research Gap 9

2 Methodology **11**
2.1 Nominal System Modeling 11
2.1.1 Joint State-Space Representation 11
2.1.2 Prediction Model 12
2.1.3 Cost Function and Analytical Solution 14
2.2 Delayed System Formulation 15
2.3 Nonlinear Model Mismatch Representation 19
2.3.1 Prediction Model in the Presence of Mismatch 21
2.4 State Estimation Using a Steady-State Kalman Observer 22
2.5 Deep-Learning-Based Mismatch Estimation 23
2.5.1 Training Data Construction and Mismatch Proxy Reconstruction . 24
2.5.2 Benchmarking Protocol for Causal Sequence Model Selection 26
2.6 Closed-Loop Use of the Frozen Mismatch Predictor 32

3 Numerical Simulation **35**
3.1 Simulation Setup and Parameters . . . 35
3.2 Benchmarking of Mismatch-Preview Estimators . . . 36
3.2.1 Benchmark Dataset and Excitation Design . . . 37
3.2.2 Hyperparameter Search Space . . . 37
3.2.3 Benchmark Results and Model Selection . . . 39
3.3 Noise Reduction Performance . . . 40
3.3.1 Time-Domain Residual Response . . . 41
3.3.2 Time–Frequency Residual Distribution . . . 42
3.3.3 Time-Varying Attenuation . . . 43
3.3.4 Power-Spectral-Density Analysis . . . 45
3.3.5 Quantitative Band-Limited Comparison . . . 46
3.3.6 Closed-Loop Mismatch-Prediction Tracking . . . 47

4 Implementation and Measurement-Informed Validation **49**
4.1 Measurement Setup . . . 49
4.2 Identification of the Acoustic Paths . . . 50
4.3 Validation of the Disturbance-Entry Direction Using Measured FIR Data . 54
4.4 Validation Results . . . 57
4.4.1 Pure LSTM Controller Baseline . . . 57
4.5 Comparative Noise-Reduction Performance . . . 61
4.5.1 Time-Domain Residual Response . . . 63
4.5.2 Time–Frequency Residual Distribution . . . 64
4.5.3 Time-Varying Attenuation . . . 66
4.5.4 Power-Spectral-Density Analysis . . . 67
4.5.5 Quantitative Band-Limited Comparison . . . 69

5 Conclusions and future developments **71**
5.1 Conclusions . . . 71
5.2 Future Developments . . . 72

Bibliography **75**

A Supplementary Derivation of the Delayed Prediction Model **81**
A.1 Delayed Path Realizations . . . 81
A.2 Lifted Delayed Prediction Matrices . . . 81

A.3 Relative-Delay Case and Causality 83

B Input-to-State Stability of the Compensated Observer **85**

B.1 Purpose of the Stability Analysis 85

B.2 Boundedness of the LSTM-Based Mismatch Estimate 86

B.3 Input-to-State Stability of the Compensated Observer 88

List of Figures **93**

List of Tables **95**

List of Symbols **97**

List of Abbreviations **103**

Introduction

Noise reduction is an important problem in transportation systems, industrial environments, ducts, cabins, and enclosed acoustic spaces. Passive noise-control treatments, such as absorptive materials and structural insulation, are effective at medium and high frequencies, but they often become bulky, costly, or inefficient at low frequencies [11, 31]. Active noise control (ANC) addresses this limitation by generating a secondary sound field that destructively interferes with the unwanted acoustic disturbance [14]. In a feedforward ANC system, the primary path describes the propagation of the unwanted disturbance to the error microphone, while the secondary path describes the propagation of the control loudspeaker signal to the same error microphone [10, 21].

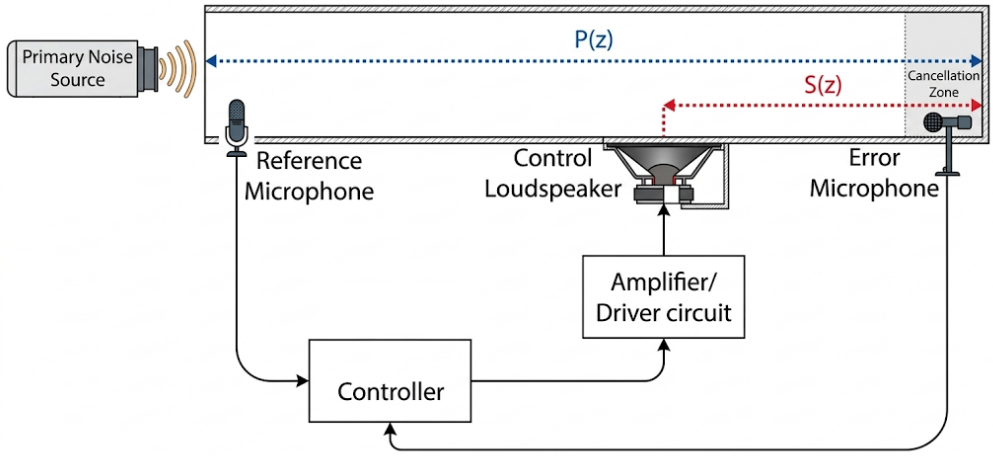


Figure 1: Feedforward active noise control configuration. The primary path $P(z)$ describes disturbance propagation to the error microphone, whereas the secondary path $S(z)$ describes propagation from the control loudspeaker to the error microphone.

Feedforward ANC systems are commonly implemented using adaptive filtered-reference algorithms, especially FxLMS and FxNLMS, because they are simple, computationally efficient, and able to adapt to changes in the acoustic environment [21, 27]. These algorithms update the control filter using the measured residual error and an estimate of the secondary acoustic path. However, their performance depends strongly on the step

size, the accuracy of the secondary-path model, and the stationarity of the disturbance. Under non-stationary or nonlinear conditions, and in the presence of actuator saturation or modelling uncertainty, convergence can become slow and the residual noise may remain significant.

Model predictive control (MPC) provides an alternative framework for ANC because it predicts the future residual error over a finite horizon and computes the secondary control signal through an optimization problem. In general, this structure can incorporate prediction horizons, acoustic delays, control weighting, and actuator constraints directly in the optimization. The controller implemented in this thesis instead uses the analytical solution of an unconstrained quadratic MPC problem and saturates only the first control input before it is applied. Delayed MPC formulations are especially relevant when the available acoustic delay can be exploited to predict the disturbance before it reaches the error microphone [24]. The main limitation is that MPC relies on the accuracy of the prediction model. If the real acoustic plant differs from the nominal model, the predicted residual error becomes inaccurate and the control action may no longer provide the expected attenuation.

Model mismatch is difficult to avoid in practical ANC systems. It may be caused by nonlinear loudspeaker behavior, actuator saturation, time-varying acoustic paths, unmodelled dynamics, sensor limitations, observer error, or changes in the disturbance characteristics. These effects are often nonlinear and history-dependent, so they cannot be represented accurately by a fixed linear model. Nonlinear and time-varying effects remain a major challenge in ANC, and purely linear control models may degrade under such conditions [28].

The central question of this thesis is how to preserve the predictive and receding-horizon structure of MPC when the nominal acoustic model is incomplete. Since future measurements are not available during real-time control, the mismatch-preview information must be estimated causally from past measured and known signals. To address this challenge, a causal sequence model is trained to infer the future mismatch behavior and the predicted mismatch sequence is then supplied to the MPC prediction model. This allows the controller to account for both the nominal acoustic response and the expected mismatch effect over the prediction horizon.

Deep learning (DL) has recently been investigated in ANC as a way to learn nonlinear relationships and temporal dependencies from data [38]. This is relevant to the present work because the mismatch between the real acoustic plant and the nominal prediction model can be nonlinear, time-varying, and dependent on the recent closed-loop history.

However, the aim of this thesis is not to replace MPC with a neural controller. Instead, DL is used as an auxiliary prediction module that provides MPC with a preview of the future mismatch. A pure DL-based controller is also evaluated later as a comparison baseline, but the proposed control strategy remains a hybrid MPC-DL framework.

The objective of this thesis is to develop and evaluate this hybrid framework for ANC under nonlinear model mismatch. The work includes causal feature construction, offline data generation using controlled nonlinear mismatch and evaluation using measured FIR/state-space acoustic-path models, benchmarking of candidate sequence models, integration of the selected fixed predictor into MPC, and numerical comparison with ANC off, FxNLMS, a pure DL controller, and MPC without mismatch preview.

The main contributions of this thesis are as follows:

- a hybrid MPC-LSTM framework is developed for ANC under nonlinear model mismatch;
- a causal feature-construction procedure is proposed using measured signals, observer residual information, and delayed mismatch-proxy information;
- the predicted mismatch sequence is integrated into both the observer correction and the lifted MPC prediction model;
- the stability of the mismatch-compensated observer is analyzed using an input-to-state stability argument, demonstrating that bounded mismatch-preview errors lead to bounded observer-estimation errors;
- a controlled and measured-data-based evaluation framework is developed, using nonlinear NARX-type mismatch for repeatable simulations and identified FIR/state-space acoustic models for realistic plant-model mismatch assessment;
- RNN, LSTM, GRU, 1D-CNN, TCN, and Transformer models are benchmarked for causal mismatch-preview prediction;
- the proposed controller is evaluated using time-domain responses, STFT spectra, time-varying attenuation, PSD analysis, and band-limited noise-reduction metrics.

The remainder of this thesis is organized as follows. Chapter 1 reviews active noise control, classical adaptive algorithms, MPC-based ANC, nonlinear ANC, and deep-learning-based approaches. Chapter 2 presents the nominal and delayed system formulations, MPC controller, nonlinear mismatch representation, causal feature construction, mismatch-preview predictor, and observer compensation strategy. Chapter 3 describes the numerical simulation setup, benchmark protocol, evaluation metrics, and closed-loop results. Chapter 4

presents the measurement setup, acoustic-path identification, implementation details, and measurement-informed validation. Finally, Chapter 5 summarizes the main findings and discusses future work.

1 | Review of the state of the art

This chapter reviews previous research related to ANC, classical adaptive algorithms, model-based and predictive control, nonlinear ANC, deep-learning-based ANC, and causal sequence prediction.

1.1. Classical Adaptive ANC Algorithms

Classical ANC research has been strongly shaped by adaptive filtering methods, particularly the least-mean-square (LMS) family and its filtered-reference variants [21, 22]. In feedforward ANC, the reference signal is processed to generate the secondary control input, while the residual error measured at the error microphone is used to update the controller. Early simulation studies demonstrated LMS-based adaptive sound cancellation in a duct under acoustic propagation delay [5]. This structure has made adaptive filtering the dominant practical approach for many linear ANC applications.

The filtered-x LMS (FxLMS) algorithm explicitly accounts for the secondary acoustic path between the control loudspeaker and the error microphone. The normalized version, FxNLMS, further adjusts the adaptation step using the filtered-reference signal energy, improving robustness when the reference amplitude varies. Recent surveys confirm that FxLMS-type algorithms remain central in linear ANC because of their simplicity, low computational cost, and ease of implementation [27, 33].

The same literature also identifies important limitations. The convergence speed and steady-state error of FxLMS-type algorithms depend strongly on the step-size parameter. Their performance also depends on the secondary-path estimate and on the stationarity of the disturbance. In the presence of non-stationary noise, actuator saturation, nonlinear loudspeaker behavior, time-varying acoustic paths, and plant uncertainty, the residual noise may remain significant [21, 27]. The convergence and robustness properties of FxLMS/FxNLMS are closely related to classical adaptive-filter theory [15, 35], while frequency-domain variants further highlight the importance of delay, secondary-path modelling, and stability in practical ANC [37]. These limitations motivate comparison

against FxNLMS, but also indicate why prediction, systematic control-effort weighting, and explicit treatment of model mismatch are needed.

1.2. Model-Based and Predictive ANC

Model-based ANC methods use identified acoustic-path models to describe how the primary disturbance and secondary control signal affect the residual error. In state-space form, the primary and secondary paths can be combined into an augmented dynamic model, making it possible to predict future residual errors from the current state, the primary input, and a candidate secondary control sequence.

Previous work has shown that MPC is useful for active noise and vibration control when future prediction and control constraints need to be considered [36]. More generally, robust MPC provides a formal control framework for constrained linear systems subject to bounded disturbances and modelling errors [29]. Predictive control has also been used in uncertain robotic interaction problems, such as optimal predictive impedance control for a lower-limb rehabilitation robot [18]. In ANC, delayed MPC formulations are especially relevant because acoustic propagation delays can provide preview of the disturbance before it reaches the error microphone. Recent MPC-based ANC studies have demonstrated that this delay can be exploited to improve attenuation and to compare favorably with adaptive filtering baselines [24].

However, the MPC literature also makes clear that prediction quality depends on model accuracy. In ANC, the nominal model is usually obtained from identified primary and secondary acoustic paths. If the real acoustic system differs from this nominal model, the predicted residual error becomes inaccurate. This mismatch may arise from identification errors, unmodelled dynamics, nonlinear loudspeaker behavior, actuator saturation, or changes in the acoustic environment. Thus, MPC provides a strong framework for prediction and control-effort optimization, but it remains vulnerable when the nominal linear model does not represent the real closed-loop behavior accurately.

This observation is important for the present work because it separates two issues: the control optimization can remain linear and computationally structured, while the missing model component may still be nonlinear and history-dependent. The open question is how to supply this missing information to the MPC prediction model without replacing the controller by a fully nonlinear or purely data-driven controller.

1.3. Nonlinear and Robust ANC

Nonlinear effects are a major challenge in practical ANC systems. They can be caused by loudspeaker distortion, actuator saturation, nonlinear electronics, nonlinear acoustic propagation, or nonlinear coupling between the control signal and the measured residual error. Linear adaptive filters and nominal linear state-space models may not represent these effects accurately. Surveys of nonlinear ANC have reviewed several approaches for dealing with such effects, including Volterra filters, functional-link neural networks, spline filters, kernel adaptive filters, and nonlinear distributed ANC algorithms [28].

NARX-type models are particularly relevant when the system behavior depends on past inputs and outputs. They can represent nonlinear and history-dependent dynamics using delayed signal histories, making them suitable for describing nonlinear mismatch effects in a controlled and repeatable way. Previous work has used NARX models for nonlinear ANC, especially when nonlinear dynamics cannot be captured by a fixed linear model [30]. Other nonlinear adaptive methods, such as filtered-s LMS and functional-link neural-network-based algorithms, have also been proposed to improve ANC performance under nonlinear noise or nonlinear plant conditions [8].

Although nonlinear ANC methods can improve performance in some cases, they also have limitations. Volterra and kernel methods may become computationally expensive as memory length or nonlinear order increases. Neural nonlinear controllers may require substantial training data and may not naturally include actuator limitations or prediction horizons. Robust control methods can account for uncertainty, but they may require conservative uncertainty bounds that are difficult to obtain experimentally. Moreover, many nonlinear ANC methods are designed to directly update the control filter or approximate the nonlinear control law. They do not usually provide a causal future mismatch sequence that can be inserted into a linear MPC prediction model.

1.4. Deep Learning in ANC

Deep learning has recently been investigated as a way to learn nonlinear mappings and temporal dependencies in ANC. Deep ANC methods have shown that neural networks can be trained to generate control signals from data, especially when classical linear adaptive filters are insufficient for nonlinear systems [38]. Other studies have explored recurrent, convolutional, and hybrid neural structures for ANC control or enhancement tasks [23, 32, 41]. Low-latency neural ANC has also been studied using recurrent and attentive architectures, where causality and delay constraints are central design issues [39,

40]. More recent causal neural ANC approaches further emphasize that learned ANC systems must satisfy real-time causality requirements and should be evaluated against strong baselines [1]. In broader nonlinear control, LSTM-based uncertainty estimation has also been combined with reinforcement-learning and optimal-control structures for systems with uncertain dynamics [20].

These studies show that deep learning can be useful for modelling nonlinear and time-dependent behavior in ANC. However, many DL-based ANC methods use the neural network as the main controller, directly mapping available signals to the secondary control signal. Although this can be effective, it may reduce interpretability and does not preserve the optimization structure of MPC. A pure neural controller also does not naturally minimize a finite-horizon quadratic cost within an explicit predictive optimization framework.

The limitation for the present problem is therefore not the absence of neural networks in ANC, but the role assigned to them. Existing DL-based ANC methods commonly use the neural network to generate or estimate the cancelling control signal, whereas MPC-based ANC methods typically retain a nominal model-based prediction structure. This leaves room for a hybrid formulation in which the learned model is used only to provide a causal future mismatch preview, while the linear MPC controller remains responsible for the finite-horizon control optimization.

1.5. Causal Sequence Prediction Models

Causal mismatch-preview estimation can be viewed as a sequence-prediction problem. At each sampling instant, a model receives a window of past measured and known signals and estimates a future sequence over the MPC horizon. Since future measurements are not available during real-time operation, architectures used for this task must respect causality.

Several neural architectures are suitable for temporal prediction. Recurrent neural networks process sequences through hidden states, but standard recurrent networks can suffer from vanishing gradients when long temporal dependencies are present [25]. Long short-term memory (LSTM) networks were introduced to address this limitation and are widely used for sequence modelling with longer memory [13, 16]. Gated recurrent units (GRUs) provide a related gated structure with fewer parameters and can be considered a lighter alternative to LSTMs [6, 7]. Temporal convolutional networks (TCNs) use causal and dilated convolutions to model temporal dependencies efficiently over long windows [2]. Transformer models use attention mechanisms to represent dependencies across a sequence

and have become important in many sequence-modelling applications [34].

For ANC and real-time control, architecture selection must consider not only prediction accuracy but also causality, computational cost, and latency. Previous ANC studies often select one neural architecture and tune it for a specific task. A systematic benchmark across recurrent, convolutional, and attention-based models under the same mismatch-preview task is therefore useful for separating model-family effects from differences in data preparation or training protocol.

1.6. Identified Research Gap

The reviewed literature shows that classical FxLMS and FxNLMS algorithms are simple and adaptive, but their performance can degrade under nonlinear, non-stationary, and uncertain acoustic conditions. MPC-based ANC provides prediction and structured control optimization, but it depends strongly on the accuracy of the nominal acoustic model. Nonlinear and deep-learning-based ANC methods can model nonlinear behavior, but many existing approaches either replace the controller, increase computational complexity, or do not provide a causal future mismatch sequence that can be incorporated into an MPC prediction horizon.

Therefore, the remaining gap is an ANC framework that combines causal mismatch-preview learning with a linear MPC controller. Such a framework should preserve MPC as the main finite-horizon optimizer, estimate nonlinear and history-dependent mismatch from online-available signals, provide a future mismatch sequence over the prediction horizon, and be evaluated under controlled nonlinear mismatch and measured-data-based plant-model mismatch.

2 | Methodology

2.1. Nominal System Modeling

2.1.1. Joint State-Space Representation

Consider the active noise control (ANC) configuration consisting of a primary acoustic path, denoted by the superscript p, and a secondary acoustic path, denoted by the superscript s. In discrete time, the two paths are modeled as

$$\begin{cases} x_{k+1}^p = A^p x_k^p + B^p u_k^p, \\ y_k^p = C^p x_k^p + D^p u_k^p, \end{cases} \tag{2.1}$$

and

$$\begin{cases} x_{k+1}^s = A^s x_k^s + B^s u_k^s, \\ y_k^s = C^s x_k^s + D^s u_k^s, \end{cases} \tag{2.2}$$

where $k \in \mathbb{Z}_{\geq 0}$ denotes the sampling instant, $u_k^p \in \mathbb{R}$ is the input associated with the primary path, $u_k^s \in \mathbb{R}$ is the control input applied to the secondary source, and $x_k^p \in \mathbb{R}^{n_p}$ and $x_k^s \in \mathbb{R}^{n_s}$ are the corresponding state vectors. Accordingly, the primary-path matrices satisfy

$$A^p \in \mathbb{R}^{n_p \times n_p}, \qquad B^p \in \mathbb{R}^{n_p \times 1}, \qquad C^p \in \mathbb{R}^{1 \times n_p}, \qquad D^p \in \mathbb{R},$$

while the secondary-path matrices satisfy

$$A^s \in \mathbb{R}^{n_s \times n_s}, \qquad B^s \in \mathbb{R}^{n_s \times 1}, \qquad C^s \in \mathbb{R}^{1 \times n_s}, \qquad D^s \in \mathbb{R}.$$

The outputs $y_k^p \in \mathbb{R}$ and $y_k^s \in \mathbb{R}$ represent, respectively, the primary disturbance contribution and the anti-noise contribution observed at the error sensor. For controller design, it is convenient to combine the two subsystems into an augmented state-space model.

Define the augmented state vector

$$x_k \triangleq \begin{bmatrix} x_k^p \\ x_k^s \end{bmatrix}, \tag{2.3}$$

together with the block matrices

$$A \triangleq \begin{bmatrix} A^p & 0 \\ 0 & A^s \end{bmatrix}, \qquad B_{up} \triangleq \begin{bmatrix} B^p \\ 0 \end{bmatrix}, \qquad B_{us} \triangleq \begin{bmatrix} 0 \\ B^s \end{bmatrix}, \qquad C \triangleq \begin{bmatrix} C^p & -C^s \end{bmatrix}. \tag{2.4}$$

The resulting joint model is then written as

$$\begin{cases} x_{k+1} = Ax_k + B_{up}u_k^p + B_{us}u_k^s, \\ y_k = y_k^p - y_k^s = Cx_k + D^p u_k^p - D^s u_k^s. \end{cases} \tag{2.5}$$

Equation (2.5) provides a compact description of the residual sound pressure at the error sensor. In particular, the output y_k represents the error signal to be minimized by the controller, and is obtained as the superposition of the primary disturbance and the secondary anti-noise field.

2.1.2. Prediction Model

Let N denote the prediction horizon. The future primary input sequence, secondary control input sequence, and predicted residual output sequence are defined as

$$U^p = \begin{bmatrix} u_k^p \\ u_{k+1}^p \\ \vdots \\ u_{k+N}^p \end{bmatrix}, \qquad U^s = \begin{bmatrix} u_k^s \\ u_{k+1}^s \\ \vdots \\ u_{k+N}^s \end{bmatrix}, \qquad Y = \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+N} \end{bmatrix}. \tag{2.6}$$

Thus, $U^p \in \mathbb{R}^{N+1}$, $U^s \in \mathbb{R}^{N+1}$, and $Y \in \mathbb{R}^N$. The additional terminal input sample is required to retain the direct feedthrough terms D^p and D^s in the lifted prediction matrices. For clarity, the derivation is presented for the SISO case, with one primary reference input, one secondary actuator, and one error sensor.

Using the augmented model in Equation (2.5), the future residual output sequence can be expressed as

$$Y = Ox_k + M^p U^p - M^s U^s, \tag{2.7}$$

where $O \in \mathbb{R}^{N\times(n_p+n_s)}$ is the observability prediction matrix, while M^p and $M^s \in \mathbb{R}^{N\times(N+1)}$ are the prediction matrices associated with the primary and secondary paths, respectively.

The matrix O is defined as

$$O = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix}. \tag{2.8}$$

The primary-path prediction matrix is given by

$$M^p = \begin{bmatrix} C^pB^p & D^p & 0 & \cdots & 0 & 0 \\ C^pA^pB^p & C^pB^p & D^p & \cdots & 0 & 0 \\ C^p(A^p)^2B^p & C^pA^pB^p & C^pB^p & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ C^p(A^p)^{N-1}B^p & C^p(A^p)^{N-2}B^p & C^p(A^p)^{N-3}B^p & \cdots & C^pB^p & D^p \end{bmatrix}. \tag{2.9}$$

Similarly, the secondary-path prediction matrix is defined as

$$M^s = \begin{bmatrix} C^sB^s & D^s & 0 & \cdots & 0 & 0 \\ C^sA^sB^s & C^sB^s & D^s & \cdots & 0 & 0 \\ C^s(A^s)^2B^s & C^sA^sB^s & C^sB^s & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ C^s(A^s)^{N-1}B^s & C^s(A^s)^{N-2}B^s & C^s(A^s)^{N-3}B^s & \cdots & C^sB^s & D^s \end{bmatrix}. \tag{2.10}$$

Since the current state x_k and the future primary disturbance sequence U^p are assumed to be known or estimated at the current sampling instant, their contribution can be grouped into the constant term

$$c = Ox_k + M^pU^p. \tag{2.11}$$

Therefore, the predicted residual error sequence can be written as an explicit function of the future secondary control sequence:

$$Y = c - M^sU^s. \tag{2.12}$$

Equation (2.12) is the prediction model used inside the MPC optimization problem. It shows that the controller acts through the secondary path prediction matrix M^s in order

to reduce the predicted residual error over the prediction horizon.

2.1.3. Cost Function and Analytical Solution

The MPC objective is to minimize the predicted residual error while penalizing excessive control effort. This trade-off is expressed through the quadratic cost function

$$J(U^s) = Y^\top Q Y + (U^s)^\top R U^s, \tag{2.13}$$

where $Q \in \mathbb{R}^{N \times N}$ is a positive semi-definite weighting matrix that penalizes the predicted residual error, and $R \in \mathbb{R}^{(N+1)\times(N+1)}$ is a positive definite weighting matrix that penalizes the control effort.

Substituting Equation (2.12) into Equation (2.13) gives

$$J(U^s) = (c - M^s U^s)^\top Q (c - M^s U^s) + (U^s)^\top R U^s. \tag{2.14}$$

The optimal unconstrained control sequence is obtained by differentiating the cost function with respect to U^s and setting the derivative equal to zero:

$$\frac{\partial J(U^s)}{\partial U^s} = 0. \tag{2.15}$$

This gives the analytical unconstrained solution

$$U^{s*}_{\text{unc}} = H^{-1} g, \tag{2.16}$$

where

$$H = (M^s)^\top Q M^s + R, \qquad g = (M^s)^\top Q c. \tag{2.17}$$

The vector U^{s*}_{unc} contains the unconstrained optimal sequence of future secondary control inputs over the prediction horizon. In accordance with the receding-horizon implementation of MPC, only the first element of this sequence is selected:

$$u^s_{k,\text{unc}} = \left[U^{s*}_{\text{unc}}\right]_1 . \tag{2.18}$$

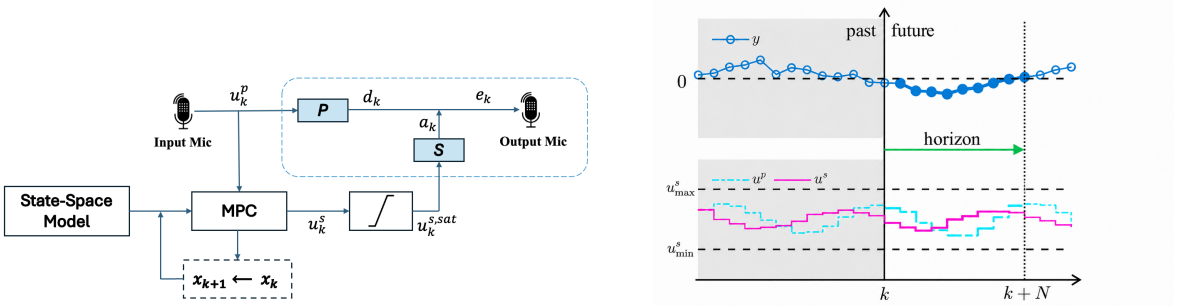
The analytical solution in Equation (2.16) corresponds to an unconstrained MPC problem. In this work, actuator limits are not included directly inside the optimization problem. Instead, the first element of the unconstrained MPC solution is saturated before being

applied to the secondary source. The applied control input is therefore defined as

$$u_k^{s,\text{sat}} = \begin{cases} u_{\max}^s, & \text{if } u_{k,\text{unc}}^s > u_{\max}^s, \\ u_{\min}^s, & \text{if } u_{k,\text{unc}}^s < u_{\min}^s, \\ u_{k,\text{unc}}^s, & \text{otherwise}, \end{cases} \tag{2.19}$$

where $u_{\min}^s$ and $u_{\max}^s$ denote the minimum and maximum admissible values of the secondary control input, respectively. At the next sampling instant, the state information and the input predictions are updated, the optimization problem is solved again, and a new control action is computed.

The nominal MPC-based ANC configuration is shown in Figure 2.1. At the current time instant k, the controller predicts the residual error signal over the horizon N and computes the future secondary control sequence. However, this nominal formulation requires the future primary input sequence U^p, which is not directly available in real-time implementation.



(a) System diagram. (b) Sampling instants.

Figure 2.1: Nominal MPC-based ANC.

2.2. Delayed System Formulation

The MPC formulation presented in the previous section requires the future primary input sequence U^p in order to compute the term

$$c = Ox_k + M^p U^p. \tag{2.20}$$

However, in a real-time implementation, future values of the primary input are not directly available at the current sampling instant. This represents an important causality issue, since the control input must be computed using only information that is available at the

present time.

In active noise control systems, this limitation can be addressed by exploiting the acoustic propagation delay of the primary and secondary paths. If the primary input is measured before it reaches the error sensor, the primary path delay provides preview information about the u^p that will affect the error sensor in the future. Therefore, the delayed dynamics of the primary and secondary acoustic paths are explicitly included in the prediction model.

Let N_d^p and N_d^s denote the number of discrete-time sampling delays in the primary and secondary paths, respectively. The measured transfer functions can be written as the product of their delay-free counterparts and pure delay terms:

$$H^p_{\text{meas}}(z) = H^p_{\text{delayless}}(z)z^{-N_d^p}, \qquad H^s_{\text{meas}}(z) = H^s_{\text{delayless}}(z)z^{-N_d^s}. \tag{2.21}$$

The corresponding delayed augmented state vector is defined as:

$$x'_k = \begin{bmatrix} x^p_{k-N_d^p} \\ x^s_{k-N_d^s} \end{bmatrix}. \tag{2.22}$$

Using this definition, the joint delayed state-space model becomes

$$\begin{cases} x'_{k+1} = \begin{bmatrix} A^p_{dl} & 0 \\ 0 & A^s_{dl} \end{bmatrix} x'_k + \begin{bmatrix} B^p_{dl} \\ 0 \end{bmatrix} u^p_{k-N_d^p} + \begin{bmatrix} 0 \\ B^s_{dl} \end{bmatrix} u^s_{k-N_d^s}, \\ y_k = y^p_k - y^s_k = \begin{bmatrix} C^p_{dl} & -C^s_{dl} \end{bmatrix} x'_k + D^p_{dl} u^p_{k-N_d^p} - D^s_{dl} u^s_{k-N_d^s}. \end{cases} \tag{2.23}$$

For compactness, the delayed augmented matrices are defined as

$$A_{dl} = \begin{bmatrix} A^p_{dl} & 0 \\ 0 & A^s_{dl} \end{bmatrix}, \qquad B^{p,a}_{dl} = \begin{bmatrix} B^p_{dl} \\ 0 \end{bmatrix}, \qquad B^{s,a}_{dl} = \begin{bmatrix} 0 \\ B^s_{dl} \end{bmatrix}, \qquad C_{dl} = \begin{bmatrix} C^p_{dl} & -C^s_{dl} \end{bmatrix}. \tag{2.24}$$

Thus, Equation (2.23) can be written as

$$\begin{cases} x'_{k+1} = A_{dl} x'_k + B^{p,a}_{dl} u^p_{k-N_d^p} + B^{s,a}_{dl} u^s_{k-N_d^s}, \\ y_k = C_{dl} x'_k + D^p_{dl} u^p_{k-N_d^p} - D^s_{dl} u^s_{k-N_d^s}. \end{cases} \tag{2.25}$$

The delayed primary and secondary input sequences over the prediction horizon N are defined like in Equation (2.6), but with the appropriate delay shifts. The available preview length is determined by the delay difference

$$N_d = N_d^p - N_d^s. \tag{2.26}$$

To ensure causality, the prediction horizon is selected to satisfy

$$N + 1 \leq N_d. \tag{2.27}$$

This condition guarantees that all elements of the delayed primary input sequence required in the prediction model are available when the control sequence is computed. In particular, because the direct feedthrough term requires the terminal delayed primary sample $u^p_{k+N-N_d}$, the condition $N + 1 \leq N_d$ ensures that the latest required primary sample occurs no later than the past relative to the current computation instant.

In practice, a large primary path delay and a small secondary path delay are desirable. A larger primary delay provides more preview information about the disturbance, while a smaller secondary delay allows the control signal to affect the error sensor more rapidly. Therefore, in this work, the secondary delay is taken as zero and only the relative delay in the primary path is considered:

$$N_d^s = 0, \qquad N_d = N_d^p. \tag{2.28}$$

Under this assumption, the delayed joint model becomes

$$\begin{cases} x'_{k+1} = A_{dl}x'_k + B^{p,a}_{dl}u^p_{k-N_d} + B^{s,a}_{dl}u^s_k, \\ y_k = C_{dl}x'_k + D^p_{dl}u^p_{k-N_d} - D^s_{dl}u^s_k. \end{cases} \tag{2.29}$$

Based on the delayed joint model, the predicted residual error sequence can be expressed as

$$Y = O_{dl}x'_k + M^p_{dl}U'^p - M^s_{dl}U'^s, \tag{2.30}$$

where O_{dl}, M^p_{dl}, and M^s_{dl} are the delayed counterparts of the prediction matrices defined in Equations (2.8), (2.9), and (2.10), respectively. The detailed construction of the delayed lifted matrices and the causality index argument are provided in Appendix A. The delayed

constant term independent of U'^s is defined as

$$c' = O_{dl}x'_k + M^p_{dl}U'^p. \tag{2.31}$$

Therefore, Equation (2.30) can also be written as

$$Y = c' - M^s_{dl}U'^s. \tag{2.32}$$

As in the nominal MPC formulation, the same quadratic cost function is used for the delayed MPC formulation:

$$J(U'^s) = Y^\top QY + (U'^s)^\top RU'^s. \tag{2.33}$$

Substituting Equation (2.30) into the cost function and setting the derivative with respect to U'^s equal to zero gives the unconstrained optimal control sequence

$$U'^{s*}_{\text{unc}} = H'^{-1}g', \tag{2.34}$$

where

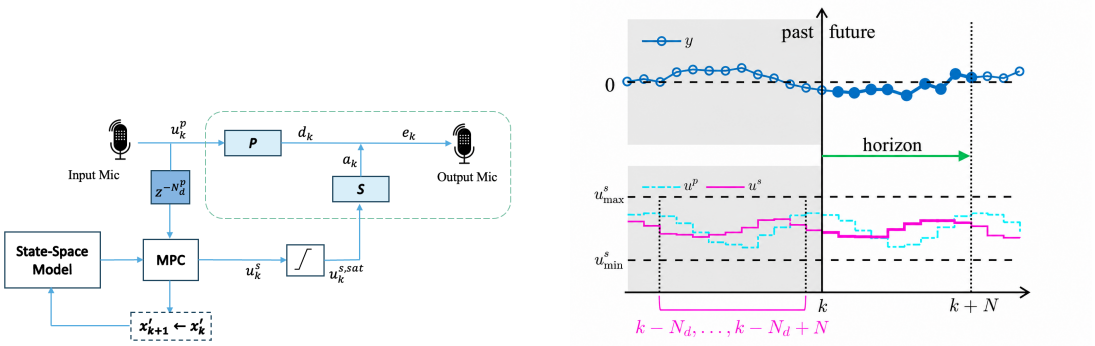
$$H' = (M^s_{dl})^\top QM^s_{dl} + R, \qquad g' = (M^s_{dl})^\top Qc'. \tag{2.35}$$

At each sampling instant, only the first element of the unconstrained optimal delayed control sequence is selected:

$$u^s_{k,\text{unc}} = \left[U'^{s*}_{\text{unc}}\right]_1. \tag{2.36}$$

If actuator limits are considered, this value is saturated before being applied. Therefore, as in the nominal formulation, the implemented control law is an unconstrained MPC law followed by input saturation, not a fully constrained MPC law.

The delayed MPC-based ANC configuration is shown in Figure 2.2. By introducing the primary path delay, the primary input sequence required for prediction is shifted into the past. Therefore, the disturbance samples required by the MPC prediction model can be available at time k, provided that the available preview is sufficiently long.



(a) System diagram. (b) Sampling instants.

Figure 2.2: MPC-based ANC with primary path delay.

The main advantage of the delayed formulation is that it makes the MPC law causal. The required primary input sequence U'^{p} is shifted into the past by the primary path delay. Therefore, provided that the available acoustic preview is sufficiently large, the disturbance samples required for prediction are already available when the controller computes the secondary control sequence.

For simplicity of notation, the subscript dl and the prime notation are omitted in the remainder of this thesis. Therefore, unless otherwise stated, the matrices A, B_{up}, B_{us}, and C refer to the delay-free matrices of the delayed system formulation, and the state x_k denotes the delayed augmented state. Similarly, the input sequences U^p and U^s denote the corresponding shifted primary and secondary input sequences used in the delayed MPC prediction model.

2.3. Nonlinear Model Mismatch Representation

The MPC controller developed in the previous sections relies on the identified nominal state-space model of the ANC system. Although this model is used for prediction and control design, it cannot fully capture the true dynamics of a practical system. In real ANC implementations, model mismatch may arise from acoustic coupling effects, parametric uncertainties, identification errors, unmodeled dynamics, nonlinear behavior, and environment-dependent variations [26, 29].

To represent the discrepancy between the nominal model and the true system behavior, an additive state mismatch term is introduced into the state equation. The mismatch contribution is defined as

$$\tilde{x}_k = B^d d_k, \tag{2.37}$$

where d_k denotes the lumped model mismatch and B^d defines the direction through which the mismatch affects the augmented state vector. The term lumped indicates that different sources of uncertainty and unmodeled effects are collected into a single scalar signal d_k.

In this work, it is assumed that the lumped model mismatch enters the state equation through the known direction B^d. The vector B^d is normalized as $\|B^d\|_2 = 1$, and the mismatch signal is assumed to be bounded:

$$|d_k| \leq \bar{d}, \tag{2.38}$$

where $\bar{d}$ is a positive constant representing an upper bound on the magnitude of the model mismatch.

Therefore, the true system used in the simulation is written as

$$x_{k+1} = Ax_k + B_{up}u_k^p + B_{us}u_k^s + \tilde{x}_k. \tag{2.39}$$

By substituting Equation (2.37) into Equation (2.39), the true system can equivalently be expressed as

$$x_{k+1} = Ax_k + B_{up}u_k^p + B_{us}u_k^s + B^d d_k. \tag{2.40}$$

The output equation is kept in the same form as the nominal model:

$$y_k = Cx_k + D^p u_k^p - D^s u_k^s. \tag{2.41}$$

Thus, the mismatch affects the residual error signal indirectly through its influence on the state evolution.

The main contribution of this work is to estimate the unknown lumped mismatch signal d_k and compensate for its effect inside the MPC prediction model. By including the estimated mismatch in the prediction, the controller is not based only on the identified nominal model, but also accounts for the discrepancy between the nominal and true system dynamics.

2.3.1. Prediction Model in the Presence of Mismatch

Over the prediction horizon N, the mismatch sequence is defined as

$$D_m = \begin{bmatrix} d_k \\ d_{k+1} \\ \vdots \\ d_{k+N-1} \end{bmatrix}. \tag{2.42}$$

Including the mismatch contribution in the state evolution, the predicted residual error sequence becomes

$$Y = Ox_k + M^p U^p - M^s U^s + M^d D_m, \tag{2.43}$$

where M^d is the mismatch prediction matrix. Since the mismatch enters the state equation through B^d, this matrix is defined as

$$M^d = \begin{bmatrix} CB^d & 0 & \cdots & 0 \\ CAB^d & CB^d & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B^d & CA^{N-2}B^d & \cdots & CB^d \end{bmatrix}. \tag{2.44}$$

The terms that are independent of the future secondary control sequence are grouped into the mismatch-affected constant term:

$$c_m = Ox_k + M^p U^p + M^d D_m. \tag{2.45}$$

Therefore, the predicted residual error sequence can be written compactly as

$$Y = c_m - M^s U^s. \tag{2.46}$$

Using the same quadratic MPC formulation derived in the nominal and delayed cases, but replacing the nominal constant term with the mismatch-affected term c_m, the unconstrained optimal secondary control sequence becomes

$$U^{s*} = H^{-1} g_m, \tag{2.47}$$

where

$$H = (M^s)^\top Q M^s + R, \qquad g_m = (M^s)^\top Q c_m. \tag{2.48}$$

Compared with the nominal MPC formulation, the matrix H remains unchanged. The effect of the mismatch appears only through the vector g_m, because the constant term c_m contains the additional contribution $M^d D_m$. Therefore, the mismatch changes the optimal secondary control sequence by modifying the predicted residual error over the horizon.

This formulation shows how the model mismatch affects the ideal MPC prediction. However, in practical implementation, the true mismatch sequence D_m is not directly available. Therefore, the following section introduces the estimation and compensation strategy used to approximate this contribution inside the observer and MPC formulation.

2.4. State Estimation Using a Steady-State Kalman Observer

The MPC prediction model requires the current state vector x_k. However, in practical ANC implementation, the internal states of the identified acoustic model are not directly measurable. The available measured signal is the residual error signal y_k, while the primary input u_k^p and the secondary control input u_k^s are known. Therefore, a state observer is required to estimate the augmented state used by the MPC controller.

Let $\hat{x}_{c,k}$ denote the compensated state estimate. Assuming that the pair (A, C) is detectable, the steady-state Kalman observer gain L is used to define the compensated observer as

$$\hat{x}_{c,k+1} = A\hat{x}_{c,k} + B_{up}u_k^p + B_{us}u_k^s + Le_{c,k} + B^d\hat{d}_k, \tag{2.49}$$

where $\hat{d}_k$ is the estimated lumped mismatch signal.

The observer correction error is defined as

$$e_{c,k} = y_k - C\hat{x}_{c,k} - D^p u_k^p + D^s u_k^s. \tag{2.50}$$

The quantity $e_{c,k}$ represents the output innovation after removing the known primary- and secondary-path direct-feedthrough contributions.

Over the prediction horizon, the estimated mismatch sequence is

$$\hat{D}_{m,k} = \begin{bmatrix} \hat{d}_k \\ \hat{d}_{k+1} \\ \vdots \\ \hat{d}_{k+N-1} \end{bmatrix}. \tag{2.51}$$

Using the estimated state and mismatch sequence, the constant term in the MPC prediction model becomes

$$c_{\text{comp}} = O\hat{x}_{c,k} + M^p U^p + M^d \hat{D}_{m,k}. \tag{2.52}$$

Thus, the same MPC solution derived previously is used with c_m replaced by c_{comp}:

$$U^{s*} = H^{-1} g_{\text{comp}}, \qquad g_{\text{comp}} = (M^s)^\top Q c_{\text{comp}}. \tag{2.53}$$

Only the first element of U^{s*} is applied to the secondary source. The stability of the compensated observer in the presence of the additive mismatch-estimation term $B^d \hat{d}_k$ is discussed in Appendix B.

2.5. Deep-Learning-Based Mismatch Estimation

The previous sections introduced the model mismatch as an unknown lumped signal that affects the state evolution of the ANC system through the known direction B^d. Although this representation provides a compact way to include model uncertainty in the observer and MPC formulation, estimating the mismatch contribution is difficult in practice.

The difficulty arises because model mismatch in ANC is not generally a simple static error. Instead, it may be caused by several coupled effects, such as non-minimum phase behavior, hidden or poorly observable states, actuator saturation, sensor saturation, modeling uncertainty, and nonlinear dynamics. Consequently, the mismatch may be time-varying, input-dependent, and dependent on the previous evolution of the system.

Classical mathematical estimation approaches remain important when the uncertainty has a known structure or can be described using fixed parametric or stochastic models. However, in this work, the model mismatch is treated as a complex lumped effect that may not admit a simple analytical description. Therefore, a deep-learning-based approach is adopted to estimate the mismatch sequence required by the compensated observer and MPC prediction model.

Specifically, the deep learning model is used to estimate the future mismatch sequence over the prediction horizon in Equation (2.51).

The first element of this sequence, $\hat{d}_k$, is injected into the compensated observer through the term $B^d \hat{d}_k$, while the complete estimated sequence $\hat{D}_{m,k}$ is included in the MPC prediction model through the term $M^d \hat{D}_{m,k}$. Therefore, the deep learning model is not used to replace the nominal state-space model, the Kalman observer, or the MPC controller.

Instead, it provides an estimated mismatch sequence that allows the observer and controller to compensate for the discrepancy between the nominal model and the true system behavior.

2.5.1. Training Data Construction and Mismatch Proxy Reconstruction

The deep-learning model is used to estimate the future mismatch sequence required by the compensated observer and MPC prediction model. At sampling instant k, the estimator can only use information available up to the previous sampling instant. Therefore, the DL input is constructed from past measured and known signals up to time $k-1$, and the network predicts the future mismatch sequence over the MPC prediction horizon.

The feature vector at time $k-1$ is defined as

$$\phi_{k-1} = \begin{bmatrix} u^p_{k-1} \\ u^s_{k-1} \\ y_{k-1} \\ \Delta y_{k-1} \\ d_{\text{proxy},k-2} \end{bmatrix}, \tag{2.54}$$

where u^p_{k-1} is the primary input, u^s_{k-1} is the secondary control input, and y_{k-1} is the measured residual output of the real system. The quantity Δy_{k-1} denotes the output mismatch between the real system and the nominal model, defined below in Equation (2.58), and the last component of the feature vector, $d_{\text{proxy},k-2}$, is the previously reconstructed mismatch proxy.

To capture the temporal behavior of the mismatch, a sequence of past feature vectors is used as the DL input. For a sequence length S_L, the input window is written as

$$\Phi_k = [\phi_{k-S_L},\, \phi_{k-S_L+1},\, \ldots,\, \phi_{k-1}]\,. \tag{2.55}$$

The DL output is the estimated future mismatch sequence over the MPC prediction horizon, denoted by $\hat{D}_{m,k}$ in Equation (2.51).

For supervised learning, the desired mismatch sequence is not directly measurable because the true system state is not available. Therefore, the training targets are constructed offline after collecting input-output data from the real system. A nominal mismatch-free model is simulated using the same logged primary and secondary inputs. The nominal

model is

$$x_{\text{nom},k+1} = Ax_{\text{nom},k} + B_{up}u_k^p + B_{us}u_k^s, \tag{2.56}$$

with the corresponding nominal output

$$y_{\text{nom},k} = Cx_{\text{nom},k} + D^p u_k^p - D^s u_k^s. \tag{2.57}$$

Let y_k denote the measured output of the real system. The output mismatch between the real system and the nominal model is then defined as

$$\Delta y_k = y_k - y_{\text{nom},k}. \tag{2.58}$$

To reconstruct a scalar mismatch proxy, consider the state error between the real system and the nominal model,

$$e_{m,k} = x_k - x_{\text{nom},k}. \tag{2.59}$$

Using the true and nominal dynamics, this error satisfies

$$e_{m,k+1} = Ae_{m,k} + B^d d_k \tag{2.60}$$

and the output mismatch is related to the state error by $\Delta y_k = Ce_{m,k}$. The scalar proxy $d_{\text{proxy},k}$ is then reconstructed offline as an approximation of the unknown mismatch d_k.

Since $e_{m,k}$ is not measurable, it is approximated along the known mismatch direction B^d. The corresponding output-space direction is defined as $g_d = CB^d$, with $g_d \neq 0$. Using the measured output mismatch at time $k-1$, a proxy of the state error is reconstructed as

$$e_{\text{proxy},k-1} = B^d \left(g_d^\top g_d\right)^{-1} g_d^\top \Delta y_{k-1}. \tag{2.61}$$

Then, from the one-step relation

$$\Delta y_k = CAe_{m,k-1} + CB^d d_{k-1}, \tag{2.62}$$

the scalar mismatch proxy is computed as

$$d_{\text{proxy},k-1} = \left(g_d^\top g_d\right)^{-1} g_d^\top \left[\Delta y_k - CAe_{\text{proxy},k-1}\right]. \tag{2.63}$$

This reconstruction uses only the logged output mismatch and the known nominal model matrices, and therefore does not require access to the true system state.

After the complete dataset has been collected, the proxy sequence is computed offline. The supervised learning target corresponding to the input window Φ_k is then defined as

$$D_{\text{proxy},k} = \begin{bmatrix} d_{\text{proxy},k} \\ d_{\text{proxy},k+1} \\ \vdots \\ d_{\text{proxy},k+N-1} \end{bmatrix}. \tag{2.64}$$

Thus, the DL is trained to learn the mapping $\Phi_k \longmapsto D_{\text{proxy},k}$. Once trained, the DL provides the estimated mismatch sequence $\hat{D}_{m,k}$ online from the most recent input window, and this estimate is used by both the compensated observer and the MPC prediction model.

2.5.2. Benchmarking Protocol for Causal Sequence Model Selection

The mismatch-preview estimator must satisfy the causality requirement of the real-time ANC controller. At sampling instant k, the estimator can only use measured and known information available up to time $k-1$. Similar causality and latency constraints have been emphasized in deep-learning-based ANC studies [38, 39]. Therefore, the learning problem is formulated as a causal sequence-to-vector regression problem,

$$\hat{D}_{m,k} = f_\theta(\Phi_k), \tag{2.65}$$

where $f_\theta(\cdot)$ denotes a trainable sequence model with parameters θ. The input window Φ_k contains past feature vectors, while the output $\hat{D}_{m,k}$ is the predicted mismatch sequence over the MPC prediction horizon.

Six causal sequence architectures are benchmarked under the same evaluation protocol. The candidate models are grouped into recurrent, convolutional, and attention-based architectures, as shown in Figure 2.3. These families represent different mechanisms for modeling temporal dependence: recursive memory, temporal filtering, and adaptive attention over past samples [2, 25, 34].

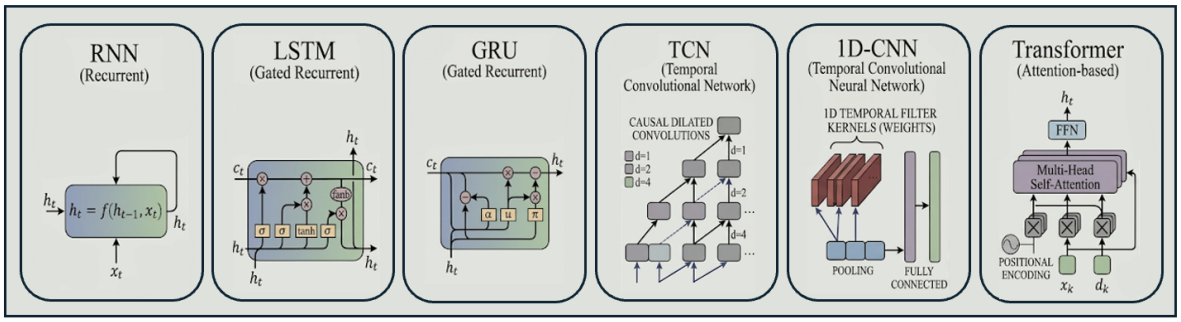


Figure 2.3: Candidate causal sequence architectures considered for mismatch-preview estimation. The benchmark includes recurrent models, convolutional temporal models, and an attention-based model.

The recurrent group includes the standard recurrent neural network (RNN), long short-term memory (LSTM), and gated recurrent unit (GRU). Recurrent models are natural candidates for dynamic systems because they process the input sequence sequentially and update an internal hidden state [25]. A standard RNN is included as the simplest recurrent baseline:

$$h_t = \sigma\left(W_\phi \phi_t + W_h h_{t-1} + b_h\right), \qquad \hat{D}_{m,k} = W_o h_{k-1} + b_o. \tag{2.66}$$

The advantage of the RNN is its simplicity and low computational cost. Its main limitation is that long temporal dependencies may be difficult to learn because the hidden state is repeatedly overwritten.

The LSTM is included because its gated memory structure is designed to retain information over longer temporal ranges, which is useful when the mismatch depends on delayed acoustic and control effects [13, 16]. Its update equations are

$$\begin{aligned} i_t &= \sigma(W_i[\phi_t, h_{t-1}] + b_i), \\ f_t &= \sigma(W_f[\phi_t, h_{t-1}] + b_f), \\ o_t &= \sigma(W_o[\phi_t, h_{t-1}] + b_o), \\ \tilde{c}_t &= \tanh(W_c[\phi_t, h_{t-1}] + b_c), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\ h_t &= o_t \odot \tanh(c_t). \end{aligned} \tag{2.67}$$

The LSTM can model longer memory effects than a standard RNN, but it has more parameters and a higher computational cost.

The GRU is used as a lighter gated recurrent alternative [6, 7]. It combines memory update and reset mechanisms as

$$\begin{aligned} z_t &= \sigma(W_z[\phi_t, h_{t-1}] + b_z), \\ r_t &= \sigma(W_r[\phi_t, h_{t-1}] + b_r), \\ \tilde{h}_t &= \tanh(W_g[\phi_t, r_t \odot h_{t-1}] + b_g), \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \end{aligned} \tag{2.68}$$

The GRU often provides a good compromise between modeling capacity and complexity, although it may be less expressive than an LSTM when a separate long-term memory state is beneficial.

The convolutional group includes a causal one-dimensional CNN and a temporal convolutional network (TCN). These models use temporal filters to extract patterns from the input window. For a causal convolutional layer,

$$h_t^{(\ell)} = \sigma\left(\sum_{j=0}^{K-1} W_j^{(\ell)} h_{t-j}^{(\ell-1)} + b^{(\ell)}\right), \tag{2.69}$$

where K is the kernel length and ℓ denotes the layer index. A 1D-CNN is included because it can efficiently learn local temporal features in the mismatch proxy. Its limitation is that its memory length is restricted by the convolutional receptive field, which depends on the kernel size and number of layers.

The TCN extends the causal CNN by using dilated convolutions [2]:

$$h_t^{(\ell)} = \sigma\left(\sum_{j=0}^{K-1} W_j^{(\ell)} h_{t-d_\ell j}^{(\ell-1)} + b^{(\ell)}\right), \tag{2.70}$$

where d_ℓ is the dilation factor. The TCN can cover a longer history with fewer layers than a standard CNN and is easier to parallelize than recurrent models. Its main limitation is that the useful temporal range must still be selected through the kernel size, number of layers, and dilation pattern.

The attention-based group is represented by a causal Transformer model. The Transformer is included because its self-attention mechanism can assign different weights to different samples in the input window, which is useful when the relevant mismatch information is not limited to nearby samples [34]. For a causal self-attention layer, the

attention weight from time t to a past time $\tau \leq t$ is

$$\alpha_{t,\tau} = \frac{\exp\left(q_t^\top k_\tau / \sqrt{d_a}\right)}{\sum_{\ell \leq t} \exp\left(q_t^\top k_\ell / \sqrt{d_a}\right)}, \qquad z_t = \sum_{\tau \leq t} \alpha_{t,\tau} v_\tau. \tag{2.71}$$

The Transformer is useful for adaptive long-range temporal weighting, but it generally requires more trainable parameters, more data, and greater computational resources than recurrent or convolutional models.

To ensure a level playing field, the task-defining quantities are kept fixed across all candidate models. These include the dataset, training/validation/test split, feature vector ϕ_k, history length S_L, prediction horizon N, normalization procedure, supervised target, training loss, evaluation metrics, optimizer type, maximum number of training epochs, batch-size setting, and early-stopping rule. Normalization statistics are computed only from the training set and then reused for validation and test data. When computational burden is evaluated, inference timing is measured under the same hardware and inference-batch conditions.

All models are trained with the same mean-squared prediction loss over the full prediction horizon,

$$\mathcal{L}(\theta) = \frac{1}{N_{\text{tr}}} \sum_{i=1}^{N_{\text{tr}}} \frac{1}{N} \left\| f_\theta(\Phi_i) - D_{\text{proxy},i} \right\|_2^2 . \tag{2.72}$$

Early stopping is applied using the validation loss in order to reduce overfitting [12].

Only model-specific capacity and training hyperparameters are allowed to vary across architectures. The learning rate and dropout rate are treated as shared tunable hyperparameters for all candidate models, while the remaining hyperparameters depend on the structure of each architecture. The architecture-dependent search dimensions are summarized in Table 2.1. Hyperparameters are optimized on the validation set using random search with the same search budget for each architecture. Random search is used because it is effective for neural-network hyperparameter optimization when only a subset of hyperparameters strongly affects performance [3]. The fixed search budget reduces comparison bias and improves reproducibility [4, 9]. The test set is used only after model selection.

Table 2.1: Architecture-dependent hyperparameters considered in the causal sequence model benchmark.

Model	Architecture-dependent hyperparameters
RNN	Number of layers, hidden size
LSTM	Number of layers, hidden-size configuration
GRU	Number of layers, hidden size
1D-CNN	Number of convolutional layers, number of channels, kernel size
TCN	Number of layers, number of channels, kernel size, dilation pattern
Transformer	Number of encoder layers, attention dimension, number of heads, feed-forward dimension

The model comparison is based on both average and worst-case prediction performance. The following metrics are evaluated on the validation set during model selection and on the held-out test set only for the final selected model. For a generic evaluation set with N_{ev} samples, the root-mean-square error is defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_{\mathrm{ev}} N} \sum_{i=1}^{N_{\mathrm{ev}}} \left\| \hat{D}_{m,i} - D_{\mathrm{proxy},i} \right\|_2^2}. \tag{2.73}$$

The maximum absolute error is defined as

$$\mathrm{MaxAE} = \max_{i=1,\ldots,N_{\mathrm{ev}}} \max_{j=1,\ldots,N} \left| \hat{d}_{i,j} - d_{\mathrm{proxy},i,j} \right|. \tag{2.74}$$

The RMSE quantifies the average prediction quality over the horizon, while MaxAE captures the worst-case prediction error. The final architecture is selected based on the trade-off between low RMSE, low MaxAE, and acceptable computational complexity for causal implementation. The numerical benchmark setup and results are reported in the numerical simulation chapter.

Algorithm 2.1 Offline data generation and causal sequence-model selection

Define the nominal plant model, nonlinear mismatch model, fixed history length S_L, prediction horizon N, candidate architectures $\mathcal{A}$, and hyperparameter search spaces $\mathcal{H}_a$.
Generate offline excitation signals $\{u_i^p(k)\}_{i=1}^{N_{\text{seq}}}$ with different frequency contents, amplitudes, and temporal envelopes.
for each excitation sequence $i = 1, \ldots, N_{\text{seq}}$ **do**
    Simulate the nonlinear ANC system and the nominal mismatch-free model using the same generated inputs.
    Log the available signals $u^p(k)$, $u^s(k)$, and $y(k)$.
    Compute the nominal output $y_{\text{nom}}(k)$ and the output mismatch $\Delta y(k) = y(k) - y_{\text{nom}}(k)$.
    Reconstruct the offline mismatch proxy $d_{\text{proxy}}(k)$ using the nominal model matrices and the known direction B^d.
    Construct the causal feature vectors ϕ_k from the logged signals, output mismatch, and delayed proxy values.
end for
Construct causal input windows Φ_k of fixed length S_L from the feature vectors.
Construct the corresponding supervised target sequences $D_{\text{proxy},k}$.
Split the generated dataset into training, validation, and test sets at the sequence level.
Compute normalization parameters from the training set only.
Normalize the input features and target mismatch sequences using the training-set normalization parameters.
for each candidate architecture $a \in \mathcal{A}$ **do**
    for each hyperparameter configuration $h \in \mathcal{H}_a$ **do**
        Train the causal sequence model $f_\theta^{(a,h)}$ using the training set.
        Evaluate the trained model on the validation set using RMSE, maximum absolute error, and computational cost.
        Save the trained model, validation metrics, and training time.
    end for
    Select the best model for architecture a according to the validation criterion.
end for
Compare the best models from all architectures using the validation metrics.
Select the final model $f_{\theta^\star}$ based on prediction accuracy and computational cost.
Evaluate the selected model once on the held-out test set.
Save $f_{\theta^\star}$, the normalization parameters, the fixed history length S_L, the prediction horizon N, and the final test-set performance metrics.

2.6. Closed-Loop Use of the Frozen Mismatch Predictor

After model selection, the LSTM mismatch-preview predictor is used with fixed offline-trained parameters during closed-loop control. At each sampling instant, the causal feature window Φ_k is formed from the available signal history and passed through the trained predictor to obtain $\hat{D}_{m,k}$. The complete sequence $\hat{D}_{m,k}$ is inserted into the MPC prediction through $M^d\hat{D}_{m,k}$, while its first element is used in the compensated observer through $B^d\hat{d}_k$. The MPC optimization and observer correction are updated at every sampling instant, but the LSTM weights are not modified online. If the operating conditions change sufficiently that the predictor becomes inaccurate, the model can be retrained offline using newly collected data and then redeployed.

Algorithm 2.2 summarizes the closed-loop implementation with the frozen mismatch predictor. The algorithm separates the operations performed before applying the control input from the operations performed after the new residual measurement is available.

Algorithm 2.2 Closed-loop hybrid MPC–LSTM control with a frozen mismatch predictor

Load the selected offline-trained predictor $f_{\theta^\star}$, normalization parameters, fixed history length S_L, prediction horizon N, lifted matrices O, M^p, M^s, M^d, observer gain L, and mismatch direction B^d.
Initialize $\hat{x}_{c,0}$, $x_{\mathrm{nom},0}$, and the causal feature buffer. Set unavailable delayed proxy values to zero during buffer initialization.
for each sampling instant $k = 0, 1, 2, \ldots$ **do**
    Form the causal input window Φ_k from the feature buffer using Equation (2.55).
    Evaluate the fixed predictor $\hat{D}_{m,k} = f_{\theta^\star}(\Phi_k)$, and convert the output back to the model scale if target normalization is used.
    Construct the primary preview U_k^p and the compensated MPC constant term $c_{\mathrm{comp},k} = O\hat{x}_{c,k} + M^p U_k^p + M^d \hat{D}_{m,k}$.
    Compute the unconstrained control sequence $U_k^{s*} = H^{-1}(M^s)^\top Q c_{\mathrm{comp},k}$.
    Set $u_{k,\mathrm{unc}}^s = [U_k^{s*}]_1$, compute $u_k^{s,\mathrm{sat}}$ using Equation (2.19), and apply $u_k^s = u_k^{s,\mathrm{sat}}$ to the secondary source.
    Measure the residual output y_k.
    Compute $y_{\mathrm{nom},k}$ and Δy_k using Equations (2.57) and (2.58).
    Compute the observer innovation $e_{c,k}$ from Equation (2.50) and update the compensated observer using Equation (2.49), with $\hat{d}_k = \hat{d}_k^{(1)}$.
    Update the nominal mismatch-free model using Equation (2.56).
    if $k \geq 1$ **then**
        Reconstruct $e_{\mathrm{proxy},k-1}$ and $d_{\mathrm{proxy},k-1}$ using Equations (2.61) and (2.63).
    else
        Set $d_{\mathrm{proxy},k-1} = 0$ for buffer initialization.
    end if
    Construct the new causal feature vector ϕ_k by shifting Equation (2.54) from $k-1$ to k.
    Append ϕ_k to the feature buffer and discard the oldest feature vector.
end for

Figure 2.4 summarizes the complete signal flow of the proposed architecture. At each sampling instant, the measured primary input u_k^p and the applied secondary control u_k^s drive both the real system and the nominal model. The primary-input preview U_k^p, the compensated state estimate $\hat{x}_{c,k}$, and the predicted mismatch sequence $\hat{D}_{m,k}$ are supplied to the MPC controller. The measured and nominal outputs are used to form Δy_k and the delayed mismatch proxy, which are stored with the remaining causal features in the history window Φ_k. The LSTM then supplies the complete mismatch preview to the MPC

and its first element $\hat{d}_k^{(1)}$ to the compensated observer. The MPC optimization is solved at every sampling instant according to the receding-horizon principle. The LSTM predictor remains fixed during closed-loop operation.

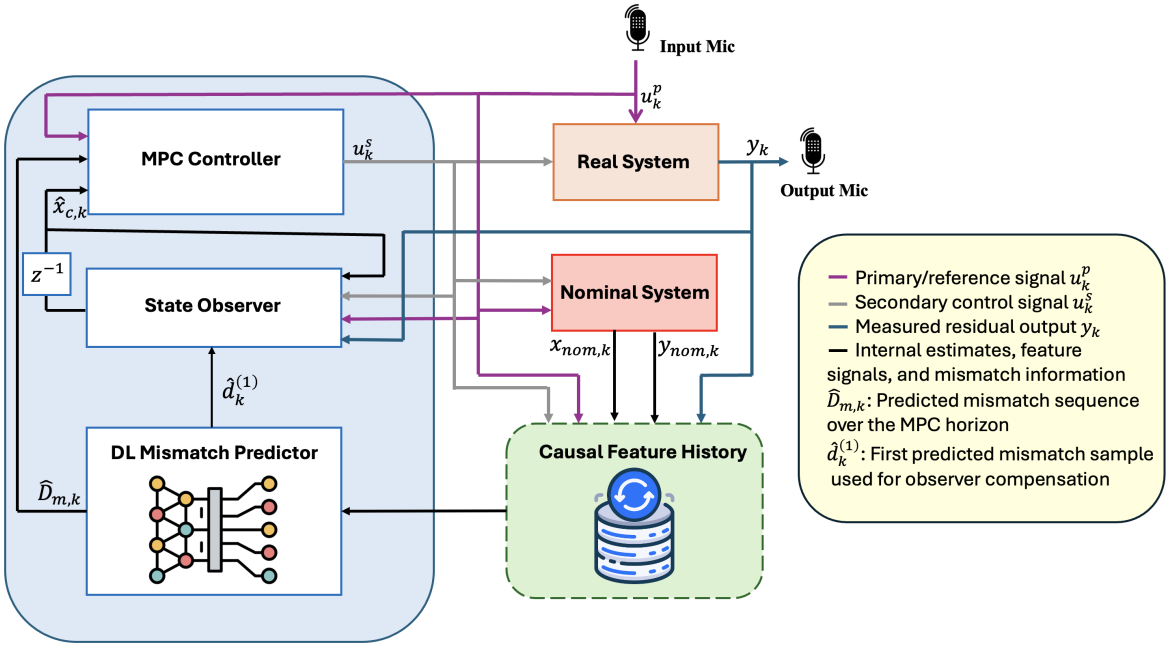


Figure 2.4: Closed-loop signal flow of the proposed hybrid MPC–LSTM controller, compensated observer, nominal model, causal feature history, and mismatch-preview predictor.

3 | Numerical Simulation

3.1. Simulation Setup and Parameters

The proposed DL-MPC framework is evaluated in simulation using the acoustic-path setting adopted in [24]. The sampling frequency is set to $f_s = 4$ kHz, and the primary and secondary acoustic path models are identified over the frequency range 100–800 Hz. The same frequency range is used for the band-limited performance evaluation. The reference disturbance signals include realistic traffic, airplane, and knock-noise recordings, together with a band-limited white-noise signal. Each input signal is normalized before simulation so that the controllers are compared under the same input-amplitude convention.

The controller parameters are selected consistently with the delayed-MPC ANC configuration in [24]. The MPC prediction horizon is set to $N = 9$, and the weighting matrices are chosen as $Q = I$ and $R = 10^{-3}I$. The secondary control signal is saturated according to $|u_k^s| \leq 1$ in order to represent actuator amplitude limits and to avoid unrealistically large control commands. For comparison, the FxNLMS controller is also implemented with two step sizes, $\mu = 0.001$ and $\mu = 0.01$, representing stable and faster-converging adaptive-control settings, respectively.

To evaluate the ability of the proposed method to compensate unmodelled dynamics, an artificial nonlinear model mismatch is introduced into the plant. The purpose of this mismatch is not to represent one unique physical nonlinearity, but to create a controlled and repeatable uncertainty that depends on the recent closed-loop history. This allows the effect of mismatch and the benefit of mismatch-preview compensation to be examined explicitly. The plant model used in the closed-loop simulation is therefore written as in Equation (2.40).

In the simulation study, the equivalent model mismatch is assumed to enter through the primary-path input channel. In the augmented ANC state-space model, B_{up} denotes the input direction associated with the primary/reference signal u_k^p. Therefore, the distur-

bance distribution matrix is selected as the normalized primary-path input direction:

$$B^d = \frac{B_{up}}{\|B_{up}\|_2}, \qquad \|B^d\|_2 = 1. \tag{3.1}$$

This choice is physically motivated by the feedforward ANC structure, where the reference microphone captures the unwanted noise before cancellation, and the primary disturbance at the error microphone is produced by propagation through the primary acoustic path [17, 22].

The mismatch is generated using a bounded quadratic NARX-type model. Let

$$\xi_k = \begin{bmatrix} u^p_{k-1} & \cdots & u^p_{k-\ell} & u^s_{k-1} & \cdots & u^s_{k-\ell} & y_{k-1} & \cdots & y_{k-\ell} \end{bmatrix}^\top \tag{3.2}$$

collect the most recent ℓ samples of the primary input, secondary control input, and residual error signal. The nonlinear mismatch is computed as

$$v_k = c_0 + c_1^\top \xi_k + \xi_k^\top H_{\mathrm{pol}} \xi_k, \tag{3.3}$$

$$d_k = \bar{d} \tanh\left(\frac{v_k}{\beta}\right), \tag{3.4}$$

where $c_0 \in \mathbb{R}$, $c_1 \in \mathbb{R}^{3\ell}$, and $H_{\mathrm{pol}} \in \mathbb{R}^{3\ell \times 3\ell}$ are fixed across all simulations. The NARX memory length is set to $\ell = 10$, the mismatch bound is $\bar{d} = 1.2$, and the saturation parameter is $\beta = 0.5$. The hyperbolic tangent bounds the mismatch amplitude, while the dependence on ξ_k makes the mismatch nonlinear and history-dependent. The same mismatch model and parameters are used for all controllers. Since the regressor ξ_k contains the closed-loop variables u^s and y, the mismatch sequence is recomputed separately for each controller using its own closed-loop trajectory.

3.2. Benchmarking of Mismatch-Preview Estimators

This section reports the numerical benchmark used to select the causal sequence model adopted as the mismatch-preview estimator. The benchmark was designed to evaluate prediction accuracy, robustness to non-stationary excitation, and computational suitability for later use in the compensated observer and MPC prediction model.

3.2.1. Benchmark Dataset and Excitation Design

The benchmark data were generated offline using the nominal state-space model and the nonlinear mismatch representation introduced in Chapter 2. The primary input u_k^p was synthetically generated as randomized band-limited broadband excitation. In total, 100 independent excitation sequences were generated, each with duration 15 s. For each sequence, the frequency passband and amplitude profile were randomized to expose the learning models to different disturbance characteristics. Random frequency-band selections changed the spectral content, while local amplitude bursts and attenuations introduced non-stationary behavior during the simulation.

For each excitation record, the nonlinear ANC system was simulated together with the nominal mismatch-free model under the same generated input sequences. The difference between the measured output of the nonlinear system and the corresponding nominal output was then used to reconstruct the mismatch proxy according to the procedure described in Section 2.5.1. This procedure ensures that the learning target is tied to the same equivalent disturbance channel used later by the compensated observer and MPC formulation, rather than being treated as an arbitrary data-fitting signal.

3.2.2. Hyperparameter Search Space

The collected records were normalized and then divided into training, validation, and test subsets before model fitting using a fixed 70%–20%–10% split, respectively. To ensure a fair numerical comparison, all candidate architectures were evaluated on this same split with the same normalized feature vectors, history length, prediction horizon, training objective, stopping rule, and inference hardware. In the simulations, the history length was fixed to $S_L = 32$, the prediction horizon N was set equal to the MPC prediction horizon, and the networks were trained using the mean-squared prediction loss over the full horizon. Adam was used as the optimizer, with a maximum of 500 epochs and a mini-batch size of 128. For each candidate architecture, hyperparameters were optimized using random search with 80 trials, so that all model families were assigned the same search budget.

Learning rate and dropout were treated as shared tunable training hyperparameters because they affect all six model classes. The candidate learning rates were

$$\eta \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 2\times10^{-1}\}, \tag{3.5}$$

and the candidate dropout probabilities were

$$p_{\text{drop}} \in \{0, 0.05, 0.10, 0.20, 0.30\}. \tag{3.6}$$

The maximum number of training epochs was set to 500. After the best validation configuration was selected for each architecture, the corresponding model was retrained using 10 independent random initializations to evaluate repeatability.

The remaining hyperparameters were architecture dependent and were therefore tuned separately for each model family. The model-specific search space is summarized in Table 3.1.

Table 3.1: Model-specific hyperparameter search space used for causal sequence model benchmarking.

Model	**Architecture-specific hyperparameters**
RNN	Number of recurrent layers: $\{1, 2, 3\}$. Hidden units: 1 layer: $\{32, 64, 128\}$, 2 layers: $\{(32, 32), (64, 64), (128, 128), (32, 64), (64, 128)\}$, 3 layers: $\{(32, 32, 32), (64, 64, 64), (128, 128, 128), (32, 64, 128)\}$.
LSTM	Number of LSTM layers: $\{1, 2, 3\}$. Hidden units: 1 layer: $\{32, 64, 128\}$, 2 layers: $\{(32, 32), (64, 64), (128, 128), (32, 64), (64, 128)\}$, 3 layers: $\{(32, 32, 32), (64, 64, 64), (128, 128, 128), (32, 64, 128)\}$.
GRU	Number of GRU layers: $\{1, 2, 3\}$. Hidden units: 1 layer: $\{32, 64, 128\}$, 2 layers: $\{(32, 32), (64, 64), (128, 128), (32, 64), (64, 128)\}$, 3 layers: $\{(32, 32, 32), (64, 64, 64), (128, 128, 128), (32, 64, 128)\}$.
1D-CNN	Number of causal convolutional layers: $\{1, 2, 3, 4\}$. Number of channels: $\{16, 32, 64, 128\}$. Kernel size: $\{3, 5, 7\}$. Fully connected output size: $\{32, 64, 128\}$.

TCN	Number of temporal blocks: $\{2, 3, 4\}$. Number of channels: $\{16, 32, 64, 128\}$. Kernel size: $\{2, 3, 5\}$. Dilation pattern: $\{(1, 2), (1, 2, 4), (1, 2, 4, 8)\}$.
Transformer	Number of encoder layers: $\{1, 2, 3\}$. Model dimension: $\{32, 64, 128\}$. Number of attention heads: $\{2, 4, 8\}$, subject to divisibility by the model dimension. Feed-forward dimension: $\{2d_{\text{model}}, 4d_{\text{model}}\}$.

3.2.3. Benchmark Results and Model Selection

Table 3.2 summarizes the numerical benchmark results. The reported values include the normalized validation RMSE, validation MaxAE, computational burden, and total training time for each candidate architecture.

Table 3.2: Benchmark summary of candidate causal sequence models.

Model	**RMSE**	**MaxAE**	**CB (ms/window)**	**TT (s)**
1D-CNN	0.01772	0.18634	0.052208	956.44
TCN	0.02650	0.26472	0.072761	1549.00
RNN	0.06380	0.51287	0.446640	2008.00
LSTM	0.00610	0.15810	0.099017	703.64
GRU	0.00770	0.27356	0.067378	718.68
Transformer	0.06010	0.48265	0.071690	1109.00

The convolutional models provide relatively low computational burden, especially the 1D-CNN. However, their prediction accuracy is lower than that of the best gated recurrent models. The standard RNN and Transformer show larger RMSE and MaxAE values, indicating weaker robustness for the considered mismatch-preview task. Among all candidates, the LSTM provides the best overall trade-off. It achieves the lowest normalized RMSE of 0.00610 and the lowest MaxAE of 0.15810, while maintaining a moderate computational burden of 0.099017 ms/window and the shortest training time of 703.64 s.

Figure 3.1 further compares the candidate predictors in terms of robustness and efficiency on the validation data. Figure 3.1(a) shows the validation RMSE distribution across random initializations, while Figure 3.1(b) compares the validation MaxAE with the training time. The LSTM occupies the most favorable region because it combines the lowest MaxAE with a low median RMSE and moderate computational cost.

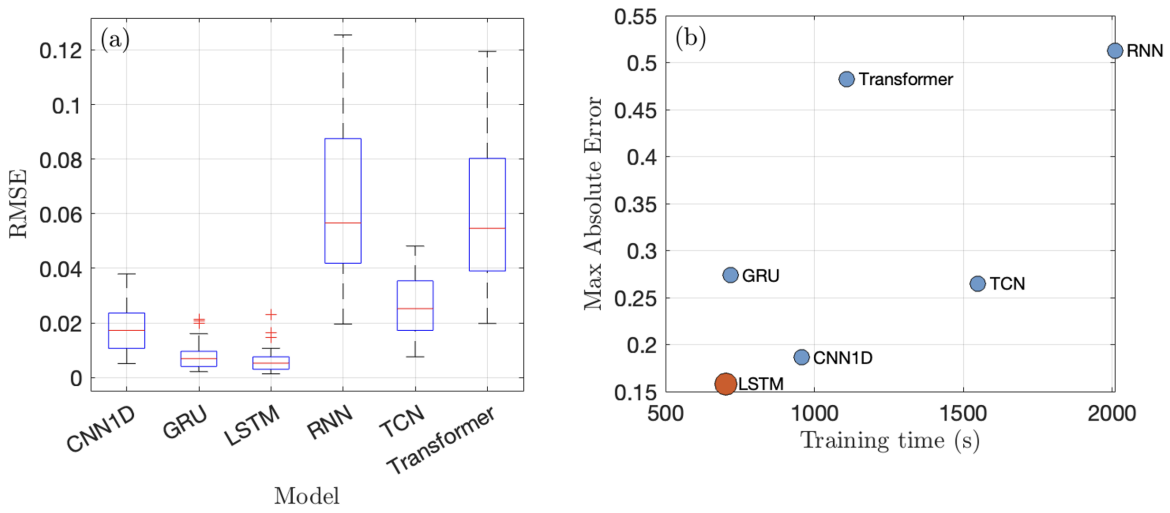


Figure 3.1: Validation performance of the candidate mismatch-preview predictors: (a) RMSE distribution across random initializations; (b) MaxAE versus training time.

The selected LSTM was also compared with a persistence baseline, in which the future mismatch sequence is approximated using the most recent available proxy value. After inverse normalization, the LSTM reached a mean test RMSE of 0.0256 in physical units, compared with 0.1228 for the persistence baseline. This corresponds to a 79.2% reduction in prediction error. Furthermore, the multi-step RMSE remained bounded over the full prediction horizon, supporting the suitability of the LSTM for forecasting the future mismatch sequence required by the proposed MPC framework.

Based on these results, the LSTM architecture is selected as the final mismatch-preview estimator in the remaining simulations.

3.3. Noise Reduction Performance

The proposed Hybrid MPC-LSTM controller uses the trained causal sequence model selected in Section 3.2 to predict the future mismatch sequence. This predicted sequence is included in the MPC prediction model through the lifted disturbance matrix M^d, and its first sample is used for observer compensation. During closed-loop simulation, the LSTM predictor is used with fixed offline-trained parameters. The MPC without LSTM is obtained by setting $\hat{D}_{m,k} = 0$, so that the difference between the two MPC cases directly reflects the contribution of the learned mismatch preview.

The compared cases are ANC off, FxNLMS, MPC without mismatch preview, and the proposed Hybrid MPC-LSTM controller. For the FxNLMS algorithm, the adaptive filter

length is set to $K = 64$, allowing relatively fast convergence while keeping the adaptive filter order fixed across the simulations. The default step size is chosen as $\mu = 0.001$ to ensure stable convergence. A larger step size of $\mu = 0.01$ is also considered in Table 3.3 to represent accelerated convergence behavior. Performance is evaluated using broadband residual behavior and band-limited noise reduction over 100–800 Hz, both computed relative to the ANC-off case.

3.3.1. Time-Domain Residual Response

Figure 3.2 presents the time-domain residual error signals obtained using the different ANC algorithms. For traffic noise in Figure 3.2(a), the ANC-off response contains continuous broadband fluctuations. FxNLMS remains close to the uncontrolled response, indicating that the adaptive filter does not suppress the broadband disturbance effectively under the nonlinear mismatch. MPC reduces the residual signal to a narrow band around zero, and the Hybrid MPC-LSTM controller further reduces the remaining fluctuations. This indicates that the learned mismatch preview improves the prediction model used by MPC rather than merely smoothing the output. Notably, MPC and Hybrid MPC-LSTM operate without requiring a convergence process, whereas FxNLMS requires a convergence period and multiple iterations to update its adaptive filter

For airplane noise in Figure 3.2(b), the uncontrolled amplitude is relatively small during the first part of the record and increases after approximately $t = 4$ s, with larger fluctuations and peaks around $t = 5.5$–7.3 s. FxNLMS follows this growth and leaves a visible residual signal. MPC remains close to zero even when the disturbance amplitude increases, while the Hybrid MPC-LSTM response is the smallest and smoothest among the controlled cases. This behavior shows that the proposed controller maintains attenuation when the signal amplitude and spectral content vary during the simulation.

For knock noise in Figure 3.2(c), the disturbance is dominated by abrupt transient bursts. FxNLMS leaves several visible residual peaks, while MPC attenuates most of the burst energy. Hybrid MPC-LSTM gives the lowest residual amplitude, indicating improved robustness to impulsive disturbances.

For white noise in Figure 3.2(d), FxNLMS remains close to the uncontrolled broadband response. MPC provides a substantial reduction, and Hybrid MPC-LSTM produces the smallest residual fluctuations throughout the record.

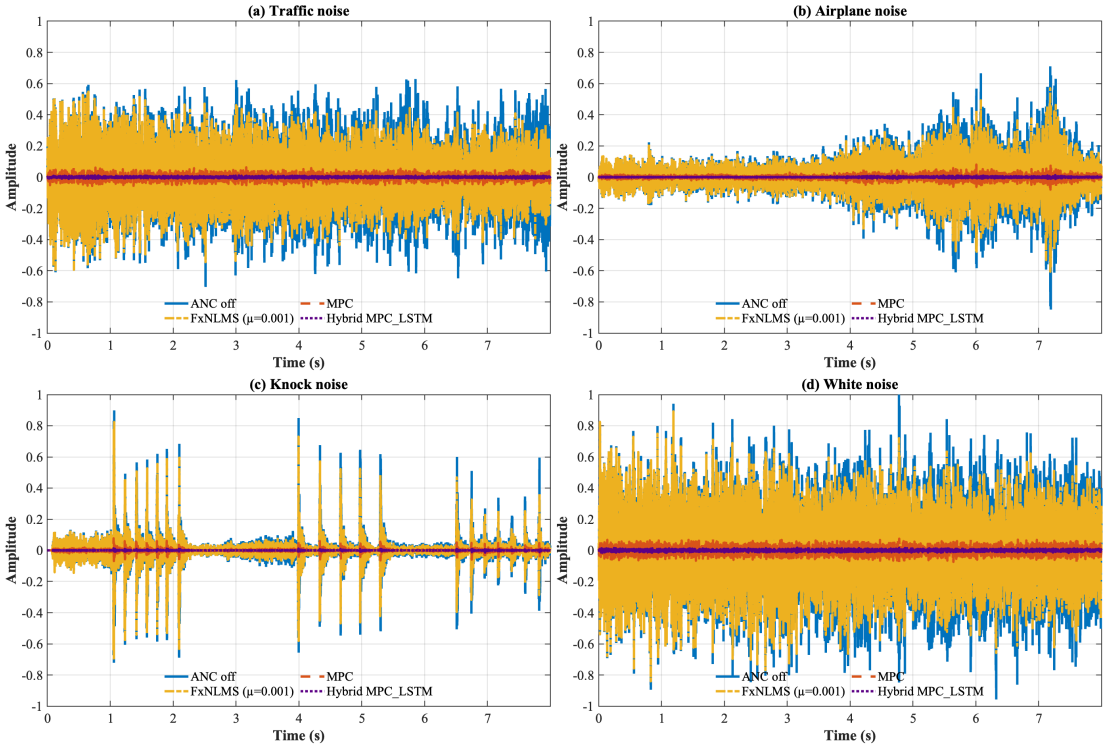


Figure 3.2: Time-domain residual error signals for different disturbance types.

3.3.2. Time–Frequency Residual Distribution

Figure 3.3 illustrates the Short-Time Fourier Transform (STFT) spectra of the residual error signals. All spectrograms are plotted using the same STFT parameters and color scale, so the color intensity can be compared directly across algorithms. In the ANC-off and FxNLMS cases, traffic noise and white noise show continuous broadband energy over the controlled band. Airplane noise shows a clear increase in energy after approximately $t = 4$ s, consistent with the amplitude growth observed in the time-domain plot. Knock noise produces vertical high-energy bands at the burst times, reflecting its impulsive nature.

For all four disturbances, FxNLMS leaves a time-frequency pattern similar to the ANC-off case, especially in the 100–800 Hz band. MPC suppresses a large portion of this energy, but visible residual components remain, particularly at low and mid frequencies and at the knock transient instants. The Hybrid MPC-LSTM controller produces the darkest spectrograms among the controlled methods. For traffic and white noise, the broadband residual energy is reduced over almost the full time interval. For airplane noise, the high-energy region after $t = 4$ s is strongly attenuated. For knock noise, the vertical burst components are still identifiable because the excitation is impulsive, but their intensity is

substantially reduced compared with MPC and FxNLMS. These observations show that mismatch-preview compensation improves not only the average residual level but also the distribution of residual energy over time and frequency.

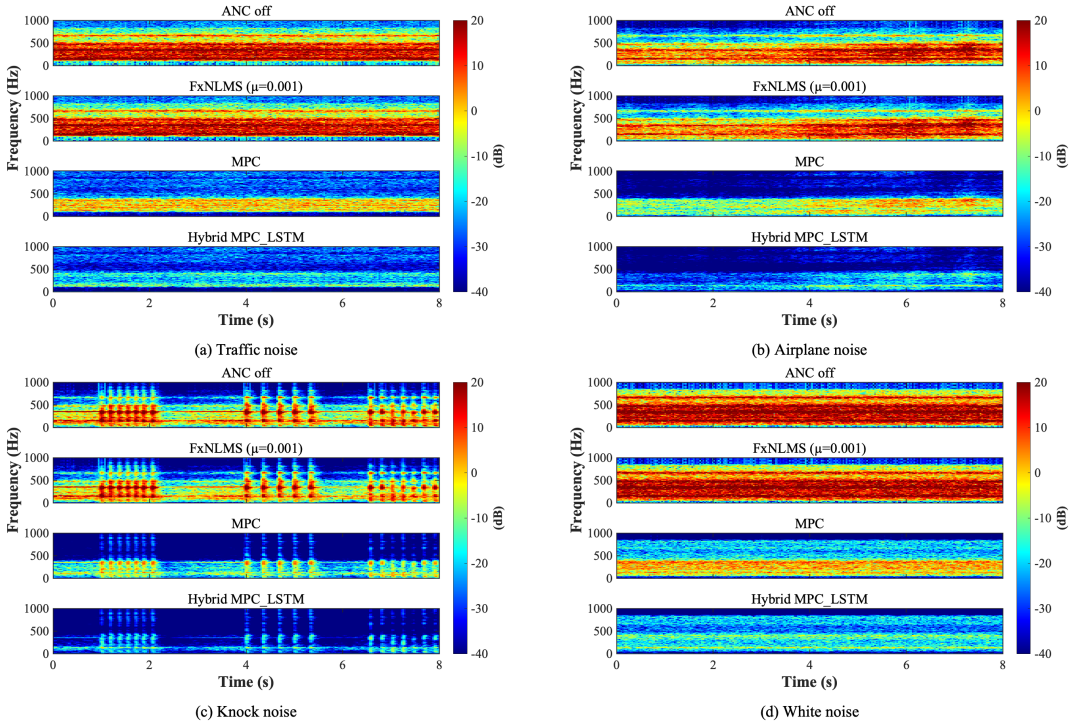


Figure 3.3: Short-Time Fourier Transform spectra of the residual error signals for different disturbance types. All spectrograms are plotted using the same STFT parameters and the same color scale.

3.3.3. Time-Varying Attenuation

Figure 3.4 shows the time-varying attenuation calculated every second. For the one-second interval $\mathcal{I}_r$, it is computed as

$$\mathrm{ANR}(r) = 10 \log_{10} \left(\frac{\sum_{k \in \mathcal{I}_r} y_k^2 + \epsilon}{\sum_{k \in \mathcal{I}_r} y_{\mathrm{off},k}^2 + \epsilon} \right). \tag{3.7}$$

Here, y_k is the controlled residual error signal, $y_{\mathrm{off},k}$ is the ANC-off residual error signal, and $\epsilon > 0$ avoids division by zero. More negative values indicate stronger attenuation.

For traffic noise, shown in Figure 3.4(a), FxNLMS remains close to 0 dB and gradually reaches only a few decibels of attenuation, indicating limited suppression of the broadband disturbance. The MPC controller provides a much larger and nearly constant reduction,

with ANR values close to −20 dB over most of the record. The Hybrid MPC-LSTM controller gives the lowest curve throughout the simulation, remaining approximately in the −34 to −36 dB range. This shows that the learned mismatch preview improves attenuation even when the traffic signal has broadband and slowly varying spectral content.

For airplane noise in Figure 3.4(b), FxNLMS again provides only weak attenuation, while the MPC curve stays around −18 to −20 dB. The Hybrid MPC-LSTM controller achieves a substantially lower ANR than MPC for the full duration. A small loss of attenuation is visible around the middle of the record, where the Hybrid MPC-LSTM curve rises from approximately −33 dB toward −30 dB, but it remains clearly below the MPC curve and later returns to stronger attenuation. This behavior indicates that the proposed controller remains robust when the disturbance amplitude and spectral content vary with time.

For knock noise in Figure 3.4(c), the disturbance contains more abrupt transient components. FxNLMS remains close to the ANC-off reference and therefore provides limited reduction. MPC maintains an attenuation of roughly −18 to −20 dB, but its ANR becomes slightly less negative near the end of the record. The Hybrid MPC-LSTM controller achieves the strongest attenuation for most windows, approximately between −31 and −34 dB. Although the attenuation decreases slightly after the later transient intervals, it remains well below the MPC curve, showing better robustness to impulsive changes.

For white noise in Figure 3.4(d), the broadband disturbance is more stationary. FxNLMS gives only small attenuation, MPC remains close to −20 dB, and the Hybrid MPC-LSTM controller stays near −34 to −35 dB throughout the simulation. The small variation of the Hybrid MPC-LSTM curve indicates stable performance under broadband excitation. Overall, Figure 3.4 confirms that the proposed controller consistently improves the time-varying attenuation compared with MPC and FxNLMS across continuous, non-stationary, impulsive, and broadband disturbance cases.

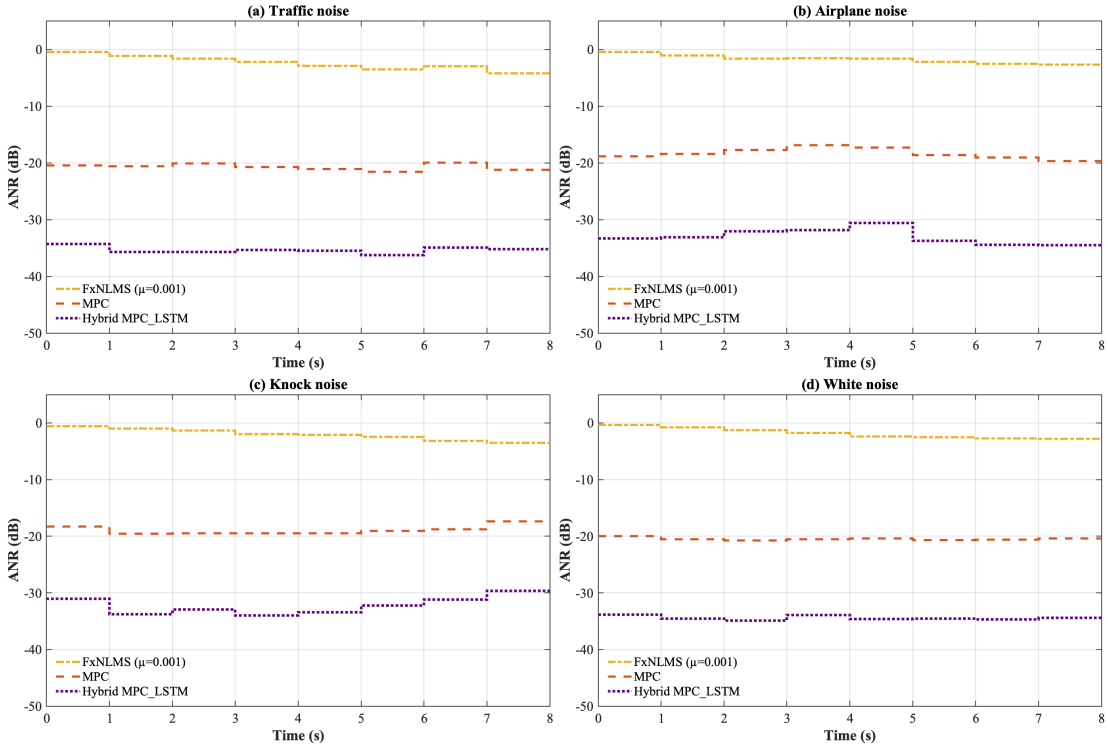


Figure 3.4: Time-varying attenuation of the residual error signals computed over short time windows. More negative ANR values indicate stronger attenuation relative to the ANC-off case.

3.3.4. Power-Spectral-Density Analysis

Figure 3.5 presents the power spectral density (PSD) of the residual error signals. For traffic noise in Figure 3.5(a), the ANC-off and FxNLMS spectra almost overlap across much of the controlled band, with pronounced components below approximately 450 Hz. MPC lowers the PSD by a large margin in this region, and Hybrid MPC-LSTM provides an additional reduction, especially between approximately 100 and 400 Hz, where most of the traffic-noise energy is concentrated.

For airplane noise in Figure 3.5(b), the uncontrolled spectrum has strong low- and mid-frequency components, with dominant peaks below approximately 400 Hz. FxNLMS again remains close to the ANC-off spectrum. MPC suppresses the main components, while Hybrid MPC-LSTM produces the lowest PSD over most of the 100–800 Hz range. The separation between MPC and Hybrid MPC-LSTM is particularly visible below 400 Hz, which is also the range where the time-domain signal shows the largest amplitude growth.

For knock noise in Figure 3.5(c), the spectrum is less smooth because the disturbance is generated by intermittent transients. ANC off and FxNLMS retain strong components

around the low- and mid-frequency peaks. MPC reduces these components substantially, but residual peaks remain. Hybrid MPC-LSTM gives the lowest PSD over most of the controlled band and reduces the dominant transient-related components more strongly than MPC.

For white noise in Figure 3.5(d), the uncontrolled and FxNLMS spectra are broadband and remain high over the controlled band. MPC reduces the PSD across a wide frequency range, while Hybrid MPC-LSTM achieves a further broadband reduction from approximately 100 to 800 Hz. Above the controlled band, all methods approach very low PSD values, which is expected because the controller and performance evaluation are focused mainly on the identified 100–800 Hz range.

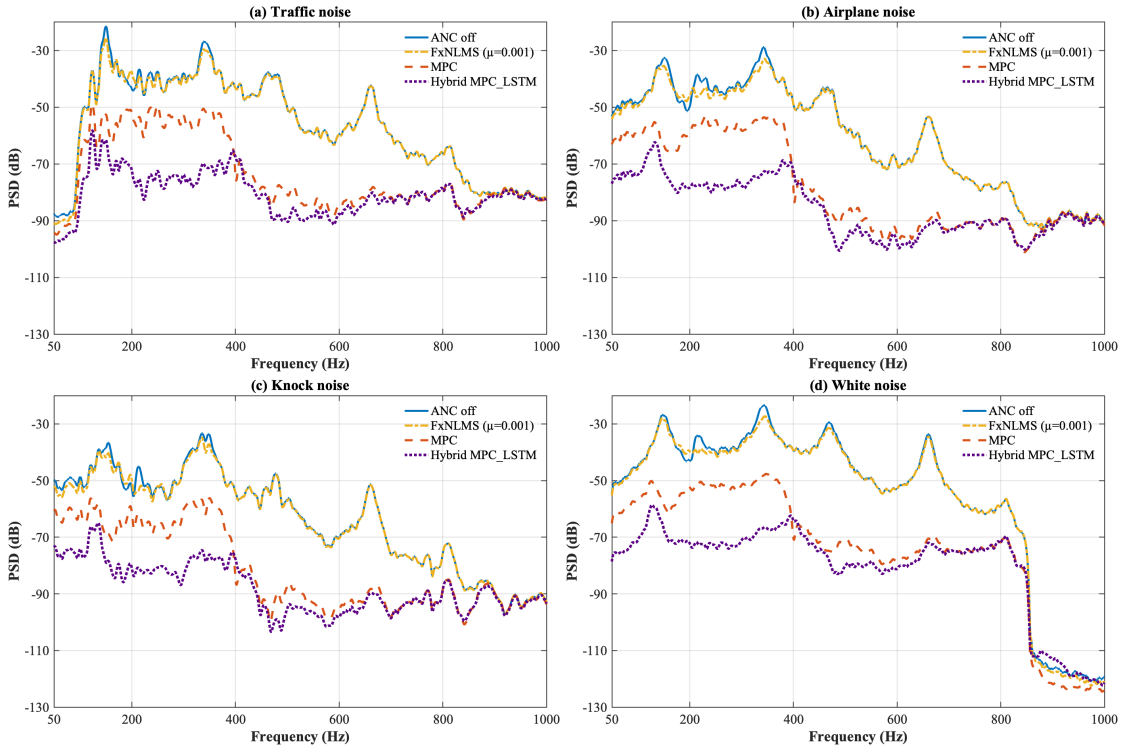


Figure 3.5: Power spectral density of the residual error signals for different disturbance types.

3.3.5. Quantitative Band-Limited Comparison

The quantitative band-limited reductions are summarized in Table 3.3. FxNLMS with $\mu = 0.001$ achieves only 1.8129–2.3216 dB reduction, confirming the limited attenuation visible in the time-domain, STFT, ANR, and PSD plots. Increasing the step size to $\mu = 0.01$ improves FxNLMS to 6.4260–7.7223 dB, but it remains far below the predictive controllers. The MPC controller achieves much higher reductions, ranging from

18.9880 dB for airplane noise to 20.6470 dB for traffic noise. The proposed Hybrid MPC-LSTM controller provides the best performance for every disturbance type, with reductions of 35.5020 dB for traffic noise, 33.8600 dB for airplane noise, 33.5420 dB for knock noise, and 34.6470 dB for white noise. Relative to MPC, the learned mismatch-preview compensation adds 13.6–14.9 dB of additional attenuation, demonstrating that the LSTM predictor improves the robustness of MPC under the artificially introduced nonlinear NARX-type mismatch.

Algorithm	**Traffic**	**Airplane**	**Knock**	**White**
FxNLMS ($\mu = 0.001$)	2.3216	2.2337	1.8129	1.8792
FxNLMS ($\mu = 0.01$)	7.6370	7.7223	6.4260	6.8361
MPC	20.6470	18.9880	19.9510	20.5690
Hybrid MPC-LSTM	35.5020	33.8600	33.5420	34.6470

Table 3.3: Band-limited noise reduction in the 100–800 Hz frequency range for different disturbance types. Values are reported in dB relative to the ANC-off case.

3.3.6. Closed-Loop Mismatch-Prediction Tracking

Figure 3.6 shows the closed-loop behavior of the learned mismatch predictor. Since the nonlinear NARX-type mismatch is artificially introduced in simulation, the true mismatch signal $d(k)$ is available and can be compared with the first element of the LSTM-predicted mismatch sequence. The plotted estimate corresponds to the first predicted sample, $\hat{d}(k) = \hat{D}_{m,k}(1)$, which is the term directly used in the observer compensation and in the first step of the MPC mismatch preview.

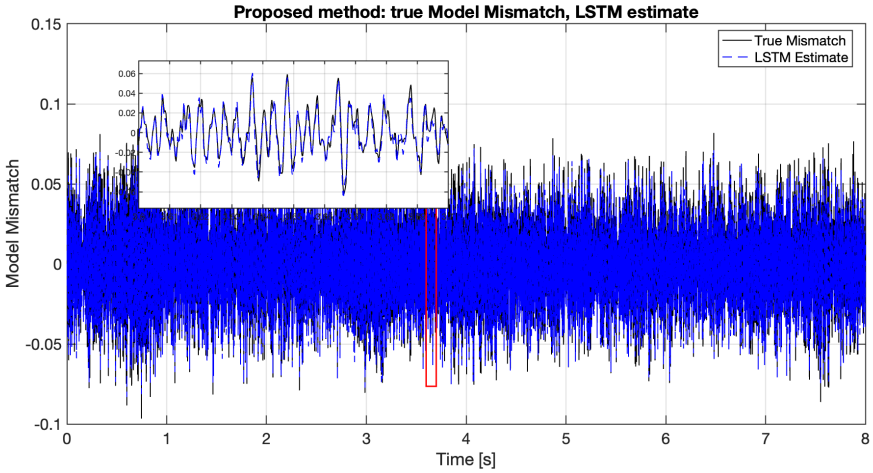


Figure 3.6: Closed-loop mismatch prediction during the proposed Hybrid MPC-LSTM simulation. The black curve shows the true nonlinear mismatch $d(k)$ generated by the NARX-type mismatch model, while the blue dashed curve shows the first predicted LSTM mismatch sample $\hat{d}(k) = \hat{D}_{m,k}(1)$. The inset highlights a short time interval to show the local tracking behavior.

4 | Implementation and Measurement-Informed Validation

This chapter presents a measurement-informed validation of the proposed controller. The primary and secondary acoustic paths are identified from a Kundt's tube setup and are used to construct measured FIR reference models and reduced-order state-space models for controller design. The reported controller comparisons are therefore model-based evaluations using experimentally identified acoustic paths, not physical closed-loop real-time experiments.

4.1. Measurement Setup

Figure 4.1 depicts the experimental configuration. A Kundt's tube was used as the test rig, with the primary and secondary acoustic paths corresponding to $P(z)$ and $S(z)$, respectively, in the feedforward ANC configuration shown in Figure 1. All parameters, including sampling rate, horizon, weighting matrices, and saturation limits, were identical to those specified in the numerical simulations of Section 3.

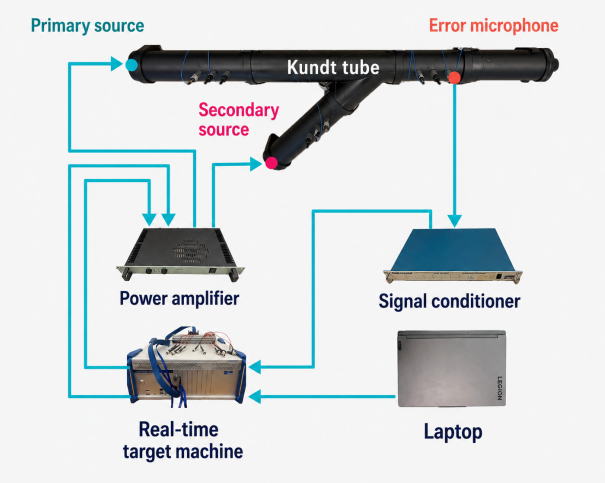


Figure 4.1: Kundt's-tube measurement setup

4.2. Identification of the Acoustic Paths

For the measurement-informed study, the primary- and secondary-path state-space models were identified independently from measurements performed on the Kundt's tube setup. The identification procedure used to obtain these models is described in this section. For each acoustic path, a broadband white-noise signal was applied to the corresponding loudspeaker as the input signal x_{in}, and the output response x_{out} was recorded by the error microphone. In order to improve the reliability of the identified models, several measurements were collected for both the primary and secondary paths. The recorded data included measurements with different durations, and the spectra were averaged over the available data segments to reduce the effect of measurement noise and experimental variability.

The raw signals were acquired at 16 kHz. For each measurement, the first 5 s and the last 5 s were removed before processing in order to eliminate the initial silent interval, triggering delay, and possible end transients. The remaining data were then divided into overlapping time segments. For each segment, the mean value was removed and the Fourier transforms of the input and output signals were computed. Based on these measurements, the frequency response function was estimated using the H_1 estimator as

$$H_{\text{meas}}(e^{j\omega}) = \frac{E\{X_{\text{in}}^*(e^{j\omega})X_{\text{out}}(e^{j\omega})\}}{E\{X_{\text{in}}^*(e^{j\omega})X_{\text{in}}(e^{j\omega})\}}, \tag{4.1}$$

where X_{in} and X_{out} denote the Fourier transforms of x_{in} and x_{out}, respectively, $(\cdot)^*$ denotes the complex conjugate, and $E\{\cdot\}$ is the averaging operator over the considered data

segments.

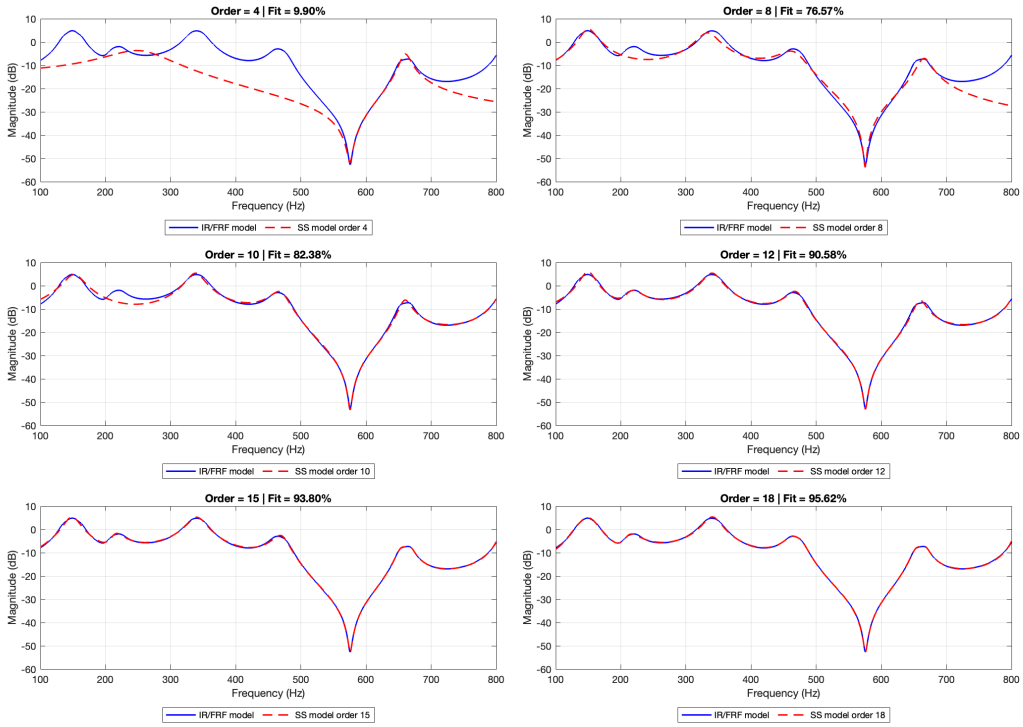
Only the frequency range from 100 Hz to 800 Hz was retained for the identification procedure, since this is the frequency band relevant to the active noise control problem considered in this work. The band-limited frequency response was transformed into the time domain to obtain the corresponding impulse-response model. Since the controller was implemented at 4 kHz, the identified impulse response was subsequently downsampled from 16 kHz to 4 kHz, and the corresponding frequency response was recomputed at the controller sampling frequency.

A parametric discrete-time model was then fitted to the measured 4 kHz frequency response using the `tfest` routine from the MATLAB System Identification Toolbox. The identified transfer function has the general form

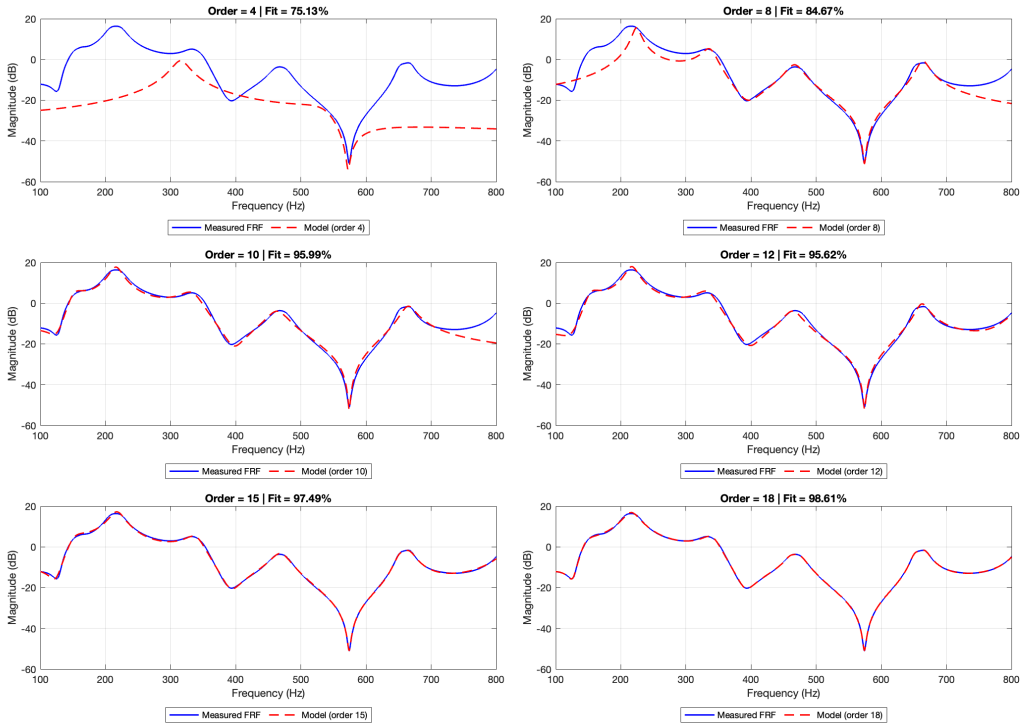
$$G(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_q z^{-q}}{1 + a_1 z^{-1} + \cdots + a_r z^{-r}}, \tag{4.2}$$

where q and r denote the numerator and denominator orders, respectively.

Several candidate model orders were tested for both the primary and secondary paths. For each order, the frequency response of the candidate state-space model was compared in the 100–800 Hz band with the measurement-derived reference FRF reconstructed from the processed impulse response, and a fit score was computed. The final order was selected automatically from the order sweep by choosing the lowest-order model whose fit was sufficiently close to the best obtained fit. This provides a compromise between model accuracy and model complexity, which is consistent with standard system-identification practice where the final model is selected by balancing accuracy and parsimony rather than by minimizing the fitting error alone [26]. The resulting order-sweep comparisons for the primary and secondary paths are presented in Figure 4.2.



(a) Primary path.



(b) Secondary path.

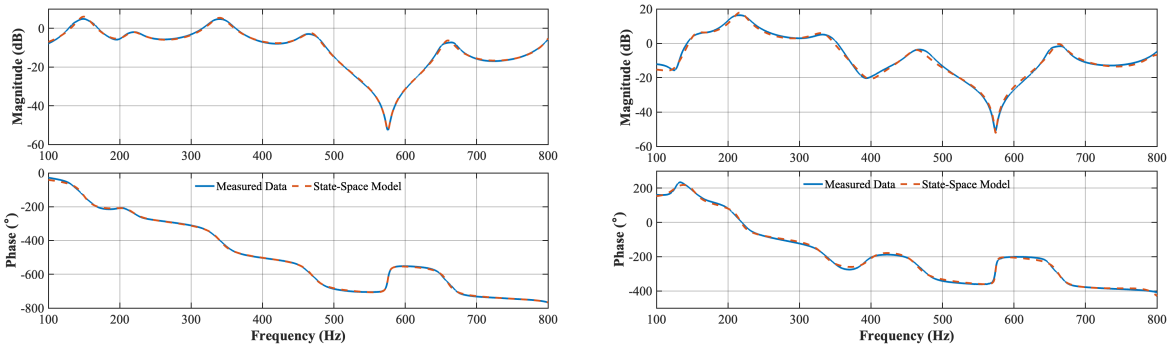
Figure 4.2: Order-sweep results for the primary and secondary acoustic-path models. The fit score was evaluated in the 100–800 Hz frequency band.

The identified transfer functions were subsequently converted into equivalent discrete-time state-space realizations using the standard relation

$$G(z) = C(zI - A)^{-1}B + D. \tag{4.3}$$

This procedure yielded the matrices (A^p, B^p, C^p, D^p) and (A^s, B^s, C^s, D^s) for the primary and secondary acoustic paths, respectively.

After the model order was selected, the frequency responses of the corresponding state-space models were compared with the measured frequency responses. Figure 4.3 shows the magnitude and phase responses of the measured data and the selected state-space models for both the primary and secondary acoustic paths. The close agreement in the 100–800 Hz frequency band indicates that the identified models capture the dominant acoustic dynamics required for control design.



(a) Primary path. (b) Secondary path.

Figure 4.3: Frequency-response comparison between the measured data and the selected state-space models. (a) Primary path. (b) Secondary path.

In addition to the dynamic behavior captured by the state-space models, the measured acoustic paths also exhibit propagation delays. These delays were estimated from the peaks of the identified impulse responses at 4 kHz. The dominant peak of the primary-path impulse response was found at $N_p = 19$ samples, while the dominant peak of the secondary-path impulse response was found at $N_s = 9$ samples. Since only the relative delay between the primary and secondary paths is relevant for the delayed MPC formulation, the secondary path was taken as the delay reference. Therefore, the secondary-path delay was assumed to be zero in the controller model, while the primary path was assigned the relative delay $N_d = N_p - N_s = 10$ samples. At the controller sampling frequency of 4 kHz, this corresponds to a delay of 2.5 ms.

With this convention, the measured acoustic paths can be represented as

$$H^p_{\text{meas}}(z) = H^p_{\text{delayless}}(z)z^{-N_d}, \qquad H^s_{\text{meas}}(z) = H^s_{\text{delayless}}(z), \tag{4.4}$$

where $H^p_{\text{delayless}}(z)$ and $H^s_{\text{delayless}}(z)$ denote the corresponding delay-free dynamics. Thus, the identification procedure provides both the nominal state-space models and the relative acoustic delay required for the delayed MPC implementation.

4.3. Validation of the Disturbance-Entry Direction Using Measured FIR Data

In the simulation study, the lumped state disturbance was assumed to enter through the primary-path input channel. Accordingly, the disturbance distribution matrix was selected as

$$B^d = \frac{B_{up}}{\|B_{up}\|_2}, \qquad \|B^d\|_2 = 1. \tag{4.5}$$

This choice is physically motivated by the feedforward ANC structure, where the primary disturbance at the error microphone is generated by propagation of the reference noise through the primary acoustic path. Therefore, model mismatch associated with the primary path can be represented as an additive equivalent state disturbance acting through the same state-space channel as the primary input.

Although this choice is physically motivated, its suitability was further evaluated using measured FIR data. The purpose of this validation was to determine whether the assumed disturbance-entry direction is able to reconstruct the mismatch between the measured FIR plant response and the nominal state-space prediction. Several physically meaningful state-disturbance candidates were considered:

$$B^d \in \left\{ \frac{B_{up}}{\|B_{up}\|_2}, \frac{B_{us}}{\|B_{us}\|_2}, \frac{B_{up}+B_{us}}{\|B_{up}+B_{us}\|_2}, \frac{L}{\|L\|_2} \right\}, \tag{4.6}$$

where B_{up} and B_{us} denote the primary- and secondary-path input directions, respectively, and L is the observer gain direction.

For each candidate B^d, the corresponding disturbance-to-output prediction matrix M_d was constructed over the MPC prediction horizon. The measured future output obtained from the FIR plant was denoted by Y_{FIR}, while the nominal state-space prediction was

denoted by Y_{nom}. The prediction mismatch was therefore defined as

$$\Delta Y = Y_{\text{FIR}} - Y_{\text{nom}}. \tag{4.7}$$

For each prediction window, an equivalent disturbance sequence was then obtained by solving the regularized least-squares problem

$$D_{\text{eq}} = \arg\min_{D} \|\Delta Y - M_d D\|_2^2 + \lambda_d \|D\|_2^2, \tag{4.8}$$

where $D = [d_k, d_{k+1}, \ldots, d_{k+N-1}]^\top$ is the disturbance sequence over the prediction horizon and λ_d is a small regularization parameter. The corrected prediction was then computed as

$$Y_{\text{corr}} = Y_{\text{nom}} + M_d D_{\text{eq}}. \tag{4.9}$$

The performance of each candidate disturbance direction was evaluated by comparing the nominal prediction error with the corrected prediction error. The nominal and corrected RMSE values were computed as

$$\text{RMSE}_{\text{nom}} = \sqrt{\frac{1}{N_{\text{eval}}} \sum_{i=1}^{N_{\text{eval}}} \|Y_{\text{FIR}} - Y_{\text{nom}}\|_2^2}, \tag{4.10}$$

and

$$\text{RMSE}_{\text{corr}} = \sqrt{\frac{1}{N_{\text{eval}}} \sum_{i=1}^{N_{\text{eval}}} \|Y_{\text{FIR}} - Y_{\text{corr}}\|_2^2}, \tag{4.11}$$

where N_{eval} denotes the total number of evaluated prediction windows. The fit improvement was then expressed in decibels as

$$\text{Fit}_{\text{dB}} = 20 \log_{10} \left(\frac{\text{RMSE}_{\text{nom}}}{\text{RMSE}_{\text{corr}}} \right). \tag{4.12}$$

Therefore, a larger positive value indicates that the equivalent disturbance correction significantly reduces the prediction error. In addition, the coefficient of determination was calculated as

$$R^2 = 1 - \frac{\sum_{i=1}^{N_{\text{eval}}} \|Y_{\text{FIR}} - Y_{\text{corr}}\|_2^2}{\sum_{i=1}^{N_{\text{eval}}} \|Y_{\text{FIR}} - Y_{\text{nom}}\|_2^2}. \tag{4.13}$$

A value of R^2 close to one indicates that the selected disturbance direction can explain most of the FIR/state-space mismatch.

Table 4.1 and Figure 4.4 compare the physically meaningful candidate disturbance directions using the mean fit improvement and mean R^2. The primary-path direction $B^d = B_{up}/\|B_{up}\|_2$ achieved the best performance among the evaluated candidates, with a mean fit improvement of 45.345 dB and a mean R^2 of 0.99997. This indicates that an equivalent disturbance entering through the primary-path input direction can reconstruct almost all of the measured FIR/state-space mismatch. The combined direction $B_{up} + B_{us}$ and the secondary-path direction B_{us} also reduced the mismatch, but their fit improvements were considerably lower. The observer-gain direction L produced almost no improvement, indicating that it is not suitable as the disturbance-entry direction for this system.

Validation of candidate disturbance-entry directions

Candidate	Fit (dB)	R^2	RMSE	D_{eq} RMS
$\boldsymbol{B_{up}}$	45.345	0.99997	1.1281×10^{-4}	1.1509
$\boldsymbol{B_{up} + B_{us}}$	12.820	0.94775	4.5738×10^{-3}	2.9615
$\boldsymbol{B_{us}}$	10.076	0.90171	6.2545×10^{-3}	5.0735
$\boldsymbol{L}$	1.3314×10^{-6}	3.0657×10^{-7}	0.019987	0.0078527

Table 4.1: Validation results for candidate disturbance-entry directions using measured FIR data.

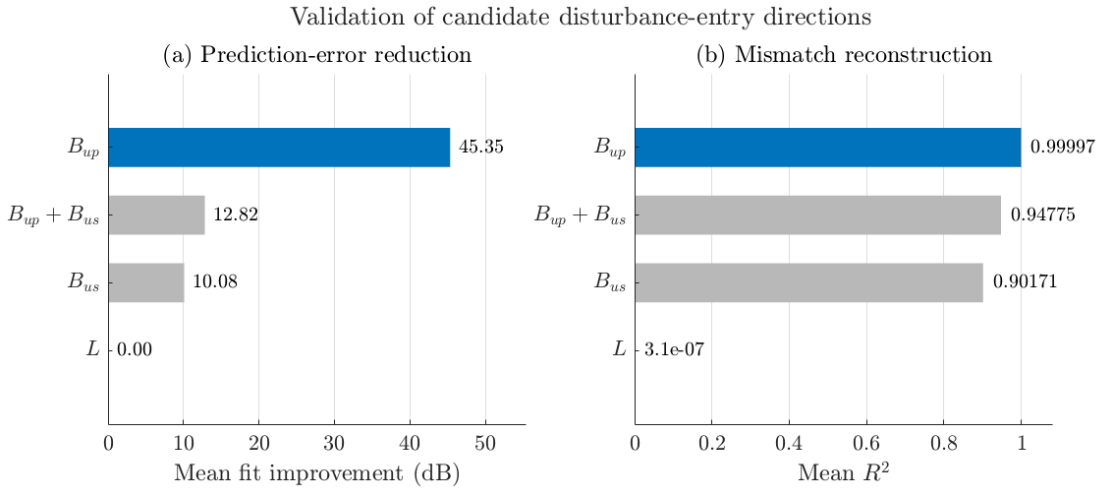


Figure 4.4: Validation of physically meaningful candidate disturbance-entry directions using measured FIR data: (a) mean fit improvement in dB and (b) mean mismatch reconstruction R^2. The primary-path direction $B^d = B_{up}/\|B_{up}\|_2$ provides the highest fit improvement and nearly perfect mismatch reconstruction.

These results validate the simulation assumption in Equation (4.5). Among the physically interpretable state-disturbance directions, the primary-path direction provides the strongest mismatch reconstruction and the lowest corrected prediction error. Therefore, $B^d = B_{up}/\|B_{up}\|_2$ was retained as the disturbance distribution matrix in the proposed MPC–LSTM framework.

4.4. Validation Results

4.4.1. Pure LSTM Controller Baseline

To evaluate whether a neural network alone can replace the model-based controller, a pure LSTM controller was implemented as an additional validation baseline. This controller is different from the proposed hybrid MPC-LSTM method. In the proposed method, the LSTM estimates a future mismatch sequence, and the linear MPC controller computes the secondary control input using the unconstrained quadratic solution followed by input saturation. In the pure LSTM baseline, there is no MPC, no observer, and no state-space prediction model. The LSTM directly generates the secondary-control signal from the primary/reference input.

The pure LSTM baseline is motivated by recent deep-learning-based ANC studies, where neural networks are used to learn nonlinear mappings between reference signals and cancelling control signals [38, 40]. Since ANC is a time-domain control problem with strong temporal dependence, LSTM networks are suitable candidates because they are designed to model sequential data and long-range dependencies [16]. In this thesis, however, the pure LSTM controller is used only as a comparison baseline. Its purpose is to test whether direct neural control is sufficient, or whether it is more effective to preserve the model-based MPC structure and use learning only for mismatch-preview compensation.

Let u_k^p denote the primary/reference input and u_k^s denote the secondary-control signal generated by the LSTM. The pure LSTM controller learns the sequence mapping

$$u_{1:T}^p \longrightarrow u_{1:T}^s,$$

where T is the training segment length. The implemented network is a unidirectional LSTM, and its recurrent computation proceeds chronologically. Consequently, the output at time k satisfies

$$u_k^s = f_\theta(u_1^p, \ldots, u_k^p)\,,$$

and does not depend on future reference samples. No bidirectional recurrent layers or

future-sample look-ahead are used. Although complete segments are processed during training, the hidden and cell states propagate only forward in time. During online inference, these states and the control output are updated one sample at a time.

A true optimal target signal for u_k^s is not available. Therefore, the LSTM is not trained in a supervised manner with respect to a labelled control signal. Instead, it is trained through the identified secondary-path FIR model. The generated control signal is passed through the secondary-path model, and the loss is computed from the residual error predicted at the error microphone. The same identified secondary-path FIR model is used during training and validation of this baseline. The reported results therefore represent matched secondary-path conditions and do not evaluate robustness to secondary-path modelling error.

The identified secondary-path impulse response is denoted by $s(n)$. The anti-noise contribution generated by the LSTM control signal is therefore

$$y_k^s = (s * u^s)_k,$$

where $*$ denotes convolution. The residual error used for training is

$$e_k = y_k^p - y_k^s,$$

where y_k^p is the primary target signal at the error microphone. The LSTM parameters are updated by minimizing a differentiable acoustic loss function,

$$J = \frac{1}{|\mathcal{I}|}\sum_{k\in\mathcal{I}} e_k^2 + \lambda_u \frac{1}{|\mathcal{I}|}\sum_{k\in\mathcal{I}} (u_k^s)^2 + \lambda_{\Delta u}\frac{1}{|\mathcal{I}|}\sum_{k\in\mathcal{I}} \left(u_k^s - u_{k-1}^s\right)^2,$$

where $\mathcal{I}$ is the set of valid samples after removing the initial secondary-path transient. The first term minimizes the residual microphone error, the second term penalizes excessive control effort, and the third term penalizes rapid variations in the control signal. This training strategy is analogous to a differentiable acoustic version of the filtered-reference idea: the network output is not compared with a desired control signal, but is evaluated through the secondary acoustic path.

To keep the neural control signal within the admissible amplitude range, the raw LSTM output z_k is passed through a bounded activation,

$$u_k^s = U_{s,\max}\tanh(z_k).$$

Since $\tanh(\cdot) \in [-1, 1]$, the control signal satisfies

$$|u_k^s| \leq U_{s,\max}.$$

In the validation study, $U_{s,\max} = 1$. This bounded activation imposes the same amplitude limit used by the post-optimization saturation of the MPC control input, although the pure LSTM controller does not include predictive optimization or explicit control-effort weighting.

A sequence-to-sequence LSTM formulation was used for the reported pure LSTM baseline. This choice is appropriate because the ANC objective is inherently sequence-based. The secondary-path convolution couples neighbouring control samples, so the residual error at one time instant depends not only on the current control value but also on previous control values. Training over complete time segments therefore matches the physical acoustic objective more directly than treating each control sample as an independent regression target. A sliding-window, window-to-one formulation was also considered as an alternative causal implementation, but the sequence-to-sequence formulation was selected for the reported baseline because it gives a direct mapping from the reference sequence to the full secondary-control sequence and avoids reconstructing the control signal from many overlapping windows.

The pure LSTM baseline used the same dataset partitioning and normalization procedure as the sequence-model benchmark in Section 3.2. The dataset contained 100 independently generated randomized band-limited white-noise excitation records, each with a duration of 15 s, corresponding to 60000 samples per record at 4 kHz. Complete records were divided at the sequence level into 70 training, 20 validation, and 10 test records. Normalization parameters were computed using only the training records and were then applied unchanged to the validation and test records. The remaining architecture and optimization settings are summarized in Table 4.2.

Item	Setting
Training data	100 randomized band-limited white-noise records, each 15 s long at 4 kHz
Data split	70/20/10 complete records for training, validation, and testing, respectively
Normalization	Benchmark normalization computed from the training records only
Segment length	$T = 4096$ samples (1.024 s)
Architecture	Unidirectional LSTM(64)–dropout(0.10)–unidirectional LSTM(32)–FC(32)–ReLU–FC(1)
Output constraint	$u_k^s = U_{s,\max} \tanh(z_k)$, with $U_{s,\max} = 1$
Loss weights	$\lambda_u = 10^{-5}$, $\lambda_{\Delta u} = 10^{-5}$
Optimizer	Adam with initial learning rate 10^{-3}
Learning-rate schedule	Multiplication by 0.5 every 2000 iterations
Training length	10000 iterations; no early stopping
Batch construction	One randomly selected 4096-sample training segment per iteration
Secondary-path model	Same identified FIR model used for training and validation

Table 4.2: Training configuration of the pure LSTM controller baseline.

The pure LSTM controller was first compared directly with MPC without mismatch preview under the white-noise disturbance. Figure 4.5 shows the residual signals in the time domain and their corresponding power spectral densities. The pure LSTM controller reduces the uncontrolled response, confirming that a causal neural controller can learn a useful cancelling action from the reference signal. However, its residual remains visibly larger than that obtained with MPC, particularly across the identified 100–800 Hz control band.

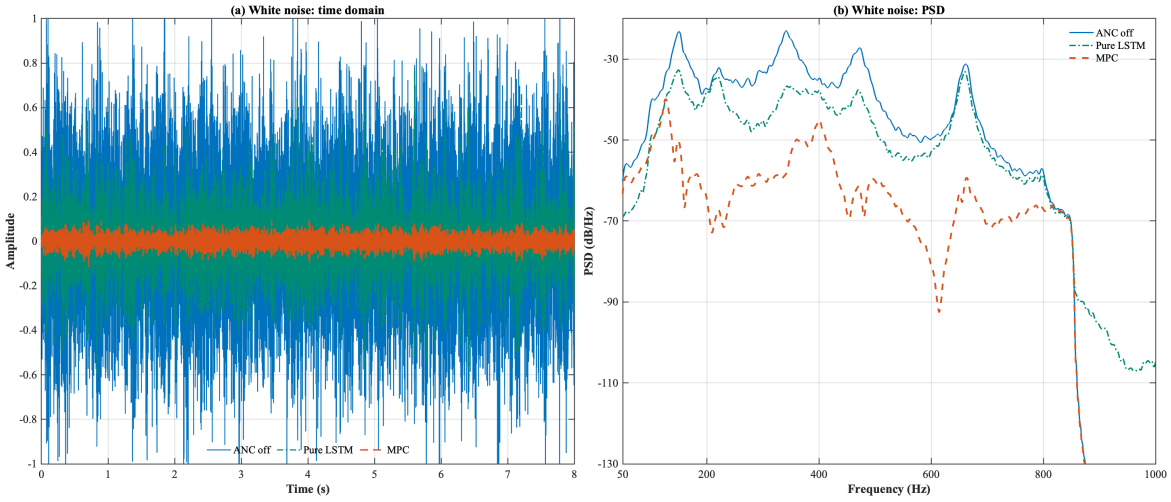


Figure 4.5: White-noise comparison of ANC off, the pure LSTM controller, and MPC without mismatch preview: (a) time-domain residual signals and (b) power spectral densities.

For the white-noise realization in Figure 4.5, the pure LSTM controller achieves a band-limited noise reduction of 8.0834 dB, whereas MPC without mismatch preview achieves 21.1750 dB, corresponding to an additional reduction of 13.0916 dB. This result is consistent with the different roles of the two controllers. The pure LSTM learns a direct reference-to-control mapping from the training distribution, but it does not explicitly propagate the identified plant state or recompute a finite-horizon residual-error optimum during operation. MPC instead uses the identified state-space dynamics and updates the receding-horizon control action from the current state estimate, which provides a structured use of the acoustic-path model [24, 36]. Although deep neural controllers can be effective for nonlinear ANC [38], this comparison shows that, for the plant, data, and training configuration considered here, replacing the model-based controller entirely with the evaluated LSTM degrades attenuation. It therefore supports retaining the state-space and MPC framework and using the LSTM as an auxiliary mismatch-preview estimator rather than as the complete controller.

4.5. Comparative Noise-Reduction Performance

This section compares the proposed Hybrid MPC-LSTM controller with the ANC-off reference and three controller baselines: FxNLMS, the pure LSTM controller, and MPC without mismatch preview. FxNLMS is evaluated using two step sizes, $\mu = 0.001$ and $\mu = 0.01$, to examine the effect of adaptation speed. The pure LSTM baseline evaluates

direct neural control, whereas MPC without mismatch preview isolates the performance of the nominal model-based controller. The proposed Hybrid MPC-LSTM controller retains the MPC structure while incorporating LSTM-based mismatch-preview compensation.

Before comparing the closed-loop attenuation results, Figure 4.6 shows the equivalent model-mismatch signal obtained during offline data generation for LSTM training. Over the identified 100–800 Hz band, the measured FIR model was used as a high-order experimentally identified reference model, while the reduced-order state-space realization was retained as the nominal MPC model [26]. Thus, the output discrepancy was formed as $\Delta y_k = y_{\mathrm{FIR},k} - y_{\mathrm{nom},k}$.

The mismatch signal was reconstructed using the proxy formulation developed in Section 2.5.1. Specifically, the state-error proxy was first obtained from Equation (2.61), and the scalar mismatch proxy was then computed from the one-step relation in Equations (2.62) and (2.63). The quantity plotted as $d_{\mathrm{eq}}(k)$ in Figure 4.6 is this reconstructed realization of $d_{\mathrm{proxy},k}$ obtained from the measurement-derived models. Its varying amplitude illustrates that the FIR/state-space discrepancy is dynamic rather than a constant bias. The LSTM is trained to predict the corresponding future proxy sequence, which is incorporated into the Hybrid MPC-LSTM prediction through $M^d \hat{D}_{m,k}$; its first element is also used for observer compensation.

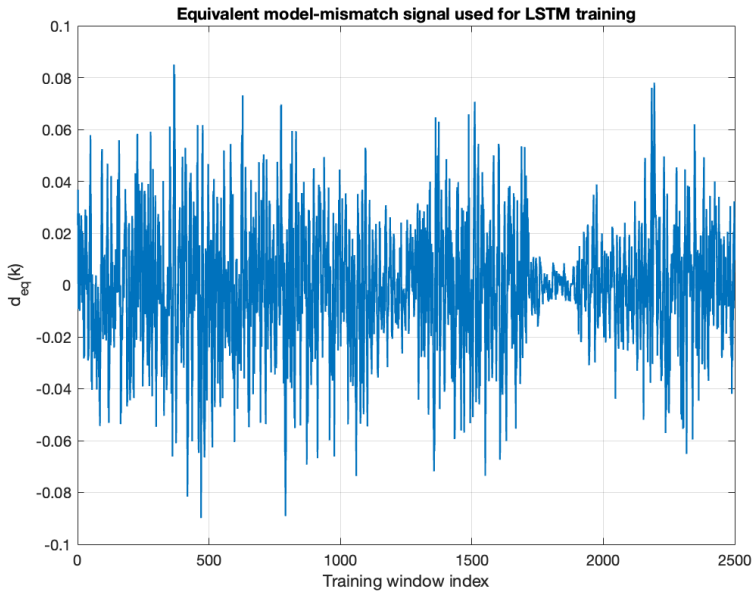


Figure 4.6: Equivalent model-mismatch proxy reconstructed using Equation (2.63) from the output discrepancy between the measurement-derived FIR reference model and the nominal state-space MPC model.

4.5.1. Time-Domain Residual Response

Figure 4.7 compares the residual error signals over the same 8 s interval. The plotted FxNLMS response uses the default stable step size $\mu = 0.001$; the effect of increasing the step size is reported separately in Table 4.3.

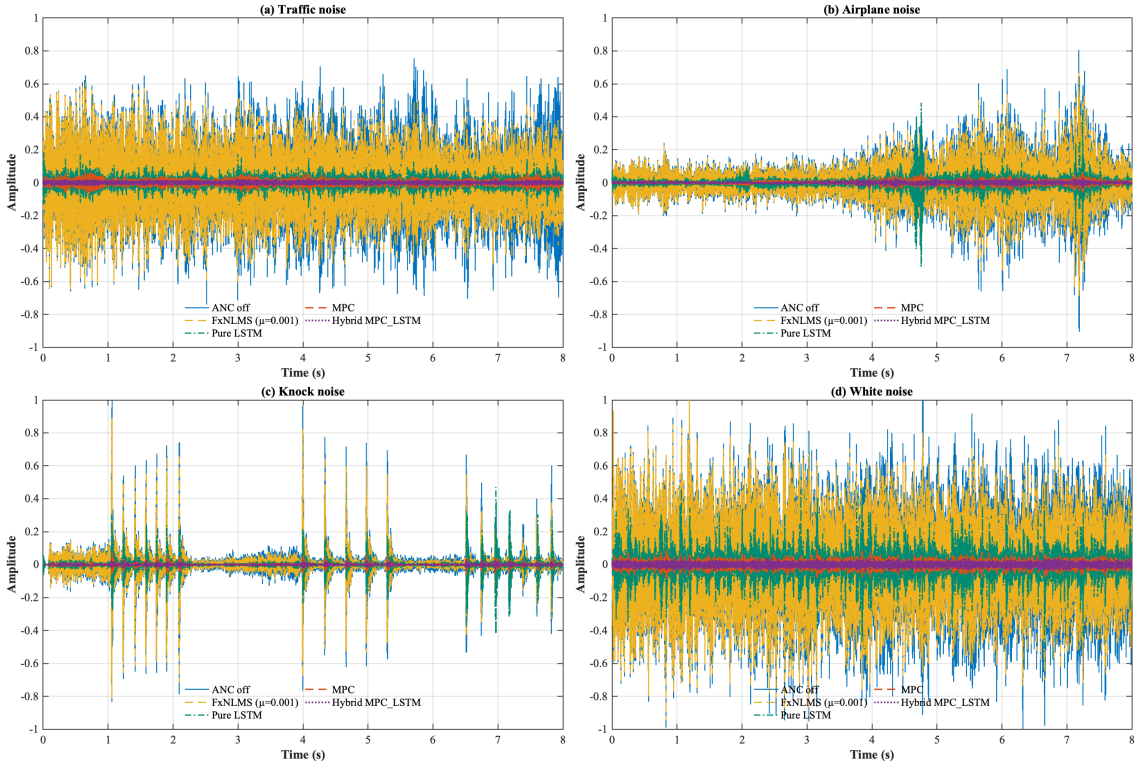


Figure 4.7: Time-domain residual error signals

For traffic noise in Figure 4.7(a), the ANC-off signal has a persistent broadband envelope. The FxNLMS curve remains close to this envelope throughout the record, showing that the small adaptation step does not converge rapidly enough to compensate the measured-path dynamics and their mismatch. The pure LSTM compresses the residual substantially, but a visibly wider band remains around zero than for either MPC method. MPC without mismatch preview produces a narrow residual immediately, while Hybrid MPC-LSTM gives the smallest sustained envelope. The improvement of the hybrid method is consistent with the learned preview correcting the portion of the response not represented by the nominal state-space model.

Airplane noise in Figure 4.7(b) is non-stationary: its amplitude is relatively low before $t \approx 4$ s and increases markedly thereafter, with strong events near $t \approx 4.7$ s, 5.5–6.3 s, and 7.1–7.4 s. FxNLMS follows the growth of the ANC-off response and therefore leaves

a large residual. The pure LSTM generally attenuates the signal, but the sharp excursion around 4.7 s indicates sensitivity of the direct learned control law to a rapid change in the disturbance envelope. Both MPC methods remain concentrated near zero through these changes. Hybrid MPC-LSTM has the narrowest residual because the nominal finite-horizon prediction is supplemented by the estimated mismatch sequence.

The knock input in Figure 4.7(c) contains separate impulsive groups around 1.0–2.1 s, 4.0–5.3 s, and 6.5–8.0 s. The ANC-off and FxNLMS curves retain nearly the full positive and negative peaks. Pure LSTM reduces the background component but leaves clear transient residuals, particularly in the final group. MPC suppresses most of the impulse amplitude, and Hybrid MPC-LSTM further reduces the residual between and during the impulses. Some narrow peaks remain because an abrupt broadband event cannot be cancelled perfectly by a causal controller with finite acoustic bandwidth and bounded actuator amplitude.

For white noise in Figure 4.7(d), the ANC-off signal has an approximately stationary broadband envelope. FxNLMS again remains close to the uncontrolled case, whereas pure LSTM provides an intermediate reduction. MPC confines the residual to a narrow band around zero, and the hybrid controller reduces that band further. Unlike the airplane and knock cases, no isolated degradation interval is visible because the statistical characteristics of this disturbance remain comparatively stationary.

4.5.2. Time–Frequency Residual Distribution

The short-time Fourier transform (STFT) results in Figure 4.8 show where the residual energy occurs in both time and frequency. All panels use the same -40 to 20 dB color scale; consequently, changes from red and yellow to cyan and blue represent genuine reductions and not independent color normalization. The discussion focuses on the identified 100–800 Hz control band.

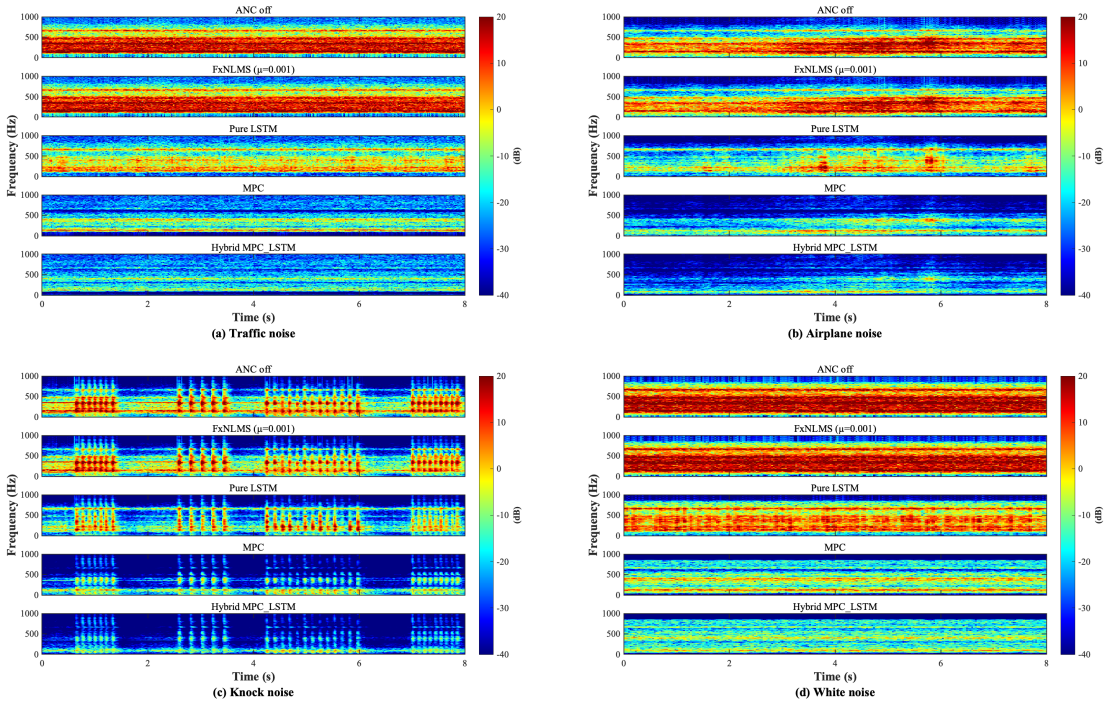


Figure 4.8: STFT spectra of the residual error signals

Traffic noise in Figure 4.8(a) has continuous energy concentrated mainly below 600 Hz. The near-identical ANC-off and FxNLMS maps confirm the weak attenuation inferred from the time-domain plot. Pure LSTM lowers the broadband level, although yellow and green components remain across much of the record. MPC and Hybrid MPC-LSTM shift most of this region toward blue and cyan, demonstrating sustained rather than transient-only suppression.

For airplane noise in Figure 4.8(b), the uncontrolled energy increases after approximately 4 s, with pronounced low- and mid-frequency activity near 4, 5.8, and 7 s. FxNLMS retains this pattern. Pure LSTM reduces the background level but leaves localized high-energy regions at the same instants, consistent with its time-domain excursions. MPC removes most of these regions, and Hybrid MPC-LSTM maintains a low residual over the varying operating intervals. This indicates that mismatch preview is useful when both the amplitude and spectral distribution change during a record.

The vertical ridges in Figure 4.8(c) identify the individual knock events and show that each impulse excites a wide frequency range. FxNLMS preserves almost the same ridge intensity as ANC off. Pure LSTM reduces their intensity but does not remove their broadband structure. MPC and Hybrid MPC-LSTM strongly attenuate the continuous background and the lower parts of the ridges; the event times remain visible because the

impulses are causal, short, and broadband. Thus, the remaining ridges indicate finite transient rejection rather than sustained instability.

White noise in Figure 4.8(d) produces a nearly uniform time–frequency field over the controlled band. The persistence of red and yellow in the FxNLMS panel shows that its small-step adaptation provides little broadband rejection over 8 s. Pure LSTM gives a clear but incomplete reduction. Both predictive controllers lower the residual across the complete record and frequency band, with no isolated time interval of increased energy. This agrees with the stationary residual envelopes observed in Figure 4.7(d).

4.5.3. Time-Varying Attenuation

Figure 4.9 reports the attenuation calculated over successive one-second intervals using Equation (3.7). Because that equation uses the controlled-to-ANC-off power ratio, more negative ANR values indicate stronger attenuation.

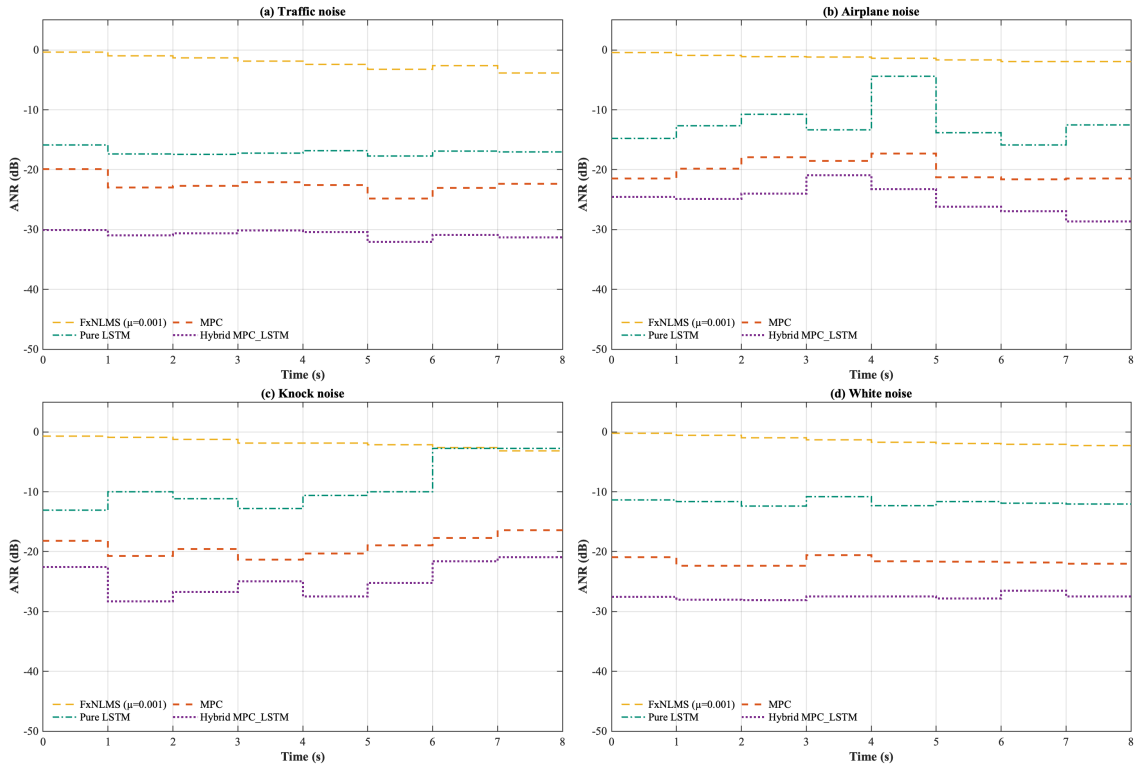


Figure 4.9: Time-varying attenuation calculated over one-second intervals

For traffic noise in Figure 4.9(a), FxNLMS improves gradually from approximately 0 to only −4 dB, which is the expected signature of slow adaptation with $\mu = 0.001$. Pure LSTM remains near −16 to −18 dB, MPC varies between approximately −20 and

−25 dB, and Hybrid MPC-LSTM remains near −30 to −32 dB. The nearly parallel MPC and hybrid trends show that both respond to changes in disturbance energy, while the roughly 8–11 dB separation quantifies the benefit of mismatch preview.

The airplane result in Figure 4.9(b) varies more strongly because the disturbance is non-stationary. Pure LSTM ranges from about −16 dB to only −4 dB in the 4–5 s interval, when the disturbance changes rapidly. MPC remains between approximately −18 and −22 dB. Hybrid MPC-LSTM also loses some attenuation during the transition, rising to about −21 dB, but remains below MPC and subsequently reaches approximately −29 dB. The simultaneous temporary rise of all learned/model-based curves reflects a harder input interval; the rapid recovery of the hybrid curve shows that this degradation is temporary rather than persistent.

For knock noise in Figure 4.9(c), the windowed values change whenever the density and magnitude of the impulses change. Pure LSTM provides approximately 10–13 dB attenuation during the first 6 s, but falls to roughly 3 dB during the final impulse group. MPC remains near −18 to −22 dB over most windows. Hybrid MPC-LSTM reaches approximately −25 to −29 dB during the central intervals and remains the strongest method after 6 s, despite a reduction to approximately −21 dB. These changes are consistent with the concentration of short impulse energy inside individual one-second evaluation windows.

White noise in Figure 4.9(d) gives the most stable curves. Pure LSTM remains close to −11 to −12 dB, MPC remains near −21 to −22 dB, and Hybrid MPC-LSTM remains near −27 to −28 dB. The limited variation confirms that the performance ordering is not caused by one isolated event and is maintained under continuous broadband excitation.

4.5.4. Power-Spectral-Density Analysis

Figure 4.10 compares the power spectral densities (PSDs) over 50–1000 Hz. The purpose of this view is to determine whether a low time-domain residual results from broadband attenuation or only from cancellation at a small number of frequencies.

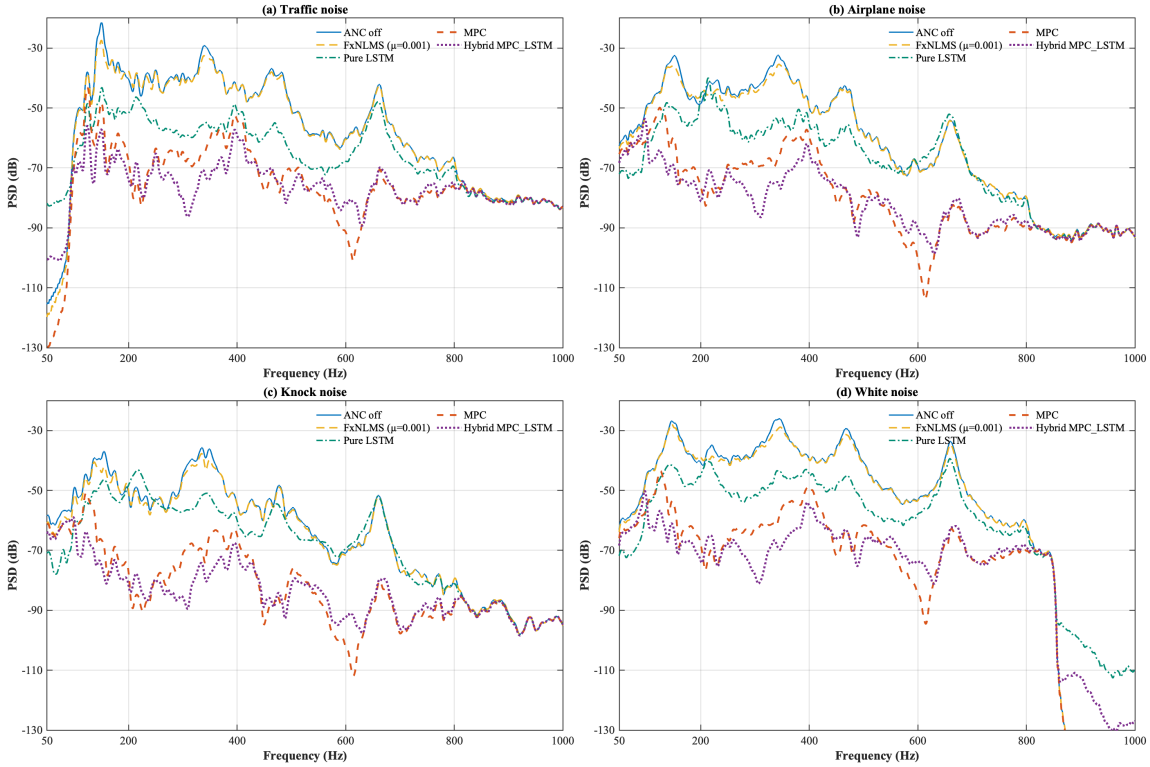


Figure 4.10: Power spectral densities of the residual error signals

For traffic noise in Figure 4.10(a), the dominant uncontrolled components occur around 150 Hz, 300–400 Hz, 470 Hz, and 650–700 Hz. FxNLMS nearly overlaps ANC off at these components. Pure LSTM reduces most of the controlled band, but several peaks remain between −45 and −55 dB. MPC and Hybrid MPC-LSTM provide substantially larger broadband reductions. Their curves cross at some narrow frequencies, which is expected because model-based cancellation redistributes residual spectral energy; the integrated 100–800 Hz result, rather than one spectral notch, determines the overall ranking.

The airplane spectrum in Figure 4.10(b) contains strong components near 150, 350, 460, and 650 Hz. Pure LSTM reduces the broad level but leaves significant residual components, including local peaks around 200–250 Hz and 650 Hz. MPC and Hybrid MPC-LSTM reduce the principal peaks across most of the controlled band, with particularly deep attenuation around 500–620 Hz. The result supports the interpretation that the state-space predictor captures the main acoustic resonances, while the learned mismatch preview improves the aggregate response when the measured plant differs from that predictor.

For knock noise in Figure 4.10(c), the broad peaks near 150, 350, 500, and 650 Hz arise from the repeated transients. FxNLMS again remains close to ANC off, and pure LSTM

gives an intermediate reduction. The two MPC curves suppress the principal components by several tens of decibels over much of the band. Narrow residual maxima remain because impulsive signals distribute energy broadly and cannot be eliminated by targeting only one resonance.

The white-noise spectrum in Figure 4.10(d) provides the clearest broadband comparison. Pure LSTM lowers the PSD relative to ANC off, but the dominant peaks remain visible. MPC and Hybrid MPC-LSTM reduce the complete 100–800 Hz region, rather than only isolated frequencies. Above approximately 850 Hz, the curves approach the measurement or numerical floor and their ordering is no longer a meaningful control comparison, because the models and performance objective were defined for the 100–800 Hz band.

4.5.5. Quantitative Band-Limited Comparison

Table 4.3 summarizes the integrated 100–800 Hz reductions corresponding to the four-domain evaluation above. These values refer to the four-disturbance evaluation used in Figures 4.7–4.10.

Algorithm	Traffic	Airplane	Knock	White
FxNLMS ($\mu = 0.001$)	2.9103	1.8101	2.1581	1.4441
FxNLMS ($\mu = 0.01$)	8.3337	6.9794	7.6683	6.0752
Pure LSTM	17.1260	11.2760	7.5834	11.9472
MPC without mismatch preview	22.6090	21.0210	19.9890	21.3413
Hybrid MPC-LSTM	31.0170	30.3000	29.8000	29.0880

Table 4.3: Band-limited noise reduction in the 100–800 Hz frequency range for different disturbance types. Values are reported in dB relative to the ANC-off case.

FxNLMS with $\mu = 0.001$ achieves only 1.4441–2.9103 dB reduction. Increasing the step size to $\mu = 0.01$ improves this range to 6.0752–8.3337 dB, confirming that the weak plotted FxNLMS result is partly associated with slow adaptation; nevertheless, the larger-step result remains below both predictive controllers. Pure LSTM achieves 7.5834–17.1260 dB, demonstrating useful direct neural cancellation but also a strong dependence on disturbance type. MPC without mismatch preview is more consistent, providing 19.9890–22.6090 dB.

Hybrid MPC-LSTM gives the largest reduction relative to MPC without mismatch pre-

view, the mean improvement is approximately 8.81 dB. Because the two MPC cases use the same nominal model, observer, cost, saturation, and prediction horizon, this difference isolates the practical contribution of the learned mismatch preview. Taken together, the time-domain, STFT, one-second attenuation, PSD, and band-limited results show that the proposed method improves both average broadband attenuation and robustness to non-stationary and impulsive disturbances without discarding the structured state-space MPC controller.

5 | Conclusions and future developments

5.1. Conclusions

This thesis addressed active noise control under model mismatch. Classical adaptive algorithms can require a significant convergence interval, while model predictive control depends on the accuracy of the nominal acoustic-path model. In practice, identification errors, unmodelled dynamics, nonlinear behavior, actuator limitations, and changes in the acoustic environment can produce a discrepancy between the real plant and the state-space model used by the controller. If this discrepancy is ignored, both state estimation and finite-horizon prediction can deteriorate.

To address this problem, a Hybrid MPC-LSTM controller was developed. The control law retains a linear state-space MPC structure, while a causal LSTM predicts the future lumped model-mismatch sequence. The complete predicted sequence is included in the lifted MPC model, and its first element is used to compensate the steady-state observer. In this way, the neural network does not replace the model-based controller. Instead, it estimates the part of the system response that is not represented by the nominal model.

The mismatch predictor was trained using causal input windows and an offline reconstructed mismatch proxy. Six sequence-model architectures were compared under a common benchmark, and the LSTM provided the best trade-off between average prediction error, worst-case error, and computational burden. During closed-loop operation, this predictor was used with fixed offline-trained parameters.

The observer analysis established the boundedness properties required by the proposed compensation scheme. The bounded sigmoid and hyperbolic-tangent activations imply a bounded LSTM hidden state. For finite fixed readout parameters, the resulting mismatch estimate is also bounded. Under detectability of (A, C) and a gain L for which $A - LC$ is Schur stable, the compensated observer-error dynamics are input-to-state stable with respect to the mismatch-prediction error. Consequently, a bounded prediction error

produces a uniformly ultimately bounded observer error, and the observer error converges to zero if the mismatch-prediction error converges to zero.

The numerical and measured-model evaluations showed a consistent performance ordering across traffic, airplane, knock, and white-noise disturbances. The proposed Hybrid MPC-LSTM controller achieved greater noise reduction than FxNLMS, the pure LSTM controller, and MPC without mismatch preview. FxNLMS was limited by its convergence behavior, while the pure LSTM baseline showed that direct neural control did not exploit the identified acoustic dynamics as effectively as MPC. MPC without mismatch preview provided strong and consistent attenuation, confirming the value of the state-space predictive structure. The additional improvement obtained by Hybrid MPC-LSTM therefore supports the central premise of this thesis: learning is effective here when it complements the model-based controller by estimating plant-model mismatch.

5.2. Future Developments

The first priority is an optimized hard real-time implementation of the complete Hybrid MPC-LSTM control loop. The current MATLAB inference time exceeds the available one-sample deadline at 4 kHz. Compiled neural-network inference, code generation, deterministic memory allocation, parallel computation, and hardware acceleration should therefore be evaluated on the target platform.

A second direction is delay-aware control design. Since the LSTM prediction and MPC computation introduce a non-negligible computational delay, future work should explicitly include this delay in the prediction model or compensate it in the timing of the generated control signal. This is important because ANC performance depends strongly on the synchronization between the primary noise and the anti-noise signal.

A third direction is the development of lighter neural mismatch predictors. Smaller LSTM networks, or compact causal neural architectures could be investigated to reduce computational cost while maintaining accurate mismatch-preview prediction. This would help make the proposed method more suitable for embedded real-time implementation.

Future work should also investigate systematic offline retraining and redeployment strategies. If the acoustic environment changes enough that the fixed predictor becomes inaccurate, newly collected data could be used to retrain or fine-tune the mismatch predictor before redeployment. Such a procedure would preserve the fixed-predictor control structure while improving robustness to long-term changes in operating conditions.

Finally, the proposed framework could be extended to multichannel ANC. Multiple ref-

erence sensors, secondary sources, and error microphones would provide a more realistic representation of practical noise-control systems, but would also introduce acoustic coupling, larger state and prediction matrices, and greater computational demand. Such an extension should therefore be considered together with model reduction and real-time implementation constraints.

Bibliography

[1] L. Bai, M. Li, S. Lian, K. Chen, and J. Lu. WaveNet-Volterra neural networks for active noise control: A fully causal approach, 2025.

[2] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. URL https://arxiv.org/abs/1803.01271.

[3] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012. URL https://www.jmlr.org/papers/v13/bergstra12a.html.

[4] X. Bouthillier, P. Delaunay, M. Bronzi, A. Trofimov, B. Nichyporuk, J. Szeto, N. M. Sepahvand, E. Raff, K. Madan, V. Voleti, S. Ebrahimi Kahou, V. Michalski, T. Arbel, C. Pal, G. Varoquaux, and P. Vincent. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems*, volume 3, 2021. URL https://proceedings.mlsys.org/paper_files/paper/2021/hash/0184b0cd3cfb185989f858a1d9f5c1eb-Abstract.html.

[5] J. C. Burgess. Active adaptive sound control in a duct: A computer simulation. *The Journal of the Acoustical Society of America*, 70(3):715–726, 1981. doi: 10.1121/1.386908.

[6] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, Doha, Qatar, 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179.

[7] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. URL https://arxiv.org/abs/1412.3555.

[8] D. P. Das and G. Panda. Active mitigation of nonlinear noise processes using a novel

filtered-s LMS algorithm. *IEEE Transactions on Speech and Audio Processing*, 12 (3):313–322, 2004. doi: 10.1109/TSA.2003.822741.

[9] J. Dodge, S. Gururangan, D. Card, R. Schwartz, and N. A. Smith. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2185–2194, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1224. URL `https://aclanthology.org/D19-1224`.

[10] S. J. Elliott. *Signal Processing for Active Control*. Academic Press, 2001. ISBN 978-0-12-237085-4. doi: 10.1016/B978-0-12-237085-4.X5000-5.

[11] S. J. Elliott and P. A. Nelson. Active noise control. *IEEE Signal Processing Magazine*, 10(4):12–35, 1993. doi: 10.1109/79.248551.

[12] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. URL `https://www.deeplearningbook.org`.

[13] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, 2017. doi: 10.1109/TNNLS.2016.2582924.

[14] C. H. Hansen and S. D. Snyder. *Active Control of Noise and Vibration*. E & FN Spon, London, UK, 1997. ISBN 978-0-419-19390-6.

[15] S. O. Haykin. *Adaptive Filter Theory*. Pearson, 5 edition, 2014. ISBN 978-0-273-77572-0.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[17] K. Iwai, S. Hase, and Y. Kajikawa. Multichannel feedforward active noise control system with optimal reference microphone selector based on time difference of arrival. *Applied Sciences*, 8(11):2291, 2018. doi: 10.3390/app8112291.

[18] M. Jalaeian-F., M. M. Fateh, and M. Rahimiyan. Optimal predictive impedance control in the presence of uncertainty for a lower limb rehabilitation robot. *Journal of Systems Science and Complexity*, 33(5):1310–1329, 2020. doi: 10.1007/s11424-020-8335-5. URL `https://doi.org/10.1007/s11424-020-8335-5`.

[19] Z.-P. Jiang and Y. Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37(6):857–869, 2001. doi: 10.1016/S0005-1098(01)00028-0.

[20] R. Khalili Amirabadi, M. Jalaeian-Farimani, and O. S. Fard. Lstm-empowered reinforcement learning in bi-level optimal control for nonlinear systems with uncertain dynamics. *ISA Transactions*, 168:465–478, 2026. ISSN 0019-0578. doi: https://doi.org/10.1016/j.isatra.2025.11.027. URL `https://www.sciencedirect.com/science/article/pii/S0019057825006457`.

[21] S. M. Kuo and D. R. Morgan. *Active Noise Control Systems: Algorithms and DSP Implementations*. Wiley Series in Telecommunications and Signal Processing. John Wiley & Sons, New York, NY, USA, 1996. ISBN 978-0-471-13424-4.

[22] S. M. Kuo and D. R. Morgan. Active noise control: A tutorial review. *Proceedings of the IEEE*, 87(6):943–973, 1999. doi: 10.1109/5.763310.

[23] S. Kwon, B.-S. Kim, and J. Park. Active noise reduction with filtered least-mean-square algorithm improved by long short-term memory models for radiation noise of diesel engine. *Applied Sciences*, 12(20):10248, 2022. doi: 10.3390/app122010248.

[24] C. Liang, F. Ripamonti, H. R. Karimi, and M. Pawełczyk. Real-time implementation of delayed model predictive control in active noise control systems. *Journal of Sound and Vibration*, page 119800, 2026. ISSN 0022-460X. doi: https://doi.org/10.1016/j.jsv.2026.119800. URL `https://www.sciencedirect.com/science/article/pii/S0022460X2600163X`.

[25] Z. C. Lipton, J. Berkowitz, and C. Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015. URL `https://arxiv.org/abs/1506.00019`.

[26] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, 2 edition, 1999.

[27] L. Lu, K. L. Yin, R. C. de Lamare, Z. Zheng, Y. Yu, X. Yang, and B. Chen. A survey on active noise control in the past decade–part i: Linear systems. *Signal Processing*, 183:108039, 2021. doi: 10.1016/j.sigpro.2021.108039.

[28] L. Lu, K. L. Yin, R. C. de Lamare, Z. Zheng, Y. Yu, X. Yang, and B. Chen. A survey on active noise control in the past decade–part ii: Nonlinear systems. *Signal Processing*, 181:107929, 2021. doi: 10.1016/j.sigpro.2020.107929.

[29] D. Q. Mayne, M. M. Seron, and S. V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[30] R. Napoli and L. Piroddi. Nonlinear active noise control with NARX models. *IEEE*

Transactions on Audio, Speech, and Language Processing, 18(2):286–295, 2010. doi: 10.1109/TASL.2009.2025798.

[31] P. A. Nelson and S. J. Elliott. *Active Control of Sound*. Academic Press, London, UK, 1992. ISBN 978-0-12-515425-3.

[32] S. Park, E. Patterson, and C. Baum. Long short-term memory and convolutional neural networks for active noise control. In *2019 5th International Conference on Frontiers of Signal Processing*, pages 121–125, 2019. doi: 10.1109/ICFSP48124.2019.8938042.

[33] D. Shi, B. Lam, W.-S. Gan, J. Cheer, and S. J. Elliott. Active noise control in the new century: The role and prospect of signal processing, 2023.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL `https://arxiv.org/abs/1706.03762`.

[35] B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1985. ISBN 978-0-13-004029-9.

[36] A. G. Wills, D. Bates, A. J. Fleming, B. Ninness, and S. O. R. Moheimani. Model predictive control applied to constraint handling in active noise and vibration control. *IEEE Transactions on Control Systems Technology*, 16(1):3–12, 2008. doi: 10.1109/TCST.2007.903062.

[37] F. Yang, Y. Cao, M. Wu, F. Albu, and J. Yang. Frequency-domain filtered-x LMS algorithms for active noise control: A review and new insights. *Applied Sciences*, 8 (11):2313, 2018. doi: 10.3390/app8112313.

[38] H. Zhang and D. Wang. Deep anc: A deep learning approach to active noise control. *Neural Networks*, 141:1–10, 2021. doi: 10.1016/j.neunet.2021.03.037.

[39] H. Zhang, A. Pandey, and D. Wang. Attentive recurrent network for low-latency active noise control. In *Interspeech 2022*, pages 956–960, 2022. doi: 10.21437/Interspeech.2022-811.

[40] H. Zhang, A. Pandey, and D. Wang. Low-latency active noise control using attentive recurrent network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1114–1123, 2023. doi: 10.1109/TASLP.2023.3244528.

[41] K. Zhang, G. Lyu, and X. Luo. A deep recurrent neural network controller for nonlinear active noise control systems. In *2020 IEEE 6th International Conference on*

Computer and Communications, pages 2393–2396, 2020. doi: 10.1109/ICCC51575.2020.9345164.

A | Supplementary Derivation of the Delayed Prediction Model

This appendix provides the detailed lifted equations associated with the delayed formulation introduced in Section 2.2. The purpose is to make explicit the delayed subsystem realizations and the prediction matrices used in Equation (2.30), without repeating the controller derivation already given in the main text.

A.1. Delayed Path Realizations

Let the measured primary and secondary acoustic paths be decomposed into a delay-free part and a pure discrete-time delay as in Equation (2.21). The corresponding delay-free state-space representations can be written with delayed input indices as

$$\begin{cases} x^p_{k+1-N^p_d} = A^p_{dl} x^p_{k-N^p_d} + B^p_{dl} u^p_{k-N^p_d}, \\ y^p_k = C^p_{dl} x^p_{k-N^p_d} + D^p_{dl} u^p_{k-N^p_d}, \end{cases} \tag{A.1}$$

and

$$\begin{cases} x^s_{k+1-N^s_d} = A^s_{dl} x^s_{k-N^s_d} + B^s_{dl} u^s_{k-N^s_d}, \\ y^s_k = C^s_{dl} x^s_{k-N^s_d} + D^s_{dl} u^s_{k-N^s_d}. \end{cases} \tag{A.2}$$

Here, the subscript dl denotes the delay-free realization of each measured acoustic path. The feedthrough terms D^p_{dl} and D^s_{dl} are retained in the lifted delayed prediction matrices, consistently with the nominal prediction matrices in Equations (2.9) and (2.10).

A.2. Lifted Delayed Prediction Matrices

Using the delayed augmented state x'_k defined in Equation (2.22), the compact delayed model can be propagated over the horizon. For $i = 1, \ldots, N$, the predicted residual output

is

$$\begin{aligned} y_{k+i} = C_{dl}A_{dl}^{i}x'_k + \sum_{j=0}^{i-1} C_{dl}A_{dl}^{i-1-j}B_{dl}^{p,a}u^{p}_{k+j-N_d^p} \\ + \sum_{j=0}^{i-1} C_{dl}A_{dl}^{i-1-j}B_{dl}^{s,a}u^{s}_{k+j-N_d^s} + D_{dl}^{p}u^{p}_{k+i-N_d^p} - D_{dl}^{s}u^{s}_{k+i-N_d^s}. \end{aligned} \tag{A.3}$$

Because $C_{dl} = [C_{dl}^{p} \quad -C_{dl}^{s}]$, the secondary contribution appears with a negative sign. Therefore, the residual sequence can be written in the form used in the main chapter,

$$Y = O_{dl}x'_k + M_{dl}^{p}U'^{p} - M_{dl}^{s}U'^{s}. \tag{A.4}$$

The delayed observability matrix is

$$O_{dl} = \begin{bmatrix} C_{dl}A_{dl} \\ C_{dl}A_{dl}^{2} \\ \vdots \\ C_{dl}A_{dl}^{N} \end{bmatrix} \in \mathbb{R}^{N\times(n_p+n_s)}. \tag{A.5}$$

The primary and secondary lifted matrices are lower triangular Toeplitz matrices:

$$M_{dl}^{p} = \begin{bmatrix} C_{dl}^{p}B_{dl}^{p} & D_{dl}^{p} & 0 & \cdots & 0 \\ C_{dl}^{p}A_{dl}^{p}B_{dl}^{p} & C_{dl}^{p}B_{dl}^{p} & D_{dl}^{p} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{dl}^{p}(A_{dl}^{p})^{N-1}B_{dl}^{p} & C_{dl}^{p}(A_{dl}^{p})^{N-2}B_{dl}^{p} & \cdots & C_{dl}^{p}B_{dl}^{p} & D_{dl}^{p} \end{bmatrix}, \tag{A.6}$$

and

$$M_{dl}^{s} = \begin{bmatrix} C_{dl}^{s}B_{dl}^{s} & D_{dl}^{s} & 0 & \cdots & 0 \\ C_{dl}^{s}A_{dl}^{s}B_{dl}^{s} & C_{dl}^{s}B_{dl}^{s} & D_{dl}^{s} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{dl}^{s}(A_{dl}^{s})^{N-1}B_{dl}^{s} & C_{dl}^{s}(A_{dl}^{s})^{N-2}B_{dl}^{s} & \cdots & C_{dl}^{s}B_{dl}^{s} & D_{dl}^{s} \end{bmatrix}. \tag{A.7}$$

Equivalently, for $1 \leq i \leq N$,

$$[M_{dl}^p]_{ij} = \begin{cases} C_{dl}^p (A_{dl}^p)^{i-j} B_{dl}^p, & 1 \leq j \leq i, \\ D_{dl}^p, & j = i+1, \\ 0, & j > i+1, \end{cases} \qquad [M_{dl}^s]_{ij} = \begin{cases} C_{dl}^s (A_{dl}^s)^{i-j} B_{dl}^s, & 1 \leq j \leq i, \\ D_{dl}^s, & j = i+1, \\ 0, & j > i+1, \end{cases} \tag{A.8}$$

for $1 \leq j \leq N+1$.

The delayed input sequences associated with these matrices are

$$U'^p = \begin{bmatrix} u^p_{k-N_d^p} \\ u^p_{k+1-N_d^p} \\ \vdots \\ u^p_{k+N-N_d^p} \end{bmatrix}, \qquad U'^s = \begin{bmatrix} u^s_{k-N_d^s} \\ u^s_{k+1-N_d^s} \\ \vdots \\ u^s_{k+N-N_d^s} \end{bmatrix}. \tag{A.9}$$

The constant term used in the delayed MPC optimization is then

$$c' = O_{dl} x'_k + M_{dl}^p U'^p, \tag{A.10}$$

so that

$$Y = c' - M_{dl}^s U'^s. \tag{A.11}$$

A.3. Relative-Delay Case and Causality

In the implementation used in this thesis, the secondary delay is set to zero, as stated in Equation (2.28). Therefore, $N_d = N_d^p$ and the primary input sequence becomes

$$U'^p = \begin{bmatrix} u^p_{k-N_d} \\ u^p_{k+1-N_d} \\ \vdots \\ u^p_{k+N-N_d} \end{bmatrix}. \tag{A.12}$$

The most recent primary sample needed by the prediction model is the last element of this sequence,

$$u^p_{k+N-N_d}. \tag{A.13}$$

At the computation instant k, the controller can use primary samples available up to time $k-1$. The condition

$$N + 1 \leq N_d \tag{A.14}$$

implies

$$k + N - N_d \leq k - 1. \tag{A.15}$$

Hence, every sample in U'^p is available when the secondary control sequence is computed. This is the detailed index argument behind the causality condition used in Section 2.2.

B | Input-to-State Stability of the Compensated Observer

B.1. Purpose of the Stability Analysis

This appendix analyzes the stability of the compensated observer used to provide the state estimate for the MPC prediction model. The objective is to show that the observer estimation error remains bounded in the presence of a bounded mismatch-prediction error. The result is established in the input-to-state stability sense with respect to the mismatch-prediction error.

Consider the augmented nominal model with an additive lumped mismatch term,

$$x_{k+1} = Ax_k + B_{up}u_k^p + B_{us}u_k^s + B^d d_k, \tag{B.1}$$

$$y_k = Cx_k + D^p u_k^p - D^s u_k^s, \tag{B.2}$$

where $x_k \in \mathbb{R}^n$ is the augmented state, u_k^p and u_k^s are the primary and secondary inputs, respectively, d_k is the lumped mismatch signal, and B^d is the corresponding mismatch direction.

The compensated observer is defined as

$$\hat{x}_{c,k+1} = A\hat{x}_{c,k} + B_{up}u_k^p + B_{us}u_k^s + Le_{c,k} + B^d\hat{d}_k, \tag{B.3}$$

where $\hat{x}_{c,k}$ is the compensated state estimate, L is the observer gain, $\hat{d}_k$ is the estimated mismatch signal, and $e_{c,k} = y_k - C\hat{x}_{c,k} - D^p u_k^p + D^s u_k^s$ is the innovation after removing the known direct-feedthrough terms.

Define the observer estimation error and the mismatch-prediction error as

$$e_{x,k} = x_k - \hat{x}_{c,k}, \tag{B.4}$$

$$\tilde{d}_k = d_k - \hat{d}_k. \tag{B.5}$$

Subtracting (B.3) from (B.1) gives

$$e_{x,k+1} = A(x_k - \hat{x}_{c,k}) - Le_{c,k} + B^d(d_k - \hat{d}_k). \tag{B.6}$$

Using the output equation and the definition of $e_{c,k}$, one obtains

$$e_{x,k+1} = (A - LC)e_{x,k} + B^d\tilde{d}_k. \tag{B.7}$$

Let

$$F = A - LC. \tag{B.8}$$

Then the observer-error dynamics can be written compactly as

$$e_{x,k+1} = Fe_{x,k} + B^d\tilde{d}_k. \tag{B.9}$$

The following assumptions are used.

Assumption B.1. *The pair (A, C) is detectable and the observer gain L is chosen such that the nominal observer-error matrix $F = A - LC$ is Schur stable, i.e., all eigenvalues of F lie strictly inside the unit circle.*

Assumption B.2. *The true lumped mismatch signal d_k is bounded, as commonly assumed in robust control formulations [29]. Hence, there exists a finite constant $\bar{d} > 0$ such that*

$$|d_k| \leq \bar{d}, \qquad \forall k \geq 0. \tag{B.10}$$

B.2. Boundedness of the LSTM-Based Mismatch Estimate

The boundedness of the LSTM-generated mismatch estimate can be established directly from the LSTM update equations. Let ϕ_k denote the input feature vector supplied to the

LSTM at time step k. For one LSTM layer, the update equations are written as

$$\begin{aligned} i_k &= \sigma(W_i[\phi_k, h_{k-1}] + b_i)\,, \\ f_k &= \sigma(W_f[\phi_k, h_{k-1}] + b_f)\,, \\ o_k &= \sigma(W_o[\phi_k, h_{k-1}] + b_o)\,, \\ \tilde{c}_k &= \tanh(W_c[\phi_k, h_{k-1}] + b_c)\,, \\ c_k &= f_k \odot c_{k-1} + i_k \odot \tilde{c}_k, \\ h_k &= o_k \odot \tanh(c_k), \end{aligned} \tag{B.11}$$

where i_k, f_k, and o_k are the input, forget, and output gates, respectively, c_k is the cell state, h_k is the hidden state, and $\odot$ denotes componentwise multiplication.

The sigmoid and hyperbolic tangent activation functions satisfy

$$0 < \sigma(s) < 1, \qquad -1 < \tanh(s) < 1, \qquad \forall s \in \mathbb{R}. \tag{B.12}$$

Therefore, the gate vectors are componentwise bounded between zero and one, while the candidate state $\tilde{c}_k$ is componentwise bounded between -1 and 1. The hidden state is given by

$$h_k = o_k \odot \tanh(c_k).$$

Since $0 < o_{k,j} < 1$ and $|\tanh(c_{k,j})| \leq 1$, each hidden-state component satisfies

$$|h_{k,j}| = |o_{k,j} \tanh(c_{k,j})| \leq 1, \qquad j = 1, \ldots, n_h, \tag{B.13}$$

where n_h is the hidden-state dimension. The cell state c_k is passed through the bounded hyperbolic tangent function before forming the hidden state. It follows that

$$\|h_k\|_2 \leq \sqrt{n_h}. \tag{B.14}$$

For the first predicted mismatch sample used by the compensated observer, let the corresponding linear readout layer be

$$\hat{d}_k = W_{\text{out}} h_k + b_{\text{out}}, \qquad W_{\text{out}} \in \mathbb{R}^{1 \times n_h}. \tag{B.15}$$

Assuming finite fixed readout parameters, the Cauchy–Schwarz inequality gives

$$\begin{aligned} |\hat{d}_k| &\leq \|W_{\text{out}}\|_2 \|h_k\|_2 + |b_{\text{out}}| \\ &\leq \|W_{\text{out}}\|_2 \sqrt{n_h} + |b_{\text{out}}| \\ &=: \bar{d}_{\text{LSTM}}. \end{aligned} \tag{B.16}$$

Thus, the LSTM-based mismatch estimate is uniformly bounded.

Combining (B.16) with Assumption B.2, namely

$$|d_k| \leq \bar{d},$$

the mismatch-prediction error

$$\tilde{d}_k = d_k - \hat{d}_k$$

satisfies

$$|\tilde{d}_k| \leq |d_k| + |\hat{d}_k| \leq \bar{d} + \bar{d}_{\text{LSTM}} =: \bar{\omega}, \qquad \forall k \geq 0. \tag{B.17}$$

Therefore, the mismatch-prediction error is bounded.

B.3. Input-to-State Stability of the Compensated Observer

The observer proof follows the standard input-to-state stability interpretation for discrete-time systems, where the mismatch-prediction error is treated as the external input to the estimation-error dynamics [19].

Theorem B.1. *Under Assumption B.1, the compensated observer-error dynamics* (B.9) *are input-to-state stable with respect to the mismatch-prediction error* $\tilde{d}_k$*. Under Assumption B.2 and finite fixed LSTM readout parameters,* $\tilde{d}_k$ *is bounded by Equation* (B.17)*; consequently, the observer estimation error is uniformly ultimately bounded. Moreover, if* $\tilde{d}_k \to 0$ *as* $k \to \infty$*, then* $e_{x,k} \to 0$*.*

Proof. Since F is Schur stable, for any symmetric positive definite matrix $Q_o = Q_o^T \succ 0$, there exists a unique symmetric positive definite matrix $P = P^T \succ 0$ satisfying the discrete-time Lyapunov equation

$$F^T P F - P = -Q_o. \tag{B.18}$$

Consider the quadratic Lyapunov function

$$V_k = e_{x,k}^T P e_{x,k}. \tag{B.19}$$

Since $P \succ 0$, the following bounds hold:

$$\lambda_{\min}(P)\|e_{x,k}\|^2 \leq V_k \leq \lambda_{\max}(P)\|e_{x,k}\|^2. \tag{B.20}$$

Using (B.9), the Lyapunov function at the next time step is

$$V_{k+1} = \left(Fe_{x,k} + B^d\tilde{d}_k\right)^T P \left(Fe_{x,k} + B^d\tilde{d}_k\right). \tag{B.21}$$

Therefore, the Lyapunov difference $\Delta V_k = V_{k+1} - V_k$ satisfies

$$\Delta V_k = e_{x,k}^T(F^T PF - P)e_{x,k} + 2e_{x,k}^T F^T PB^d\tilde{d}_k + \tilde{d}_k^T(B^d)^T PB^d\tilde{d}_k. \tag{B.22}$$

Substituting (B.18) into (B.22) gives

$$\Delta V_k = -e_{x,k}^T Q_o e_{x,k} + 2e_{x,k}^T F^T PB^d\tilde{d}_k + \tilde{d}_k^T(B^d)^T PB^d\tilde{d}_k. \tag{B.23}$$

Define

$$c_1 = \|F^T PB^d\|, \qquad c_2 = (B^d)^T PB^d. \tag{B.24}$$

Since $P \succ 0$, it follows that $c_2 \geq 0$. Using the Rayleigh–Ritz theorem and the Cauchy–Schwarz inequality, (B.23) yields

$$\Delta V_k \leq -\lambda_{\min}(Q_o)\|e_{x,k}\|^2 + 2c_1\|e_{x,k}\||\tilde{d}_k| + c_2|\tilde{d}_k|^2. \tag{B.25}$$

By Young's inequality, for any $\eta > 0$,

$$2c_1\|e_{x,k}\||\tilde{d}_k| \leq \eta\|e_{x,k}\|^2 + \frac{c_1^2}{\eta}|\tilde{d}_k|^2. \tag{B.26}$$

Choose $0 < \eta < \lambda_{\min}(Q_o)$. Then (B.25) becomes

$$\Delta V_k \leq -\sigma\|e_{x,k}\|^2 + \phi|\tilde{d}_k|^2, \tag{B.27}$$

where

$$\sigma = \lambda_{\min}(Q_o) - \eta > 0, \tag{B.28}$$

and

$$\phi = \frac{c_1^2}{\eta} + c_2 > 0. \tag{B.29}$$

From (B.20),

$$\|e_{x,k}\|^2 \geq \frac{V_k}{\lambda_{\max}(P)}. \tag{B.30}$$

Substituting (B.30) into (B.27) gives

$$\Delta V_k \leq -\frac{\sigma}{\lambda_{\max}(P)} V_k + \phi |\tilde{d}_k|^2. \tag{B.31}$$

Choose any constant α satisfying

$$0 < \alpha \leq \frac{\sigma}{\lambda_{\max}(P)}, \qquad 0 < \alpha < 1. \tag{B.32}$$

Then

$$V_{k+1} \leq (1-\alpha) V_k + \phi |\tilde{d}_k|^2. \tag{B.33}$$

Using the derived prediction-error bound (B.17),

$$V_{k+1} \leq (1-\alpha) V_k + \phi \bar{\omega}^2. \tag{B.34}$$

Recursively applying (B.34) yields

$$V_k \leq (1-\alpha)^k V_0 + \phi \bar{\omega}^2 \sum_{j=0}^{k-1} (1-\alpha)^j. \tag{B.35}$$

Since

$$\sum_{j=0}^{k-1} (1-\alpha)^j \leq \frac{1}{\alpha}, \tag{B.36}$$

we obtain

$$V_k \leq (1-\alpha)^k V_0 + \frac{\phi}{\alpha} \bar{\omega}^2. \tag{B.37}$$

Using the lower bound in (B.20), the estimation error satisfies

$$\|e_{x,k}\| \leq \sqrt{\frac{(1-\alpha)^k V_0}{\lambda_{\min}(P)} + \frac{\phi}{\alpha \lambda_{\min}(P)} \bar{\omega}^2}. \tag{B.38}$$

Taking the limit superior as $k \to \infty$ gives

$$\limsup_{k\to\infty} \|e_{x,k}\| \leq \sqrt{\frac{\phi}{\alpha \lambda_{\min}(P)}} \bar{\omega}. \tag{B.39}$$

Thus, the observer estimation error is uniformly ultimately bounded, and the ultimate bound is proportional to the mismatch-prediction error bound. This establishes input-to-state stability of the compensated observer-error dynamics with respect to $\tilde{d}_k$.

Finally, suppose that

$$\tilde{d}_k \to 0 \qquad \text{as} \qquad k \to \infty. \tag{B.40}$$

The solution of (B.9) is

$$e_{x,k} = F^k e_{x,0} + \sum_{j=0}^{k-1} F^{k-1-j} B^d \tilde{d}_j. \tag{B.41}$$

Since F is Schur stable,

$$F^k e_{x,0} \to 0. \tag{B.42}$$

Moreover, the convolution term in (B.41) also converges to zero because F is exponentially stable and $\tilde{d}_k \to 0$. Therefore,

$$e_{x,k} \to 0 \qquad \text{as} \qquad k \to \infty. \tag{B.43}$$

This completes the proof.

□

List of Figures

1 Feedforward active noise control configuration. The primary path $P(z)$ describes disturbance propagation to the error microphone, whereas the secondary path $S(z)$ describes propagation from the control loudspeaker to the error microphone. 1

2.1 Nominal MPC-based ANC 15
2.2 MPC-based ANC with primary path delay 19
2.3 Candidate causal sequence architectures 27
2.4 Proposed hybrid MPC–LSTM control architecture 34

3.1 Comparison of candidate mismatch-preview predictors 40
3.2 Time-domain residual error signals for different disturbance types. 42
3.3 Short-Time Fourier Transform spectra of the residual error signals for different disturbance types. All spectrograms are plotted using the same STFT parameters and the same color scale. 43
3.4 Time-varying attenuation of the residual error signals computed over short time windows. More negative ANR values indicate stronger attenuation relative to the ANC-off case. 45
3.5 Power spectral density of the residual error signals for different disturbance types. 46
3.6 Closed-loop mismatch prediction during the proposed Hybrid MPC-LSTM simulation. The black curve shows the true nonlinear mismatch $d(k)$ generated by the NARX-type mismatch model, while the blue dashed curve shows the first predicted LSTM mismatch sample $\hat{d}(k) = \hat{D}_{m,k}(1)$. The inset highlights a short time interval to show the local tracking behavior. . 48

4.1 Kundt's-tube measurement setup 50
4.2 Order-sweep results 52
4.3 Frequency-response comparison between the measured data and the selected state-space models. (a) Primary path. (b) Secondary path. 53
4.4 Validation of disturbance-entry directions 56

4.5 White-noise comparison of pure LSTM and MPC 61
4.6 Equivalent FIR/state-space mismatch used for LSTM training 62
4.7 Implementation time-domain residual responses 63
4.8 Implementation STFT residual spectra 65
4.9 Implementation time-varying attenuation 66
4.10 Implementation residual power spectral densities 68

List of Tables

2.1 Architecture-dependent hyperparameters considered in the causal sequence model benchmark. 30

3.1 Model-specific hyperparameter search space used for causal sequence model benchmarking. 38
3.2 Benchmark summary of candidate causal sequence models. 39
3.3 Band-limited noise reduction in the 100–800 Hz frequency range for different disturbance types. Values are reported in dB relative to the ANC-off case. 47

4.1 Validation results for candidate disturbance-entry directions using measured FIR data. 56
4.2 Training configuration of the pure LSTM controller baseline. 60
4.3 Band-limited noise reduction in the 100–800 Hz frequency range for different disturbance types. Values are reported in dB relative to the ANC-off case. 69

List of Symbols

A superscript p denotes a primary-path quantity, a superscript s denotes a secondary-path quantity, a hat denotes an estimate, and the subscript dl denotes a delay-free realization used in the delayed model. The symbol "–" indicates a dimensionless, normalized, or model-dependent quantity for which no unique SI unit is assigned.

Symbol	Description	Unit
Indices, dimensions, and operators		
k, t	Discrete-time or sequence indices	sample
i, j, τ	Generic sample, horizon, or matrix indices	–
$\mathcal{I}_r$	Sample-index set of the r-th one-second interval used for ANR calculation	sample
N	MPC prediction horizon	sample
N_d^p, N_d^s	Primary- and secondary-path delays	sample
N_d	Relative path delay, $N_d = N_d^p - N_d^s$	sample
N_p, N_s	Measured primary- and secondary-path impulse-response peak indices	sample
S_L	History length of the causal sequence-model input	sample
T	Training-segment length	sample
N_{seq}	Number of generated excitation sequences	–
$N_{\text{tr}}, N_{\text{te}}$	Numbers of training and test examples, respectively	–
N_{eval}	Number of evaluated prediction windows	–
n_p, n_s	Primary- and secondary-path state dimensions	–
n_h	LSTM hidden-state dimension	–
ℓ	Memory length of the NARX regressor; also a generic layer index where stated	sample
K	FxNLMS filter length or convolutional kernel length, according to context	sample
f_s	Sampling frequency	Hz
z	Complex variable of a discrete-time transfer function	–
I	Identity matrix of the required dimension	–
$(\cdot)^\top$	Matrix or vector transpose	–
$(\cdot)^*$	Complex conjugate	–
$\|\cdot\|_2$	Euclidean vector norm or induced matrix 2-norm	–

Continued on the next page

Symbol	Description	Unit
$E\{\cdot\}$	Ensemble or segment-averaging operator	–
$*$	Discrete convolution operator	–
$\odot$	Componentwise multiplication	–
$\sigma(\cdot)$	Logistic sigmoid activation function	–
$\tanh(\cdot)$	Hyperbolic tangent activation function	–
$\lambda_{\min}(\cdot), \lambda_{\max}(\cdot)$	Minimum and maximum eigenvalues	–
Acoustic signals, states, and mismatch quantities		
u_k^p	Primary/reference input at sampling instant k	norm.
u_k^s	Secondary control input at sampling instant k	norm.
$u_{k,\mathrm{unc}}^s$	First element of the unconstrained MPC control sequence	norm.
$u_k^{s,\mathrm{sat}}$	Saturated secondary control input applied to the actuator	norm.
$u_{\min}^s, u_{\max}^s$	Minimum and maximum admissible secondary control inputs	norm.
$U_{s,\max}$	Output-amplitude bound of the pure LSTM controller	norm.
y_k^p, y_k^s	Primary disturbance and secondary anti-noise contributions at the error sensor	norm.
y_k	Residual error signal, $y_k = y_k^p - y_k^s$	norm.
$y_{\mathrm{off},k}$	Residual signal with ANC disabled	norm.
x_k^p, x_k^s	Primary- and secondary-path state vectors	–
x_k	Augmented ANC state vector	–
x_k'	Delayed augmented state vector	–
$x_{\mathrm{nom},k}$	State of the nominal mismatch-free model	–
$\hat{x}_{c,k}$	Compensated observer state estimate	–
$e_{c,k}$	Observer output correction residual	norm.
$e_{m,k}$	State mismatch, $x_k - x_{\mathrm{nom},k}$	–
$e_{\mathrm{proxy},k}$	Reconstructed proxy of the state mismatch	–
$e_{x,k}$	Observer state-estimation error, $x_k - \hat{x}_{c,k}$	–
Δy_k	Output mismatch, $y_k - y_{\mathrm{nom},k}$	norm.
d_k	Scalar lumped model-mismatch signal	norm.
$\hat{d}_k$	Estimated lumped mismatch	norm.
$d_{\mathrm{proxy},k}$	Offline reconstructed scalar mismatch proxy	norm.
$\tilde{d}_k$	Mismatch-prediction error, $d_k - \hat{d}_k$	norm.
$\tilde{x}_k$	Additive state-mismatch contribution, $B^d d_k$	–
$\bar{d}$	Upper bound on the true mismatch magnitude	norm.
$\bar{d}_{\mathrm{LSTM}}$	Derived upper bound on the LSTM mismatch estimate	norm.
$\bar{\omega}$	Upper bound on the mismatch-prediction error	norm.
ξ_k	NARX regressor containing recent u^p, u^s, and y samples	–
v_k	Unsaturated output of the quadratic NARX mismatch model	–
$c_0, c_1, H_{\mathrm{pol}}$	Constant, linear coefficient, and quadratic coefficient matrix of the NARX model	–
β	NARX mismatch saturation-scaling parameter	–

Continued on the next page

Symbol	Description	Unit
State-space, delay, and prediction-model quantities		
A^p, B^p, C^p, D^p	Primary-path state-space matrices	–
A^s, B^s, C^s, D^s	Secondary-path state-space matrices	–
A, B_{up}, B_{us}, C	Augmented state, primary-input, secondary-input, and output matrices	–
B^d (B_d)	Known mismatch/disturbance-entry direction; B_d is used in Appendix B	–
L	Steady-state Kalman observer gain	–
F	Observer-error matrix, $F = A - LC$	–
$A_{dl}, B_{dl}^{p,a}, B_{dl}^{s,a}, C_{dl}$	Augmented delay-free matrices used in the delayed formulation	–
$\{A, B, C, D\}_{dl}^{p,s}$	Path-specific delay-free state-space matrices	–
$P(z), S(z)$	Primary and secondary acoustic-path transfer functions	–
$H_{\text{meas}}^{p,s}(z)$	Measured primary- and secondary-path transfer functions	–
$H_{\text{delayless}}^{p,s}(z)$	Delay-free primary- and secondary-path transfer functions	–
$G(z)$	Identified parametric discrete-time transfer function	–
a_i, b_i	Denominator and numerator coefficients of $G(z)$	–
q, r	Numerator and denominator orders of $G(z)$	–
$s(n)$	Identified secondary-path impulse response	–
$x_{\text{in}}, x_{\text{out}}$	Excitation and measured response used for path identification	norm.
$X_{\text{in}}, X_{\text{out}}$	Fourier transforms of x_{in} and x_{out}	–
$H_{\text{meas}}(e^{j\omega})$	Measured frequency-response estimate	–
ω	Angular frequency	rad/s
U^p, U^s	Lifted primary- and secondary-input sequences	norm.
U'^p, U'^s	Delay-shifted lifted input sequences	norm.
Y	Lifted predicted residual-output sequence	norm.
D_m	True mismatch sequence over the prediction horizon	norm.
$\hat{D}_{m,k}$	Predicted mismatch sequence available at instant k	norm.
$D_{\text{proxy},k}$	Supervised proxy target sequence	norm.
D_{eq}	Equivalent disturbance sequence reconstructed from FIR data	norm.
O, O_{dl}	Nominal and delayed observability prediction matrices	–
M^p, M^s, M^d	Primary, secondary, and mismatch lifted prediction matrices	–
M_{dl}^p, M_{dl}^s	Delayed primary- and secondary-path prediction matrices	–
$c, c', c_m, c_{\text{comp}}$	Nominal, delayed, mismatch-affected, and observer-compensated MPC constant terms	norm.
Q, R	Residual-error and control-effort weighting matrices	–
H	MPC Hessian-like matrix, $(M^s)^\top Q M^s + R$	–
$g, g', g_m, g_{\text{comp}}$	Nominal, delayed, mismatch-affected, and compensated MPC gradient vectors	–
g_d	Output-space mismatch direction, CB^d	–
J	MPC or neural-controller objective function, according to context	–

Continued on the next page

Symbol	Description	Unit
$U^{s*}_{\mathrm{unc}}, U^{s*}$	Unconstrained nominal and mismatch-compensated optimal control sequences	norm.
Learning quantities		
ϕ_k	Causal feature vector at instant k	–
Φ_k	Window of the most recent S_L feature vectors	–
$f_\theta(\cdot)$	Causal sequence predictor parameterized by θ	–
$\theta, \theta^\star$	Trainable parameters and selected trained parameters	–
h_k, c_k	Recurrent hidden state and LSTM cell state	–
i_k, f_k, o_k	LSTM input, forget, and output gates	–
z_k, r_k	GRU update and reset gates when used in the GRU equations; z_k also denotes the raw pure-LSTM output where stated	–
$\tilde{c}_k, \tilde{h}_k$	Candidate LSTM cell state and candidate GRU hidden state	–
$W_\bullet, b_\bullet$	Neural-network weight matrices and bias vectors	–
$W_{\mathrm{out}}, b_{\mathrm{out}}$	Linear LSTM readout weight and bias	–
$\hat{d}^{(1)}_k$	First sample of the predicted mismatch preview	norm.
μ	FxNLMS step size	–
η	Neural-network learning rate; Young's-inequality parameter in Appendix B	–
p_{drop}	Dropout probability	–
$\mathcal{L}(\theta)$	Mean-squared sequence-prediction loss	–
$\mathcal{A}, \mathcal{H}_a$	Candidate architecture set and hyperparameter space for architecture a	–
$\mathcal{I}$	Valid sample-index set used in pure-LSTM training	–
$\lambda_u, \lambda_{\Delta u}$	Pure-LSTM control-effort and control-variation loss weights	–
λ_d	Regularization weight in equivalent-disturbance reconstruction	–
q_t, k_t, v_t	Attention query, key, and value vectors	–
$\alpha_{t,\tau}$	Causal attention weight from instant t to τ	–
d_a, d_ℓ	Attention dimension and convolution dilation factor	–
Performance and stability quantities		
RMSE	Root-mean-square prediction error	norm.
MaxAE	Maximum absolute prediction error	norm.
ANR	Time-varying attenuation relative to ANC off	dB
PSD	Power spectral density	dB/Hz
$Y_{\mathrm{FIR}}, Y_{\mathrm{nom}}, Y_{\mathrm{corr}}$	FIR-plant, nominal, and disturbance-corrected output predictions	norm.
ΔY	FIR/state-space prediction mismatch	norm.
$\mathrm{Fit}_{\mathrm{dB}}$	Improvement from nominal to corrected prediction RMSE	dB
R^2	Coefficient of determination for mismatch reconstruction	–
Q_o, P	Positive-definite Lyapunov-equation weighting and solution matrices	–
V_k	Quadratic Lyapunov function of the observer error	–

Continued on the next page

Symbol	Description	Unit
c_1, c_2	Constants used to bound the Lyapunov cross and input terms	–
σ, ϕ	Positive decay and input-gain constants in the ISS proof	–
α	Contraction factor used in the recursive Lyapunov bound	–

List of Abbreviations

Abbreviation	Meaning
1D-CNN	One-dimensional convolutional neural network
ANC	Active noise control
CB	Computational burden
CNN	Convolutional neural network
DL	Deep learning
FC	Fully connected
FIR	Finite impulse response
FRF	Frequency response function
FxLMS	Filtered-x least-mean-square
FxNLMS	Filtered-x normalized least-mean-square
GRU	Gated recurrent unit
ISS	Input-to-state stability
LMS	Least-mean-square
LSTM	Long short-term memory
MaxAE	Maximum absolute error
MPC	Model predictive control
MSE	Mean-squared error
NARX	Nonlinear autoregressive model with exogenous inputs
PSD	Power spectral density
ReLU	Rectified linear unit
RMSE	Root-mean-square error
RNN	Recurrent neural network
SISO	Single-input single-output
STFT	Short-time Fourier transform
TCN	Temporal convolutional network
TT	Training time