Computationally Efficient Multi-Model Adaptive Control and Estimation for Uncertain Systems

by

Farid Mafi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2025

© Farid Mafi 2025

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Xianke Lin Professor, Dept. of Automotive and Mechatronics Engineering, University of Ontario Institute of Technology
Supervisor:	Amir Khajepour Professor, Dept. of Mechanical and Mechatronics Engineering, University of Waterloo
Co-supervisor:	Mohammad Pirani Professor, Dept. of Mechanical Engineering, University of Ottawa
Internal Member:	Behrad Khamesee Professor, Dept. of Mechanical and Mechatronics Engineering, University of Waterloo
Internal Member:	Eihab Abdel-Rahman Professor, Dept. of Systems Design Engineering, University of Waterloo
Internal-External Member:	Michael Fisher Professor, Dept. of of Electrical and Computer Engineering, University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Multi-model adaptive technique offers a powerful framework for handling uncertainties in dynamic systems by employing multiple models that represent different operating conditions. However, despite strong theoretical foundations, practical deployment has been severely limited by the curse of dimensionality—the exponential growth in computational requirements as system complexity increases, creating a major obstacle for real-time implementation. This thesis presents a comprehensive framework to address this computational challenge through three related contributions, achieving dramatic reductions in complexity while maintaining or improving system performance.

The main insight is that traditional multi-model approaches use far more models than necessary. Instead of using every possible model combination to cover parameter uncertainties, this thesis shows that carefully chosen subsets can achieve the same coverage with much less computation. This shift from using all models to selecting the right models makes multi-model control practical for real-time systems with limited resources.

To this end, geometric methods have been developed that analyze how models cover the space of possible system behaviors. Using computational geometry principles, new Enclosed Polytope with Minimum Models (EPMM) algorithms find the smallest set of models that still covers the uncertainty space adequately. These algorithms work like placing sensors strategically—finding the fewest locations needed to monitor an entire area. An optimization framework extends this idea to continuous parameter spaces, while a transfer function method handles high-dimensional systems efficiently by focusing on input-output relationships rather than full state representations.

To overcome the fundamental scaling limitations in high-dimensional spaces, the Parameter -Tying Theorem has been developed as a theoretical innovation showing that changing to the right coordinates can considerably simplify high-dimensional uncertainty spaces. By analyzing systems in controllable canonical form and finding monotonicity properties in how parameters affect the system, the theorem proves that the number of required models can drop from exponential in the number of parameters to potentially constant. The framework extends through five conditions—including affine relationships, symmetry, and coordinated parameter variations—making it applicable beyond strictly monotonic systems.

Furthermore, a unified estimation framework has been designed to tackle the integration of physics-based and data-driven models. A consensus multi-model Kalman filter combines different model types based on how well they perform. Two methods enable proper uncertainty handling: Koopman operator-based linearization allows analytical covariance propagation for neural networks and other nonlinear data-driven models, while an

ensemble-based approach provides model-independent uncertainty quantification without needing offline training. The consensus fusion automatically adjusts model weights based on prediction errors, ensuring smooth transitions between models as conditions change. Extensive experimental data from an electric all-wheel-drive vehicle under extreme conditions shows major performance improvements over traditional single-model approaches.

This thesis transforms multi-model approach from an attractive theory with limited practical use into a viable solution for real-world applications. By combining geometric insight, coordinate transformation theory, and heterogeneous model integration, it addresses the fundamental implementation barriers. The developed frameworks maintain mathematical rigor while achieving the computational efficiency needed by modern embedded systems. These advances enable robust adaptive control across diverse operating conditions, with immediate applications in autonomous vehicles, renewable energy systems, and other areas where handling uncertainty is critical. The principles established here provide a foundation for addressing high-dimensional uncertainty in complex dynamical systems across engineering fields.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisors, Prof. Amir Khajepour and Prof. Mohammad Pirani, for their continuous support throughout my Ph.D. studies and related research—for their patience, motivation, and immense knowledge.

In addition to my advisors, I would like to thank my committee members: Prof. Behrad Khamesee, Prof. Eihab Abdel-Rahman, and Prof. Michael Fisher, for their insightful comments and encouragement, which motivated me to broaden my research from various perspectives.

Last but not least, I would like to thank my family for their unwavering support and encouragement throughout this journey. I deeply appreciate their sacrifices, without which I would not have been able to reach this stage.

Dedication

To my mother, the one I love forever...

Table of Contents

Examining Committee ii

Author's Declaration iii

Abstract iv

Acknowledgements vi

Dedication vii

List of Figures xiii

List of Tables xvi

Notation xvii

1 Introduction 1

1.1 Motivation 1

1.2 Objectives 3

1.2.1 Optimal Model Selection and Positioning 4

1.2.2 Dimension Reduction through Coordinate Transformations 4

1.2.3 Integration of Physics-Based and Data-Driven Models 5

1.3 Thesis Outline 6

2 Background and Literature Review **8**
2.1 Multi-Model Adaptive Control: Foundations and Evolution 8
2.1.1 Motivation and Traditional Approaches 8
2.1.2 Switching-Based Multi-Model Approaches 9
2.1.3 Blending-Based Multi-Model Structures 10
2.1.4 Applications in Vehicle Dynamics and Power Systems 11
2.2 Computational Efficiency and Model Selection in Multi-Model Systems . . 13
2.2.1 The Curse of Dimensionality Challenge 13
2.2.2 Geometric Approaches to Model Selection 13
2.3 Integration of Physics-Based and Data-Driven Approaches 14
2.3.1 The Emergence of Hybrid Modeling Paradigms 14
2.3.2 Hybrid Physics-Data Driven Modeling Frameworks 15
2.3.3 Multi-Model Kalman Filtering with Heterogeneous Models 17
2.3.4 Uncertainty Quantification and Fault Tolerance 18

3 Optimal Multi-Model Selection via Geometric Analysis **20**
3.1 Introduction 20
3.2 Problem Formulation and Mathematical Preliminaries 22
3.2.1 System Description and Uncertainty Characterization 22
3.2.2 Multi-Model Framework 23
3.2.3 Theoretical Foundation for Multi-Model Inclusion 23
3.2.4 Extension to Non-linear Uncertain Plants 24
3.2.5 Extension to Input Matrices 26
3.2.6 Weight Determination and Error Dynamics 27
3.2.7 Parameter Uncertainty Space 27
3.3 Enclosed Convex Polytope with Minimum Models (EPMM) Algorithms . . 28
3.3.1 EPMM Algorithm 1: Pairwise Distance Maximization 29
3.3.2 EPMM Algorithm 2: Distance Sum Maximization 30

3.3.3 Comparative Analysis of EPMM Algorithms . . . 31
3.4 Optimization Approach for Interval Uncertainties . . . 34
3.4.1 Problem Formulation . . . 35
3.4.2 Solution Methodology . . . 35
3.4.3 Handling Subspace Constraints . . . 37
3.5 Transfer Function-Based Dimension Reduction . . . 37
3.5.1 Transfer Function and Steady-State Gain Analysis . . . 38
3.5.2 Model Selection in Transfer Function Space . . . 38
3.6 Application to Vehicle Lateral Dynamics . . . 39
3.6.1 System Description . . . 39
3.6.2 Multi-Model Structure Implementation . . . 40
3.6.3 Simulation Validation . . . 41
3.6.4 Transfer Function Analysis in Vehicle Lateral Dynamics . . . 43
3.7 Application to Low-Inertia Power Networks . . . 45
3.8 Summary and Remarks . . . 47

4 Dimension Reduction through Parameter-Tying and Canonical Transformations 53
4.1 Introduction . . . 53
4.2 Mathematical Framework and Problem Formulation . . . 54
4.2.1 System Description . . . 54
4.2.2 Parameter Uncertainty Characterization . . . 55
4.2.3 Canonical Form Transformation . . . 56
4.2.4 Parameter Uncertainty Space in Canonical Coordinates . . . 57
4.3 The Parameter-Tying Theorem . . . 57
4.3.1 Statement of the Theorem . . . 58
4.3.2 Proof of the Parameter-Tying Theorem . . . 58
4.4 Extensions and Generalizations . . . 68

4.4.1 Coordinated Parameter-Element Functions . . . 68
4.4.2 Conditions for Coordinated Extrema . . . 69
4.4.3 Independence and Completeness . . . 71
4.5 Criterion for Verifying Plant Inclusion . . . 72
4.5.1 The Plant Inclusion Theorem . . . 72
4.5.2 Proof of the Plant Inclusion Theorem . . . 73
4.6 Application to Vehicle Dynamics . . . 77
4.6.1 Motivation and Challenges . . . 77
4.6.2 Vehicle Dynamic Model . . . 79
4.6.3 State-Space Representation . . . 81
4.6.4 Application of Parameter-Tying Theorem . . . 83
4.7 Simulation Results . . . 86
4.7.1 Simulation Setup . . . 87
4.7.2 Estimation Performance . . . 89
4.8 Summary and Remarks . . . 91

5 Consensus-Based Integration of Heterogeneous Models 93
5.1 Introduction . . . 93
5.2 Multi-Model Ensemble Kalman Filter Framework . . . 94
5.2.1 Problem Formulation . . . 94
5.2.2 Multi-Model Ensemble Framework . . . 95
5.3 Method 1: Koopman Linearization for Data-Driven Models . . . 96
5.3.1 Theoretical Foundation . . . 96
5.3.2 Lifting to Higher-Dimensional Space . . . 97
5.3.3 Learning the Koopman Representation . . . 98
5.3.4 Covariance Propagation via Linearization . . . 99
5.3.5 Choice of Lifting Functions . . . 100
5.4 Method 2: Ensemble-Based Covariance Propagation . . . 101

5.4.1 Theoretical Foundation . . . 101
5.4.2 Ensemble Generation . . . 102
5.4.3 Ensemble Propagation . . . 103
5.4.4 Statistical Estimation . . . 103
5.4.5 Advanced Ensemble Techniques . . . 105
5.5 Consensus Multi-Model Fusion . . . 106
5.5.1 Iterative Fusion Procedure . . . 106
5.5.2 Observation Update . . . 107
5.5.3 Implementation and Method Comparison . . . 107
5.5.4 Algorithmic Summary . . . 108
5.6 Experimental Validation . . . 109
5.6.1 Experimental Setup . . . 109
5.6.2 State Estimation Results . . . 111
5.7 Summary and Remarks . . . 113

6 Conclusions and Future Directions **118**
6.1 Conclusions . . . 118
6.2 Future Work . . . 120

References **123**

List of Figures

1.1 Overview of multi-model adaptive control challenges and the three-part solution approach: geometric optimization for model selection, coordinate transformations for dimension reduction, and consensus fusion for heterogeneous model integration. 3

3.1 Representation of non-linear uncertain plant through time-varying weighted combination of fixed linear identification models. The weights $w_i(t)$ evolve through time via sliding window least squares optimization, enabling the linear models to collectively track and represent the non-linear dynamics. . 26

3.2 Illustration of EPMM 1 algorithm steps in 2D parameter space. (a) Initial convex hull with all models, (b) Extremum point selection, (c) Removal of nearby points, (d) Pairwise distance maximization and final selection. . . . 32

3.3 EPMM 2 selection strategy. The algorithm selects the point (shown in blue) that maximizes the sum of distances to all previously selected points (shown in green), ensuring optimal space filling. 34

3.4 Bicycle model representation showing key geometric parameters and force components. The model simplifies the four-wheel vehicle to a two-wheel representation while preserving essential lateral dynamics. 40

3.5 Multi-model structure for vehicle lateral dynamics estimation. The unknown plant (vehicle) receives steering input δ and produces outputs (lateral velocity and yaw rate). Multiple models run in parallel, each representing different parameter combinations. 41

3.6 Genetic algorithm convergence showing (a) best fitness value evolution over generations and (b) distribution of optimal parameter values for the 4 selected models. 43

3.7 2D projections of the convex hulls comparing: (a) Complete set of 125 models (blue) vs. (b) Optimally selected 4 models (red). The projections show excellent coverage despite the dramatic reduction in model count. 44

3.8 Double lane change maneuver trajectory showing target path and actual vehicle response for the three test scenarios. 45

3.9 Lateral velocity estimation results showing: (top) true vs. estimated velocity, (bottom) model weights evolution. The 4-model system successfully tracks the true velocity across all conditions. 49

3.10 Transfer function-based estimation results using only 2 models. The approach maintains good tracking performance while achieving maximum computational efficiency. 50

3.11 Two-generator system configuration. The generators are connected through a transmission line with susceptance b, representing the simplest non-trivial power network. 51

3.12 EPMM 1 algorithm progression for power system model selection. (a) Initial 100 models and their convex hull, (b) After iteration 1: extremum points selected, (c) After iteration 2: 8 models selected achieving 89% volume coverage. 51

3.13 Observed angle difference signal showing the system response to stochastic power disturbances. 52

3.14 Angular velocity estimation for both generators using the reduced 8-model set. The estimator accurately tracks the true velocities despite using only 8% of the original models. 52

4.1 Multi-model structure for vehicle state estimation. The plant (vehicle simulated in CarSim) receives steering input δ and produces measurable yaw rate r. Two identification models run in parallel, with their outputs combined through adaptive weights to estimate unmeasured states (sideslip angle, roll angle, roll rate). The dramatic reduction from 64 to 2 models is achieved through application of the Parameter-Tying Theorem. 86

4.2 Simulated vehicle behavior in CarSim at constant speeds of: (a) 50 km/h and (b) 100 km/h. 88

4.3 Estimation results at 50 km/h on mountain road. The multi-model adaptive law demonstrates effective state estimation using only yaw rate measurements. 89

4.4 Estimation results at 100 km/h on mountain road. The estimator maintains effective performance despite increased dynamic complexity at higher speeds. 91

5.1 Schematic of Method 2: Ensemble-Based Covariance Propagation. 104

5.2 Electric all-wheel-drive Equinox test vehicle used for experimental validation. The vehicle features independent wheel motors, comprehensive sensor suite, and robotic steering system enabling repeatable test maneuvers under extreme conditions. 109

5.3 Ground truth data from the unseen test maneuver on icy road. This dataset serves as the benchmark for evaluating estimation performance. 112

5.4 Performance of Method 1: Koopman Linearization for Data-Driven Models. 114

5.5 Performance of Method 2: Ensemble-Based Covariance Propagation. 115

List of Tables

3.1 EPMM 1 Algorithm 30
3.2 Key Functions Used in EPMM Algorithms 31
3.3 EPMM 2 Algorithm 33
3.4 Optimization Procedure for Interval Uncertainties 36
3.5 Genetic Algorithm Configuration 37
3.6 Genetic Algorithm Parameters for Vehicle Application 42
3.7 Estimation Performance Comparison 42
3.8 Parameter Ranges for Low-Inertia Power System 46

4.1 Parameter Variations from Nominal Values 84
4.2 Vehicle Parameters 87
4.3 Test Scenario Specifications 88

5.1 Specifications of the electric all-wheel-drive test vehicle 110

Notation

Throughout this thesis, the following mathematical notations and symbols are used:

- **Sets and Spaces:**
  - $\mathbb{R}$ denotes the set of real numbers
  - $\mathbb{R}^n$ denotes the n-dimensional Euclidean space
  - $\mathbb{R}^{n\times m}$ denotes the space of $n \times m$ real matrices
  - $M \subset \mathbb{R}^k$ denotes the parameter uncertainty space
  - $\Theta \subset \mathbb{R}^n$ denotes the parameter uncertainty space in canonical coordinates
  - S_n denotes the symmetric group of degree n
  - Δ_{n-1} denotes the standard $(n-1)$-simplex
  - $\mathcal{C}(n,k)$ denotes the set of all k-element subsets of $\{1,\ldots,n\}$
  - $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
- **Matrices and Vectors:**
  - Matrices are denoted by uppercase letters (e.g., A, B, T)
  - Vectors are denoted by lowercase letters (e.g., x, y, $\mathbf{m}$)
  - 0_n denotes the $n \times n$ zero matrix
  - I_n denotes the $n \times n$ identity matrix
  - $[\cdot]^T$ denotes the transpose of a matrix or vector
  - $[\cdot]^{-1}$ denotes the matrix inverse
  - $[\cdot]^\dagger$ denotes the Moore-Penrose pseudo-inverse
  - $A_p(\mathbf{m})$ denotes the plant state matrix dependent on parameters $\mathbf{m}$
  - $B_p(\mathbf{m})$ denotes the plant input matrix dependent on parameters $\mathbf{m}$
  - $\bar{A}$ denotes a matrix in canonical (companion) form
  - $E(t) = [e_1(t)\ e_2(t)\ \cdots\ e_N(t)]$ denotes the error vector
  - A_S denotes the principal submatrix of A indexed by set S
- **Functions and Operators:**

- $\text{Conv}\{\cdot\}$ denotes the convex hull of a set
- $\det(\cdot)$ denotes the determinant of a matrix
- $\text{tr}(\cdot)$ denotes the trace of a matrix
- $\|\cdot\|$ denotes the Euclidean norm
- $\|\cdot\|_{\mathbf{W}}$ denotes the weighted norm, $\|\mathbf{x}\|_{\mathbf{W}}^2 = \mathbf{x}^T\mathbf{W}\mathbf{x}$
- $\text{Vol}(\cdot)$ denotes the volume of a set
- $\text{vert}(\cdot)$ denotes the vertices of a polytope
- $\text{Ext}(\cdot)$ denotes the extremum points of a set
- $\text{sgn}(\sigma)$ denotes the signature of permutation σ
- $\Pi_l : M \to M \setminus \{m_l\}$ denotes the projection operator removing the l-th component
- $g_{l,ij}(m_l)$ denotes the parameter-element function for matrix entry a_{ij}
- $\mathcal{K}$ denotes the Koopman operator
- $\psi(\cdot)$ denotes lifting functions for Koopman linearization

- **Set Operations:**
  - $\cup$ denotes the union of sets
  - $\cap$ denotes the intersection of sets
  - $\setminus$ denotes set difference
  - $\subseteq$ denotes subset
  - $\in$ denotes set membership
  - $\times$ denotes the Cartesian product
  - $|\cdot|$ denotes the cardinality of a set
  - $\circ$ denotes function composition

- **System and Control Notations:**
  - $x_p(t) \in \mathbb{R}^n$ denotes the plant state vector
  - $x_i(t) \in \mathbb{R}^n$ denotes the i-th identification model state vector
  - $\mathbf{s}_t \in \mathbb{R}^n$ denotes the true system state vector at discrete time t
  - $\mathbf{x}_{m,t}^f \in \mathbb{R}^n$ denotes the forecast state from model m at time t

- $\mathbf{x}^a_{m,t} \in \mathbb{R}^n$ denotes the analysis (updated) state from model m at time t
- $\mathbf{x}^f_{1:m,t} \in \mathbb{R}^n$ denotes the consensus forecast state from models 1 to m
- $\mathbf{y}_t \in \mathbb{R}^p$ denotes the measurement vector at time t
- $d(t) \in \mathbb{R}^h$ denotes the input vector
- $\mathbf{u}_t \in \mathbb{R}^q$ denotes the control input vector at time t
- $e_i(t) = x_i(t) - x_p(t)$ denotes the identification error
- $\mathbf{d}_{m,t} = \mathbf{y}_t - \mathbf{H}_t \mathbf{x}^f_{m,t}$ denotes the innovation vector for model m
- $w_i(t)$ denotes the scalar weight for the i-th model
- $\tilde{w}_i$ denotes the matrix-valued weight in canonical coordinates
- T_p, T_i denote transformation matrices to canonical form
- $S_i = T_p T_i^{-1}$ denotes the similarity transformation matrix
- $\mathbf{T}_m$ denotes the transformation from reference space to model m space
- Σ_p denotes the uncertain plant system
- Σ_i denotes the i-th identification model system
- $f_m(\cdot)$ denotes the state transition function for model m
- $f_{\text{true}}(\cdot)$ denotes the true (unknown) system dynamics
- $h(\cdot)$ denotes the measurement function

- **Covariance and Noise Matrices:**
  - $\mathbf{P}^f_{m,t} \in \mathbb{R}^{n \times n}$ denotes the forecast error covariance for model m
  - $\mathbf{P}^a_{m,t} \in \mathbb{R}^{n \times n}$ denotes the analysis error covariance for model m
  - $\mathbf{P}^f_{1:m,t} \in \mathbb{R}^{n \times n}$ denotes the consensus forecast covariance
  - $\mathbf{Q}_{m,t} \in \mathbb{R}^{n \times n}$ denotes the process noise covariance for model m
  - $\mathbf{Q}_{\text{true}} \in \mathbb{R}^{n \times n}$ denotes the true process noise covariance
  - $\mathbf{R}_t \in \mathbb{R}^{p \times p}$ denotes the measurement noise covariance
  - $\mathbf{S}_{m,t} \in \mathbb{R}^{p \times p}$ denotes the theoretical innovation covariance
  - $\hat{\mathbf{S}}_{m,t} \in \mathbb{R}^{p \times p}$ denotes the empirical innovation covariance
  - $\mathbf{S} \in \mathbb{R}^{n \times n}$ denotes the ensemble sampling covariance matrix

- **Kalman Filter Matrices:**

- $\mathbf{F}_{m,t} \in \mathbb{R}^{n \times n}$ denotes the Jacobian of f_m at time t
- $\mathbf{H}_t \in \mathbb{R}^{p \times n}$ denotes the measurement matrix at time t
- $\mathbf{K}_{m,t} \in \mathbb{R}^{n \times n}$ denotes the model incorporation gain matrix
- $\mathbf{K}_t \in \mathbb{R}^{n \times p}$ denotes the Kalman gain for measurement update
- $\mathbf{I} \in \mathbb{R}^{n \times n}$ denotes the identity matrix (size from context)

- **Parameter and Uncertainty Notations:**
  - $\mathbf{m} = (m_1, m_2, \ldots, m_k) \in \mathbb{R}^k$ denotes the parameter uncertainty vector
  - $[p_i, q_i]$ denotes the bounds for parameter m_i
  - $\hat{\mathbf{m}}_l = \Pi_l(\mathbf{m})$ denotes the parameter vector with l-th component removed
  - $\theta = [\alpha_1 \ \alpha_2 \ \cdots \ \alpha_n]^T$ denotes the canonical parameter vector
  - c_k denotes the k-th coefficient of the characteristic polynomial
  - lb_{ij}, ub_{ij} denote lower and upper bounds for matrix element a_{ij}

- **Koopman Operator Notations:**
  - $\mathbf{z}_t \in \mathbb{R}^d$ denotes the lifted state in Koopman space, where $d > n$
  - $\mathbf{A}_K \in \mathbb{R}^{d \times d}$ denotes the autonomous dynamics matrix in Koopman space
  - $\mathbf{B}_K \in \mathbb{R}^{d \times q}$ denotes the control input matrix in Koopman space
  - $\mathbf{H}_K \in \mathbb{R}^{d \times dq}$ denotes the bilinear term matrix in Koopman space
  - $\mathbf{C}_K \in \mathbb{R}^{n \times d}$ denotes the output matrix for Koopman states
  - $\mathbf{A}_{K,t} \in \mathbb{R}^{d \times d}$ denotes the time-varying linearized dynamics
  - $\mathbf{v}_t \in \mathbb{R}^d$ denotes the deterministic input term in LTV system
  - $\mathbf{w}_t \in \mathbb{R}^d$ denotes the process noise in Koopman space
  - $\mathbf{Q}_K \in \mathbb{R}^{d \times d}$ denotes the process noise covariance in Koopman space
  - $\mathbf{R}_K \in \mathbb{R}^{n \times n}$ denotes the reconstruction error covariance
  - $\mathbf{P}_t^z \in \mathbb{R}^{d \times d}$ denotes the covariance in lifted space

- **Learning and Optimization Notations:**
  - L denotes the training dataset size
  - $\mathbf{X}_d \in \mathbb{R}^{d \times L}$ denotes the matrix of successor lifted states

- $\tilde{\mathbf{X}}_d \in \mathbb{R}^{d \times L}$ denotes the matrix of predecessor lifted states
- $\mathbf{U} \in \mathbb{R}^{q \times L}$ denotes the matrix of control inputs
- $\mathbf{Y} \in \mathbb{R}^{n \times L}$ denotes the matrix of original states
- V_1, V_2 denote the data fitting and regularization terms in optimization
- $\lambda_{\{A,B,H,C,Q,R\}}$ denote regularization hyperparameters
- α denotes the blending parameter for stability enhancement
- γ denotes the adaptation rate for covariance updates
- w denotes the window size for innovation averaging
- $\alpha_{m,t}$ denotes the adaptive scaling factor for model m

- **Ensemble Method Notations:**
  - N_m denotes the ensemble size for model m
  - $\boldsymbol{\eta}_i \in \mathbb{R}^n$ denotes the perturbation vector for ensemble member i
  - $(\mathbf{x}^f_{m,t+1})_i \in \mathbb{R}^n$ denotes the state of i-th ensemble member
  - $\overline{\mathbf{x}}^f_{m,t+1} \in \mathbb{R}^n$ denotes the ensemble mean for model m
  - β denotes the scaling factor for ensemble sampling covariance
  - ϵ denotes the eigenvalue threshold for covariance conditioning

- **Vehicle Dynamics Notations:**
  - β denotes the vehicle sideslip angle
  - r denotes the yaw rate
  - ϕ denotes the roll angle
  - $\dot{\phi}$ denotes the roll rate
  - δ denotes the steering angle
  - u denotes the longitudinal velocity (Note: context distinguishes from control input)
  - $v = u\beta$ denotes the lateral velocity
  - v_x, v_y denote longitudinal and lateral velocities in body frame
  - α_f, α_r denote front and rear tire slip angles
  - $C_{\alpha f}, C_{\alpha r}$ denote front and rear tire cornering stiffnesses

- k_ϕ denotes the roll stiffness
- c_ϕ denotes the roll damping coefficient
- ϕ_r denotes the road bank angle
- θ_r denotes the road grade angle
- F_{yf}, F_{yr} denote front and rear lateral tire forces
- F_Y denotes the total lateral force
- m, m_s denote total and sprung mass (Note: context distinguishes from model index)
- M denotes vehicle mass (alternative notation)
- $I_z, I_{x,s}$ denote yaw and roll moments of inertia
- I_c denotes the composite roll moment of inertia
- l_f, l_r denote distances from CG to front and rear axles
- a, b denote alternative notation for l_f, l_r
- h_s denotes the CG height above roll center
- ω_{ij} denotes wheel speed, where $i \in \{f, r\}$, $j \in \{l, r\}$
- μ denotes the road friction coefficient

- **Special Notations and Operators:**
  - $\dot{x}$ denotes the time derivative of x
  - $\ddot{x}$ denotes the second time derivative of x
  - $\mathcal{R}(\cdot)$ denotes the range space of a matrix
  - $S \to 0$ denotes the limit as S approaches zero
  - $\forall$ denotes "for all"
  - $\exists$ denotes "there exists"
  - $\iff$ denotes "if and only if"
  - $\triangleq$ denotes "is defined as"
  - $\approx$ denotes "approximately equal to"
  - $O(\cdot)$ denotes the big-O notation for order of magnitude
  - $\otimes$ denotes the Kronecker product

- $\odot$ denotes the Khatri-Rao product (column-wise Kronecker)
- $\mathbb{E}[\cdot]$ denotes the expected value operator
- $\mathcal{L}(\cdot)$ denotes the likelihood function
- $p(\cdot|\cdot)$ denotes conditional probability density

- **Index Sets and Subscripts:**
  - i, j, k, l typically denote indices
  - n denotes the system dimension (state vector size)
  - h denotes the input dimension
  - k denotes the number of uncertain parameters
  - N denotes the number of identification models
  - M denotes the number of models in consensus fusion (Chapter 5)
  - m denotes model index in consensus framework
  - p as subscript denotes "plant"
  - p as dimension denotes measurement vector size
  - q denotes control input dimension
  - d denotes lifted space dimension in Koopman framework
  - f, r as subscripts denote "front" and "rear"
  - l, r as subscripts in wheel notation denote "left" and "right"
  - t denotes discrete time index
  - K as subscript denotes Koopman-related quantities

Chapter 1

Introduction

1.1 Motivation

Modern engineering systems operate in increasingly complex and uncertain environments, where traditional control approaches face fundamental limitations in ensuring robust and reliable performance. From autonomous vehicles navigating unpredictable road conditions to power systems managing renewable energy integration, the challenge of handling uncertainties in dynamic systems has become crucial. This challenge is particularly important in safety-critical applications where system failures can have catastrophic consequences, necessitating control strategies that can adapt to varying operating conditions while maintaining guaranteed performance bounds.

Traditional approaches to handling uncertainties, namely adaptive control and robust control, have provided valuable theoretical foundations and practical solutions for many applications. However, these classical methods can only handle systems within a certain limited range of uncertainties and often experience system-theoretic limitations in parameter identification, potentially leading to system instability. In practical scenarios, parametric uncertainties can increase substantially due to alterations in operational conditions, component aging or malfunction, and variations in plant dynamics resulting from external influences. These considerable parametric uncertainties can lead to suboptimal transient responses and even catastrophic system failures. Furthermore, conventional adaptive control methods demand consistently activated inputs to ensure accurate identification of unknown parameters—a requirement that proves highly limiting, especially in critical situations where safety is paramount and aggressive control inputs may be undesirable or dangerous.

To address these fundamental limitations, the multi-model approach has emerged as a powerful paradigm that employs multiple models representing different operating conditions or parameter values, combining their responses to control systems with variable parameters or uncertainties. Unlike traditional methods that rely on a single model with a wide range of uncertainty in its dynamic equation, this approach advocates the use of a set of models that represent the system's behavior under different operational conditions. This paradigm shift allows for a significantly less conservative approach to designing controllers, with one designated for each model within the system model set, thereby forming either supervised switched or blending control structures.

Despite the theoretical advantages of multi-model approaches, their practical implementation faces a critical challenge: the curse of dimensionality. As the dimension of the uncertain parameter space increases, the number of required identification models grows exponentially, rendering the algorithm computationally intensive and unsuitable for real-time applications. This computational burden becomes particularly crucial in modern applications such as integrated vehicle dynamics and power networks, where multiple subsystems interact through coupled dynamics and real-time performance is essential. Furthermore, the literature lacks clear strategies for model selection and optimal placement within the parameter uncertainty space, with most existing approaches relying on ad-hoc methods or exhaustive gridding that quickly becomes infeasible for high-dimensional systems.

The evolution of technology has introduced another dimension to this challenge: the rise of data-driven methods and machine learning techniques that offer powerful alternatives to traditional physics-based modeling. While physics-based models provide interpretability and theoretical guarantees, they often falter in extreme conditions where idealized assumptions break down. Conversely, data-driven approaches demonstrate remarkable capacity to learn complex patterns but struggle with generalization beyond their training data and lack the interpretability crucial for safety-critical applications. This divide represents more than a methodological preference—it reflects a deeper tension between theoretical understanding and empirical adaptation that must be resolved for next-generation control systems.

Vehicle dynamics control exemplifies these challenges in a concrete and impactful domain. Under harsh circumstances such as sudden changes in road surface friction and extreme driving conditions, a vehicle's response to driver commands can result in potential instability or significant deviations from the desired trajectory. The primary challenge in ensuring vehicle resilience to varying road conditions and complex tire behavior lies in accurate state estimation and parameter identification. Current estimation algorithms fall short in promptly and accurately detecting alterations in road conditions, particularly during transitions between different operating regimes. Given these considerations, there

is a critical need for system identification approaches that can promptly identify abrupt changes in vehicle dynamics using all available data while maintaining computational efficiency for real-time implementation.

1.2 Objectives

The primary objective of this thesis is to develop a comprehensive framework for multi-model adaptive control and estimation that addresses the fundamental challenges of computational complexity and model integration through three interconnected and systematic contributions, as illustrated in Figure 1.1. Each objective targets a specific aspect of the problem while collectively forming a unified solution for practical implementation of multi-model systems.

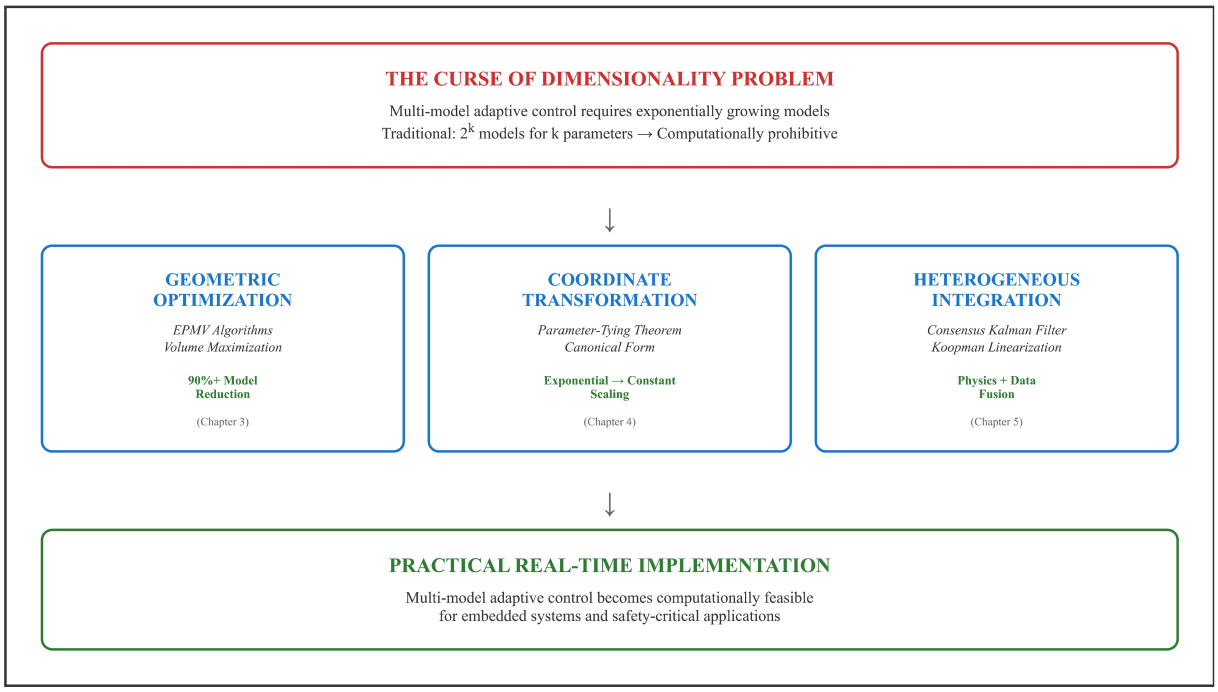


Figure 1.1: Overview of multi-model adaptive control challenges and the three-part solution approach: geometric optimization for model selection, coordinate transformations for dimension reduction, and consensus fusion for heterogeneous model integration.

1.2.1 Optimal Model Selection and Positioning

The first objective focuses on developing novel approaches for optimally selecting and positioning models within the parameter uncertainty space, maximizing coverage while minimizing computational requirements (Chapter 3). Traditional multi-model approaches suffer from the exponential growth in required models as system complexity increases, making real-time implementation computationally prohibitive. This thesis addresses this challenge by developing systematic methods based on computational geometry and optimization theory that can determine the minimal set of models required to adequately represent system uncertainties.

The approach leverages the key insight that traditional multi-model methods use far more models than necessary—instead of using every possible model combination to cover parameter uncertainties, carefully chosen subsets can achieve the same coverage with dramatically reduced computation. To realize this insight, this thesis introduces the Enclosed Polytope with Maximum Volume (EPMV) algorithms that find convex spaces with maximum volume inside the uncertainty space, ensuring optimal coverage while minimizing required models. These algorithms work strategically, like placing sensors to monitor an entire area with the fewest locations needed.

The framework extends beyond discrete model placement through an interval-based optimization method for continuous uncertainty intervals, jointly optimizing model placement and count through volume maximization. Additionally, a transfer function-based dimension reduction method enables efficient model selection for large-scale systems by emphasizing input-output characteristics rather than full state representations. Validation across vehicle lateral dynamics and power generator systems demonstrates that these methods typically reduce model requirements while maintaining estimation accuracy.

1.2.2 Dimension Reduction through Coordinate Transformations

The second objective establishes theoretical foundations for dimension reduction in multi-model systems through coordinate transformations and parameter dependencies (Chapter 4). While the first objective addresses model selection within a given parameter space, this objective tackles the fundamental scaling limitations by proving that changing to the right coordinates can considerably simplify high-dimensional uncertainty spaces. The challenge lies in developing rigorous mathematical foundations that guarantee the preservation of system properties while reducing computational complexity.

This thesis develops the Parameter-Tying Theorem as a theoretical innovation showing

that transformation to controllable canonical form, combined with analysis of monotonicity properties in how parameters affect the system, can reduce the number of required models from exponential in the number of parameters to potentially constant. The theorem provides conditions under which the effective dimension of the uncertainty space can be reduced without sacrificing coverage or performance guarantees. This represents a fundamental breakthrough in addressing the curse of dimensionality for multi-model systems.

The framework extends through multiple conditions including affine relationships, symmetry properties, and coordinated parameter variations, making it applicable beyond strictly monotonic systems. Weight transformation relationships are derived that maintain convex hull properties and stability guarantees across coordinate transformations, ensuring that dimension reduction does not compromise system performance. Validation on coupled lateral-roll vehicle dynamics with six uncertain parameters confirms the impact: sixty-four traditionally required models reduce to just two while fully covering the uncertainty space, demonstrating the dramatic computational savings possible through proper coordinate selection.

1.2.3 Integration of Physics-Based and Data-Driven Models

The third objective creates unified frameworks that seamlessly integrate physics-based and data-driven models, leveraging their complementary strengths while mitigating individual weaknesses (Chapter 5). Modern control systems require the interpretability and theoretical guarantees of physics-based models combined with the adaptability and pattern recognition capabilities of data-driven approaches. However, integrating these fundamentally different model types presents significant challenges in uncertainty quantification, covariance propagation, and fusion methodology.

This thesis develops a consensus multi-model Kalman filter framework that enables robust state estimation across diverse operating conditions by properly handling heterogeneous model types. The challenge of uncertainty propagation through nonlinear data-driven models is addressed through two novel methodologies: Koopman operator-based linearization that allows analytical covariance propagation for neural networks and other nonlinear models, and an ensemble-based approach that provides model-independent uncertainty quantification without requiring offline training.

The consensus fusion procedure automatically adjusts model weights based on their demonstrated reliability in current operating conditions, ensuring smooth transitions between models as system dynamics change. Innovation-based techniques enable online

estimation of model error covariances, ensuring proper calibration of relative uncertainties between different model types. Extensive experimental validation on an electric all-wheel-drive vehicle under extreme conditions—including high-slip maneuvers with sideslip angles over fifteen degrees, rapid transitions between ice and dry pavement, and sensor failures—demonstrates major performance improvements over traditional single-model approaches while maintaining reliable uncertainty bounds essential for safety-critical decisions.

Through these objectives, this thesis aims to bridge the gap between theoretical multi-model control concepts and practical implementation constraints, enabling the deployment of sophisticated adaptive control strategies in real-world applications where computational resources are limited and reliability is paramount.

1.3 Thesis Outline

This thesis is organized into seven chapters that systematically develop the theoretical foundations, algorithmic innovations, and practical applications of efficient multi-model approaches for uncertain dynamic systems.

Chapter 2: Background and Literature Review provides a comprehensive review of multi-model adaptive control approaches, tracing their evolution from early switching-based methods to modern hybrid frameworks. The chapter examines the mathematical foundations of multi-model systems, analyzes the computational challenges that limit practical implementation, and reviews recent advances in integrating physics-based and data-driven models. This review establishes the theoretical context and identifies the key gaps that motivate the contributions of this thesis.

Chapter 3: Optimal Multi-Model Selection via Geometric Analysis presents novel algorithms for model selection based on computational geometry principles. The chapter introduces two Enclosed Polytope with Maximum Volume (EPMV) algorithms that find convex spaces with maximum volume inside the uncertainty space, ensuring optimal coverage while minimizing required models. An interval-based optimization framework is developed for continuous uncertainty intervals, jointly optimizing model placement and count through volume maximization. A transfer function-based dimension reduction method enables efficient model selection for large-scale systems by emphasizing input-output characteristics. The effectiveness of the approach is shown in two distinct areas of dynamic systems, namely vehicle handling dynamics known as the bicycle model and power generator dynamics known as swing dynamics.

Chapter 4: Dimension Reduction through Parameter-Tying and Canonical Transformations develops the theoretical framework for reducing computational complexity through systematic transformation to canonical form. The chapter presents the parameter-tying theorem, derives weight transformation relationships that preserve system properties, and establishes verifiable criteria for ensuring robust performance in reduced-dimension spaces. The approach is validated through application to coupled lateral-roll vehicle dynamics.

Chapter 5: Consensus-Based Integration of Heterogeneous Models introduces a novel multi-model Kalman filter framework that seamlessly integrates physics-based and data-driven models. The chapter presents two distinct methodologies for handling covariance propagation in data-driven models: Koopman operator-based linearization and ensemble-based methods. The consensus fusion procedure and its iterative implementation are detailed, along with practical considerations for real-time deployment. The experimental results conducted on an electric all-wheel-drive Equinox vehicle demonstrate performance improvements over single-model techniques, with particularly significant advantages during challenging maneuvers and varying road conditions, confirming the effectiveness and robustness of the proposed methodology for safety-critical autonomous driving applications.

Chapter 6: Conclusions and Future Directions summarizes the key contributions of the thesis, discusses their theoretical and practical implications, and identifies promising directions for future research. The chapter reflects on how the developed methods advance the field of multi-model adaptive control and their potential impact on emerging applications in autonomous systems, smart infrastructure, and other domains requiring robust adaptation to uncertainty.

Throughout the thesis, theoretical developments are accompanied by rigorous mathematical analysis, algorithmic implementations are provided with consideration for practical constraints, and experimental validations demonstrate real-world applicability. The progression from foundational concepts to advanced integration techniques reflects the comprehensive nature of this research, which bridges theoretical innovation with practical implementation to advance the state of the art in adaptive control for uncertain dynamic systems.

Chapter 2

Background and Literature Review

This chapter provides a comprehensive review of multi-model approaches for handling uncertainties in dynamic systems, tracing the evolution from traditional adaptive control methods to modern hybrid frameworks that integrate physics-based and data-driven models. The review examines the theoretical foundations, computational challenges, and recent advances in multi-model adaptive control, with particular emphasis on applications to vehicle dynamics and power systems.

2.1 Multi-Model Adaptive Control: Foundations and Evolution

2.1.1 Motivation and Traditional Approaches

Handling uncertainties in system modeling remains one of the most critical challenges in dynamic systems control. Traditional approaches such as adaptive control and robust control have been extensively studied and applied, yet they possess inherent limitations when dealing with systems exhibiting broader uncertainties [1]. These classical methods can only handle systems within a certain limited range of uncertainties, often experiencing system-theoretic limitations for parameter identification and potential system instability [2, 3].

Practical scenarios frequently witness substantial increases in parametric uncertainties due to alterations in operational conditions, component aging or malfunction, and variations in plant dynamics resulting from external influences. These considerable parametric

uncertainties can lead to suboptimal transient responses and even system instability. Furthermore, conventional adaptive control methods demand consistently activated inputs to ensure accurate identification of unknown parameters, a requirement that proves highly limiting, especially in critical situations where safety is crucial [2].

To tackle these challenges associated with broader system variations and uncertainties, Multi-Model Adaptive Control (MMAC) strategies have been developed [4]. The fundamental concept behind utilizing multi-models in control design is to offer a more comprehensive description of the behavior and control performance of time-varying systems affected by uncertainties. Instead of relying on a single model with a wide range of uncertainty in its dynamic equation, this approach advocates the use of a set of models that represent the system's behavior under different operational conditions [5]. This paradigm shift allows for a significantly less conservative approach to designing a set of controllers, with one designated for each model within the system model set, thereby forming either a supervised switched [6, 7, 8] or blending [9, 10, 11] control structure.

The theoretical foundations of multi-model control have been extensively developed in the literature. [12] presented a comprehensive analysis of the new approach to adaptive control using multiple models, while [13] explored hierarchical adaptive control strategies for rapidly time-varying systems. The rationale for second-level adaptation was further elucidated in [14], providing theoretical justification for the benefits of multi-model approaches over traditional single-model methods.

2.1.2 Switching-Based Multi-Model Approaches

The development of switching-based multi-model control represents one of the earliest systematic approaches to handling multiple operating regimes. As early illustrations, multiple-model Kalman filters were introduced in [15] to control lateral and longitudinal dynamics of an F-8C aircraft, and [16] presented a framework to enhance the accuracy of state estimation through partitioning approaches. The concept of switching was formally introduced in [17] to provide necessary and sufficient conditions for adaptive stabilization.

Two primary switching strategies emerged from this early work: direct switching and indirect switching. In direct switching [18, 19], the system's output dictates when the switch to the next controller occurs, following a predetermined sequence. This approach offers simplicity in implementation but lacks flexibility in adapting to unexpected system behaviors. In contrast, indirect switching determines both the timing of switches and which controller is selected [20], providing greater adaptability at the cost of increased complexity.

In the context of uncertain linear SISO systems, researchers [21, 22, 23] explored set-point regulation and introduced a high-level controller known as a supervisor. This supervisor oversees the switching between a set of candidate set-point controllers, ensuring that the system output closely follows a constant reference input. These studies assumed that the uncertain system consists of various sub-families, each having its own linear controller for achieving set-point regulation. The selection criterion typically involved output-squared estimation errors for comparison, with the candidate controller exhibiting the smallest performance index being selected to generate the control input.

Building upon these foundations, [24] addressed continuous-time linear SISO systems with significant modeling errors, employing a family of linear candidate controllers under the guidance of high-level switching logic. Their primary focus was on the switching mechanisms found in the existing literature, highlighting the modularity aspect of switching-based multiple-model adaptive control. This modularity, where the switching logic, multiple-model system identifier, and candidate controllers can be designed independently, enhances flexibility in various applications.

The advantages of switching-based control schemes were comprehensively analyzed in [25], which compared them with traditional adaptive control techniques. These advantages include simplified parameterization, reduced computational cost, the ability to handle non-convex or discrete parameter sets, and improved modularity. The paper categorized switching logic into two groups: one relying on process estimation (using either certainty equivalence or model validation) and the other involving direct evaluation of candidate controllers.

The concept of switching and tuning within multi-model adaptive control was introduced and enhanced to prevent suboptimal responses caused by transitions between fixed controllers [7, 26]. The central concept involved utilizing switching logic for rapid adaptation and subsequently adjusting the controller's gain through an adaptive law to enhance performance. This switching and tuning approach extended the applicability of multi-model methods from uncertain LTI systems to uncertain Linear Time-Varying (LTV) systems. Further developments in adaptive control using multiple models with switching and tuning were presented in [8], demonstrating improved transient response characteristics.

2.1.3 Blending-Based Multi-Model Structures

While switching-based methods offered significant improvements over traditional adaptive control, they faced two fundamental challenges in practical implementation. First, because one of the predetermined models needs to be sufficiently close to the original system's

parameters at any given time t, the number of these fixed models, denoted as N, can become large and increases exponentially as the unknown parameter vector's dimension grows [27]. Second, the models other than the best-matching one don't contribute to the estimation of unknown parameters, leading to inefficient use of available information.

To address these limitations, [5] introduced a technique called "second-level adaptation" for identifying the plant using a set of fixed identification models, each associated with a fixed controller gain. The resulting control gain is determined by blending these fixed control gains based on the identification errors. This method combines information from all fixed models to estimate plant parameters while using a minimal number of models. The approach was further developed for uncertain LTI SISO systems in companion form using a set of adaptive identification models, constructed using the indirect model reference adaptive control scheme.

Multiple model adaptive control with mixing was comprehensively studied by [9], demonstrating the advantages of blending approaches over pure switching methods. The theoretical foundation for second-level adaptation with fixed identification models was established through the outcomes derived from multiple adaptive identification models. Subsequently, this method was expanded to control uncertain LTV SISO systems in companion form, utilizing fixed identification models. The extension to LTV systems required treating the weight vector W as a time-varying parameter $W(t)$, representing the time-dependent changes in the parameter vector $A_p(t)$.

[28] explored the application of multiple models and switching in online center of gravity estimation for automotive vehicles, demonstrating the practical benefits of blending-based approaches. Recent applications of blending-based approaches have demonstrated their effectiveness in various domains. [10] utilized blending of multiple models to analyze the stability of vehicle motion, while [11] applied similar concepts to lateral vehicle dynamics. These studies showed that blending-based methods could achieve superior performance compared to switching approaches, particularly in situations where smooth transitions between operating conditions are required.

2.1.4 Applications in Vehicle Dynamics and Power Systems

The utilization of multiple-model system identification approaches has gained significant attention in vehicle dynamics estimation. This method stands out because it necessitates less sensor data and does not rely on persistently excited inputs for identification purposes. In [29], a switching-based multiple-model system identifier was employed for real-time estimation of constant parameters in vehicle dynamics. The study focused on using a

2-DOF linear lateral vehicle dynamics model, including sideslip angle and yaw-rate, to determine the cornering stiffness of both front and rear tires, as well as the longitudinal position of the vehicle's center of gravity (CG).

These parameters were assumed to exist within a continuous parameter space, with a total of 140 identification models designed to comprehensively cover this space. The results demonstrated that all three constant parameters were accurately estimated. The same approach was extended to a 1-DOF linear roll vehicle dynamics model to identify uncertain spring stiffness, damping coefficient, and the vertical position of the vehicle's CG. Similar to the lateral dynamics case, the uncertain parameters were constrained within a known parameter space, with 240 models designed for their identification.

Vehicle dynamics control applications have been extensively studied using multi-model approaches. [30] developed an optimal yaw moment control law for improved vehicle handling, demonstrating the effectiveness of multi-model techniques in complex control scenarios. [31] extended these concepts to develop an advanced torque vectoring control system for electric vehicles with in-wheel motors using soft computing techniques, showcasing the versatility of multi-model approaches in modern vehicle architectures. [32] further refined optimal yaw moment control strategies, contributing to the growing body of knowledge on multi-model vehicle dynamics control.

The application of multi-model approaches extends beyond vehicle dynamics to power systems, where they have been employed for handling uncertainties in generator dynamics and network stability analysis [33, 34]. These applications demonstrate the versatility of multi-model methods across different engineering domains, highlighting their potential for addressing complex system uncertainties in various contexts. The system-theoretic performance metrics developed for low-inertia stability of power networks [33] and the $H\infty$ performance analysis of mechanical and power networks [34] provide valuable insights into the broader applicability of multi-model techniques.

The mathematical rigor underlying these applications is supported by fundamental texts such as [35] for mathematical analysis and [36] for matrix analysis techniques essential to multi-model system theory. These theoretical foundations ensure the robustness and reliability of multi-model approaches across diverse application domains.

2.2 Computational Efficiency and Model Selection in Multi-Model Systems

2.2.1 The Curse of Dimensionality Challenge

Despite the theoretical advantages of multi-model approaches, their practical implementation faces a significant computational challenge known as the "curse of dimensionality." As the dimension of the uncertain parameter space increases, the number of required identification models grows exponentially [4]. This exponential growth renders the algorithm computationally intensive and unsuitable for real-time applications, particularly in high-dimensional systems such as integrated vehicle dynamics [37] or complex power networks.

The computational burden becomes particularly acute when considering modern applications that require real-time performance. For instance, in autonomous vehicle control, state estimation must be performed at high frequencies (typically 100-1000 Hz) to ensure safe operation. When the number of models grows exponentially with system dimension, even efficient implementations struggle to meet these real-time constraints.

Furthermore, the literature has not adequately explored systematic criteria for selecting models within the multi-model framework. Most existing approaches rely on ad-hoc methods or exhaustive gridding of the parameter space, which quickly becomes infeasible for high-dimensional systems. This gap in the literature motivated the development of novel approaches for optimal model selection that can maintain coverage of the uncertainty space while dramatically reducing computational requirements.

Advanced state estimation techniques have been developed to address computational challenges. [38] presented an unscented Kalman filter for vehicle state estimation that offers improved computational efficiency compared to traditional approaches. [39] developed specialized methods for state estimation in roll dynamics for commercial vehicles, demonstrating the need for computationally efficient approaches in practical applications.

2.2.2 Geometric Approaches to Model Selection

To address the computational challenges, novel geometric approaches have been developed for optimal model selection within multi-model frameworks. These approaches leverage tools from computational geometry and optimization to systematically select a subset of models that maximizes coverage of the parameter uncertainty space.

The theoretical foundations for these geometric approaches draw from extensive research in computational geometry. [40] developed approximation algorithms for convex hulls that provide efficient methods for handling high-dimensional spaces. [41] presented comprehensive experimental evaluations of convex-hull algorithms, offering insights into practical implementation considerations. [42] specifically addressed the challenge of computing approximate convex hulls in high dimensions, while [43] provided a thorough treatment of convex hull computations in the broader context of discrete and computational geometry. The work of [44] on optimized bounding boxes of polytopes provides additional tools for efficient model selection in multi-dimensional parameter spaces.

2.3 Integration of Physics-Based and Data-Driven Approaches

2.3.1 The Emergence of Hybrid Modeling Paradigms

The divide between physics-based and data-driven approaches goes beyond just choosing a method—it reflects a fundamental struggle between theoretical understanding and learning from data that runs through all areas of engineering. Traditional physics-based models are transparent and built on solid theory, but they often fail in extreme conditions where their idealized assumptions no longer hold. On the other hand, data-driven approaches can learn incredibly complex patterns, but they struggle to work beyond the data they were trained on and often can't explain their decisions—a serious problem for safety-critical systems.

Early work in hybrid modeling came from [45], who demonstrated one of the first applications to vehicle dynamics by combining a physical vehicle model with a Local Linear Model Tree (LOLIMOT) neural network in a serial Wiener structure. Their approach showed significant error reduction when adapting to changing road conditions, establishing the potential benefits of combining physical understanding with data-driven flexibility.

Recent advances in data-driven methods have revolutionized vehicle dynamics modeling. [46] developed a data-driven approach using deep neural networks to model vehicle longitudinal dynamics based on multibody simulation data, demonstrating high prediction accuracy and computational efficiency for real-time applications. [47] showed that neural networks with time-delayed inputs can effectively estimate vehicle sideslip angle without requiring a reference model or parameter identification, achieving errors generally below 1° across various driving conditions.

Specialized neural network architectures have shown particularly promising results. [48] demonstrated that General Regression Neural Networks (GRNN) can effectively estimate vehicle sideslip angle by modeling it as a time series mapping of yaw speed and lateral acceleration. [49] developed a multi-model approach using specialized NARX neural networks for different road conditions, selected by a classification network, achieving 98.6% accuracy in sideslip angle estimation.

Comparative studies have provided valuable insights into the relative merits of different approaches. [50] conducted a comprehensive comparison of physical and data-driven models for longitudinal vehicle dynamics under large-scale real-world driving conditions, demonstrating that data-driven approaches can outperform traditional physical models while requiring less computational effort for training. [51] showed that deep neural networks, particularly Convolutional Neural Networks (CNNs), can effectively learn and control coupled longitudinal and lateral vehicle dynamics, outperforming conventional decoupled controllers in challenging scenarios.

The modeling of vehicle dynamics with neural networks has been further advanced by [52], who demonstrated that neural networks with physics-inspired pre-wired structures outperform conventional black-box networks. Recent studies such as [53] have shown that recurrent neural networks can effectively estimate vehicle side-slip angle across varying road surfaces without explicit friction information, outperforming traditional filtering approaches in both accuracy (0.195° vs. 0.297° median RMSE) and computational efficiency (100× faster).

2.3.2 Hybrid Physics-Data Driven Modeling Frameworks

Hybrid physics-data driven modeling has emerged as a promising direction that maintains the interpretability and theoretical foundation of physics-based models while exploiting the capacity of data-driven methods to capture complex, nonlinear behaviors. [54] proposed a hybrid physics-data-driven framework integrating physical models with online learning neural networks for vehicle dynamics modeling, demonstrating that an H_∞-based learning algorithm outperforms traditional EKF approaches in convergence speed and robustness to uncertain noise characteristics.

[55] proposed a hybrid physics-data driven observer that integrates ridge regression with a dual-attention LSTM network for vehicle dynamics state estimation, demonstrating superior accuracy over traditional UKF observers and purely data-driven methods across various driving conditions. Their approach eliminates the need for differential forms of physics models while effectively capturing nonlinear vehicle dynamics.

The effectiveness of hybrid approaches has been demonstrated across various aspects of vehicle dynamics. [56] developed a hybrid modeling approach for vehicle lateral dynamics that combines a physical model with a dual-attention GRU neural network for modeling residuals, demonstrating superior performance under limited data conditions. [57] proposed a hybrid lateral dynamics model that uses neural networks to predict time-varying parameters for a bicycle model, demonstrating improved accuracy over conventional models in nonlinear regimes while maintaining physical interpretability.

The integration of physical knowledge into neural networks has shown significant benefits. [58] demonstrated that injecting physical knowledge into intermediate layers of neural networks significantly improved prediction accuracy and reduced uncertainty for various nonlinear dynamical systems. This physics-guided approach represents a promising direction for developing more robust and interpretable hybrid models.

A critical distinction in hybrid approaches is between those capable of online adaptation versus those relying on offline training. [59] introduced a hybrid architecture that embeds neural networks into physical vehicle dynamics models to characterize modeling errors and identify time-varying parameters, though their approach requires extensive offline training and high-precision sensors. [60] demonstrated the effectiveness of a hybrid modeling approach that integrates physics-based nominal models with data-driven Koopman models for automated vehicle control under extreme conditions, providing a foundational framework for multi-model integration.

Beyond basic dynamics modeling, hybrid approaches have been extended to specialized aspects of vehicle behavior. [61] demonstrated that hybrid residual deep learning models significantly outperform both pure physics-based and pure deep learning approaches in plant transpiration estimation, with their HPDRL model showing superior generalization capabilities particularly in limited data scenarios.

For specialized vehicle subsystem estimation, several targeted hybrid approaches have shown promising results. [62] developed a hybrid modeling framework combining physics-based vehicle models with neural networks to discover tire behaviors without expensive instrumentation. [63] demonstrated a hybrid approach using piecewise affine identification to approximate nonlinear tire dynamics for model predictive control.

For vehicle mass estimation, [64] demonstrated that fusing machine learning with physics-based vehicle dynamics through fuzzy logic can achieve 97% accuracy in vehicle mass estimation under varying conditions. [65] demonstrated a dual Kalman filter approach for simultaneous estimation of vehicle mass and road irregularities, achieving errors below 5% in simulations.

For tire-road friction estimation, [66] developed a data-driven method combining time-domain and time-frequency features through a stochastic variational deep kernel learning framework, achieving significant accuracy improvements especially under small excitation conditions where traditional methods fail.

2.3.3 Multi-Model Kalman Filtering with Heterogeneous Models

Kalman filtering techniques have become the cornerstone of vehicle state estimation due to their ability to optimally combine model predictions with sensor measurements while accounting for uncertainties in both. An adaptive IMU/GNSS data fusion algorithm that enhances vehicle dynamic models through complementary terms derived from sensor measurements is developed in [67], demonstrating improved estimation accuracy despite sensor biases and model uncertainties.

A hierarchical vehicle dynamics state estimation approach combining adaptive square-root cubature Kalman filtering with a partitioned similarity-principle algorithm is proposed in [68], demonstrating excellent accuracy in estimating vehicle states and tire-road friction. The challenge of varying operating conditions has motivated adaptive filtering approaches. A Variable Structure Extended Kalman Filter that effectively combines kinematic and dynamic vehicle models through feedback mechanisms and damping factors is developed in [69].

An Adaptive Fault-Tolerant Extended Kalman Filter (AFTEKF) for vehicle state estimation that effectively handles partial sensor data loss by modeling it as a discrete distribution is developed in [70]. Recent advancements such as an innovative adaptive unscented Kalman filter that dynamically adjusts process and measurement noise covariances based on vehicle-dynamics-derived heuristics are presented in [71], achieving remarkable improvements in state estimation accuracy during extreme driving conditions.

The integration of neural networks with Kalman filtering techniques has shown particular promise. An innovative approach combining deep ensemble neural networks with Kalman filters for vehicle sideslip angle estimation, where the neural network provides both state estimates and uncertainty values to adaptively adjust the Kalman filter's measurement covariance matrix, is proposed in [72]. A mutualistic hybrid approach integrating a CNN with a UKF through end-to-end training is proposed in [73], demonstrating superior vehicle sideslip angle estimation particularly in highly non-linear driving conditions.

Research demonstrates that combining an Adaptive Neuro-Fuzzy Inference System (ANFIS) with an Unscented Kalman Filter (UKF) produced superior vehicle sideslip angle estimates compared to either technique alone in [74]. A neural-network-based Takagi-

Sugeno fuzzy reduced-order observer for vehicle dynamics that uses a sliding mode observer to identify model uncertainties from training data is developed in [75].

Work in [76] presents an innovative approach integrating artificial neural networks with unscented Kalman filtering for vehicle sideslip angle estimation using only IMU measurements, demonstrating significant improvements over single-model approaches while providing transparency to vehicle model uncertainties.

Multi-model Kalman filtering approaches have demonstrated significant advantages. Recent advances include an innovative state-dependent Interacting Multiple Model approach for vehicle sideslip angle estimation that adapts transition probabilities between different road condition models in real-time, as developed in [77]. An Interacting Multiple Model based on Square Root Cubature Kalman Filter (IMM-SCKF) for vehicle state estimation that effectively addressed noise uncertainty is proposed in [78].

Research in [79] shows that an Interacting Multiple Model Kalman filter with distinct vehicle dynamic models representing different road surface conditions significantly improves lateral motion estimation compared to single-model approaches. An Interacting Multiple Model (IMM) approach for simultaneously estimating vehicle side-slip angle and road friction coefficient using multiple tire models and extended Kalman filters is developed in [80].

An Interacting Multiple Model Kalman filter that optimally combines kinematic and dynamic vehicle models for sideslip angle estimation using low-cost sensor fusion is developed in [81]. A multi-physical model fusion framework using Interacting Multiple Models and time-series analysis for vehicle sideslip trajectory prediction is developed in [82].

2.3.4 Uncertainty Quantification and Fault Tolerance

A critical aspect of vehicle state estimation systems, particularly for safety-critical autonomous driving applications, is their resilience to sensor failures and data corruption. Work in [83] develops a cyber-physical vehicle velocity estimation framework that combines LPV observers with distributed consensus algorithms in a fault-tolerant architecture. A hybrid model-data fault detection system that integrates model-based estimators with data-driven approaches is developed in [84], demonstrating superior performance for vehicle sensor and actuator fault diagnosis.

The importance of uncertainty quantification in vehicle dynamics estimation is highlighted by a comprehensive comparison of model-based and data-driven approaches for vehicle sideslip angle estimation conducted in [85]. Their study demonstrated that tire

force measurements significantly improve estimation accuracy for both methods, while highlighting the complementary strengths of each approach.

Recent work demonstrates that deep ensembles provide well-calibrated uncertainty estimates that outperform Bayesian approaches while remaining simple to implement and scalable to large datasets, as shown in [86]. This approach to uncertainty quantification is particularly relevant for integrating data-driven models into safety-critical estimation frameworks.

A comprehensive analysis of hybrid-driven state estimation techniques, categorizing them into three primary architectural approaches: (1) Network-based correction of model estimates, (2) Learning of system parameters, and (3) Direct state estimation, is provided in [87]. This categorization provides a valuable framework for understanding different integration strategies.

Research in [88] develops a distributed cyber-physical system framework that integrates nonlinear autoregressive neural networks for behavior prediction with vehicle dynamic models for cruise control, demonstrating improved safety through a novel range policy that considers cut-in probability.

Effective validation methodologies remain critical. A comprehensive framework for vehicle dynamics model validation that includes parameter estimation during dynamic testing and iterative validation processes is established in [89], providing important context for evaluating the robustness of different modeling approaches.

Chapter 3

Optimal Multi-Model Selection via Geometric Analysis

3.1 Introduction

The multi-model approach has emerged as a powerful methodology for handling uncertainties in dynamic systems, as extensively discussed in Chapter 2. By employing multiple models that each represent different operating conditions or parameter values, this approach provides a comprehensive framework for controlling systems with significant uncertainties. However, despite its theoretical advantages, the practical implementation of multi-model control faces a critical challenge: the exponential growth in the number of identification models as the dimension of the uncertain parameter space increases.

This "curse of dimensionality" renders the algorithm computationally intensive and unsuitable for real-time applications. In practical scenarios, especially those requiring high-frequency control updates such as autonomous vehicle systems or power grid management, the computational burden of processing hundreds or thousands of models becomes prohibitive. Despite considerable efforts in the literature on multi-model approaches, there remains an evident lack of clear strategies for systematic model selection over all possible models of the system.

This chapter addresses this fundamental challenge by introducing novel approaches based on geometric analysis of the parameter uncertainty space. The key insight driving our methodology is that not all models contribute equally to the coverage of the uncertainty space—many models may be redundant or provide minimal additional information. By

leveraging principles from computational geometry and optimization theory, we develop systematic methods to identify the minimal set of models that adequately represents the system's behavior across all operating conditions.

The chapter makes three primary contributions:

1. **Geometric Model Selection Algorithms**: We introduce novel algorithms to select a subset of models by evaluating a metric in the system's parameter space. This metric quantifies the proximity of the chosen models to the convex hull encompassing all possible system models. Two specific algorithms, termed Enclosed convex Polytope with Minimum Models (EPMM), are developed with different selection strategies to accommodate various model distributions.

2. **Model Positioning under Interval Uncertainty**: We present a new approach for optimally positioning models within a parameter space defined by uncertainty intervals. This method finds a new set of models that maximally cover the space, which is particularly beneficial when a pre-defined constraint limits the number of selectable models. The approach treats both model count and positions as optimization variables.

3. **Transfer Function-Based Dimension Reduction**: We develop a transfer function-based method to simplify model selection in large-scale systems with numerous states. This method facilitates the transition from the high-dimensional parameter space to a lower-dimensional space characterized by transfer function gains, significantly reducing computational complexity while preserving essential system characteristics.

The effectiveness of these approaches is demonstrated through two distinct applications that represent different classes of dynamic systems: vehicle lateral dynamics (representing mechanical systems with parameter uncertainties) and power generator swing dynamics (representing electrical systems with network uncertainties). These applications showcase the versatility and practical impact of our methods across different engineering domains.

3.2 Problem Formulation and Mathematical Preliminaries

3.2.1 System Description and Uncertainty Characterization

Consider an uncertain linear system described in state-space form:

$$\dot{x}_p(t) = A_p(\theta)x_p(t) + B_p(\theta)d(t), \tag{3.1}$$

where:

- $x_p(t) \in \mathbb{R}^n$ represents the state vector of the plant
- $d(t) \in \mathbb{R}^m$ denotes the input vector
- $A_p(\theta) \in \mathbb{R}^{n\times n}$ is the state matrix, which depends on an uncertain parameter vector θ
- $B_p(\theta) \in \mathbb{R}^{n\times m}$ is the input matrix, also dependent on θ
- $\theta \in \mathbb{R}^q$ is the uncertain system parameter vector

The uncertainty in the parameter vector θ arises from various sources in practical applications. In vehicle dynamics, these uncertainties may stem from variations in vehicle mass due to different loading conditions, changes in tire characteristics due to wear or road conditions, or variations in suspension parameters. In power systems, uncertainties arise from varying network impedances, generator inertias, and load characteristics.

To make the problem tractable, we introduce the following fundamental assumption:

Assumption 3.1 (Convex Hull Representation)**.** There exist N known system matrix pairs $A_i \in \mathbb{R}^{n\times n}$ and $B_i \in \mathbb{R}^{n\times m}$, $i = 1, \ldots, N$, for some known integer $N > 0$, such that:

$$[A_p \ B_p] \in \mathrm{Conv}\left\{[A_i \ B_i] : i = 1, \ldots, N\right\}, \tag{3.2}$$

where $\mathrm{Conv}\{\cdot\}$ denotes the convex hull of a set of matrices.

This assumption implies that the actual system matrices can be expressed as a convex combination of known vertex matrices. This is a reasonable assumption in many practical scenarios where the system parameters vary within known bounds, and the dependency of the system matrices on these parameters is continuous.

3.2.2 Multi-Model Framework

In the multi-model scheme, the uncertain linear system of (3.1) is represented by a collection of models, each corresponding to a specific value of the uncertain parameters. For each vertex $i \in \{1, \ldots, N\}$, we construct a fixed model:

$$\dot{x}_i(t) = A_i x_i(t) + B_i d(t). \tag{3.3}$$

Each model generates a distinct output for identical inputs, reflecting the plant's potential response under different parameter values. This multiplicity of responses forms the basis for robust estimation and control. The output produced by each identification model is denoted as $y_i(t)$. Under the assumption that the states of the unknown plant are fully accessible (an assumption that can be relaxed using observers), we have $y_p(t) = x_p(t)$.

The identification error for each model is defined as:

$$e_i(t) = x_i(t) - x_p(t). \tag{3.4}$$

This error signal plays a crucial role in determining which models best represent the current plant behavior and in computing appropriate weights for model blending.

3.2.3 Theoretical Foundation for Multi-Model Inclusion

The following theorem, originally established in the multi-model control literature, provides the theoretical foundation for our approach:

Theorem 3.2 (Multi-Model Inclusion Property [5])**.** *If the plant parameter A_p lies in the convex hull of N known identification models $\{A_1, A_2, \ldots, A_N\}$ at time t_0, and if $e_i(t_0) = 0$ for all i, then for all $t \geq t_0$:*

$$A_p(t) = \sum_{i=1}^{N} w_i(t) A_i, \quad \sum_{i=1}^{N} w_i(t) e_i(t) = 0, \tag{3.5}$$

where $w_i > 0$ and $\sum_{i=1}^{N} w_i = 1$.

This theorem establishes that if the plant starts within the convex hull of the identification models, it remains within this convex hull for all future times. Moreover, the weights

that express the plant as a convex combination of the models can be determined from the identification errors.

The significance of this theorem is to guarantee that a properly selected set of models can represent the plant behavior throughout its operation, provided the initial selection adequately covers the uncertainty space. This motivates our focus on optimal model selection strategies.

3.2.4 Extension to Non-linear Uncertain Plants

While the multi-model framework presented thus far assumes a linear uncertain plant, a significant advantage of this approach lies in its ability to handle non-linear systems through time-varying convex combinations of fixed linear models. This section elucidates how the proposed methodology can effectively represent and control non-linear uncertain plants without explicit non-linear modeling.

Local Linearization Through Time-Varying Weights

Consider a non-linear uncertain plant described by:

$$\dot{x}_p(t) = f(x_p(t), d(t), \theta(t)), \tag{3.6}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^q \to \mathbb{R}^n$ is a non-linear function of the state, input, and time-varying uncertain parameters.

The key insight of our approach is that while the identification models $\{A_i, B_i\}_{i=1}^N$ remain fixed and linear, the effective plant representation at each time instant is given by:

$$A_p(t) = \sum_{i=1}^{N} w_i(t) A_i, \quad B_p(t) = \sum_{i=1}^{N} w_i(t) B_i, \tag{3.7}$$

where the time-varying weights $w_i(t)$ are not fixed but evolve continuously to track the non-linear behavior. This formulation effectively creates a piecewise linear approximation of the non-linear system, where between time instants t and $t + \Delta t$, the matrix pair $[A_p(t), B_p(t)]$ represents a local linearization that "shadows" and mimics the non-linear dynamics. Figure 3.1 illustrates this concept, showing how multiple fixed linear identification models contribute through time-varying weights to construct a representation of the non-linear uncertain plant.

Sliding Window Weight Optimization

The time-varying weights $w_i(t)$ are determined through a sliding window least squares optimization problem that minimizes the identification error over a recent time horizon. Specifically, at each time t, we solve:

$$\begin{aligned} \min_{w_i(t)} \quad & \int_{t-T_w}^{t} \left\| \sum_{i=1}^{N} w_i(\tau) e_i(\tau) \right\|^2 d\tau \\ \text{subject to} \quad & \sum_{i=1}^{N} w_i(t) = 1, \\ & w_i(t) \geq 0, \quad i = 1, \ldots, N, \end{aligned} \tag{3.8}$$

where T_w is the sliding window length, chosen to balance between tracking speed and noise rejection.

This optimization problem ensures that the weights adapt to the changing dynamics of the non-linear plant while maintaining the convexity constraints. The sliding window approach provides several advantages:

- **Adaptability**: The weights can track slow variations in the non-linear dynamics
- **Robustness**: The windowed averaging provides inherent filtering of measurement noise
- **Computational Efficiency**: The optimization problem remains convex and can be solved efficiently in real-time

Theoretical Justification

The ability of this framework to represent non-linear systems can be understood through the lens of universal approximation. As the number of identification models N increases and their distribution in the parameter space becomes denser, the time-varying convex combination can approximate any smooth non-linear function to arbitrary accuracy over bounded regions of the state space.

Remark 3.3 (Non-linear Approximation Capability). For a Lipschitz continuous non-linear function f and any $\epsilon > 0$, there exists a finite set of linear models $\{A_i, B_i\}_{i=1}^{N}$ such that

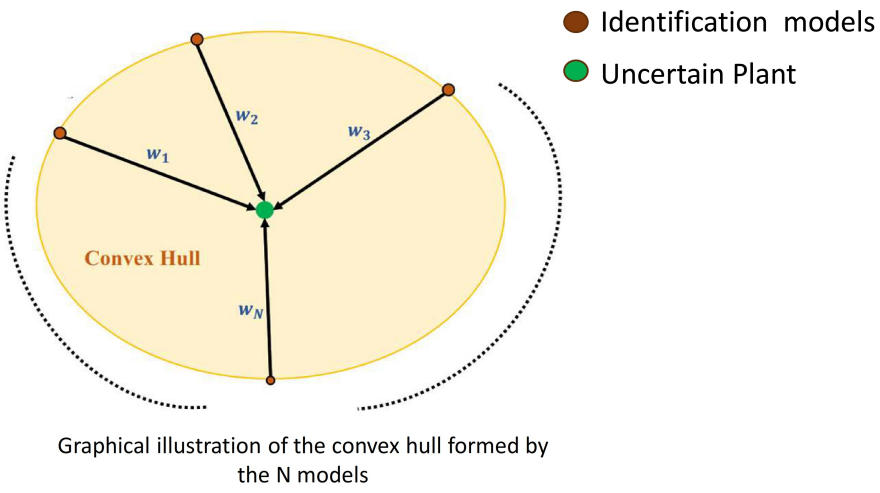


Figure 3.1: Representation of non-linear uncertain plant through time-varying weighted combination of fixed linear identification models. The weights $w_i(t)$ evolve through time via sliding window least squares optimization, enabling the linear models to collectively track and represent the non-linear dynamics.

the time-varying convex combination with optimally chosen weights $w_i(t)$ satisfies:

$$\left\| f(x_p(t), d(t), \theta(t)) - \left(\sum_{i=1}^{N} w_i(t) A_i \right) x_p(t) - \left(\sum_{i=1}^{N} w_i(t) B_i \right) d(t) \right\| < \epsilon, \tag{3.9}$$

for all $x_p(t)$ in a bounded region and all admissible inputs $d(t)$.

This property ensures that the multi-model framework can effectively handle non-linear uncertain plants without requiring explicit non-linear model structures, maintaining the computational advantages of linear model-based control while achieving the flexibility needed for non-linear systems.

3.2.5 Extension to Input Matrices

For the input matrix B_p, we need to establish conditions under which it can be represented within the model set. By assuming that A_p is non-singular (a reasonable assumption for stable systems) and has full rank, its range space is $\mathbb{R}^n$. Therefore, any vector in $\mathbb{R}^n$, including each column of B_p, must lie in the range space of A_p.

For each column vector $b_{p,j}$ of B_p, there exists a corresponding vector $x_j \in \mathbb{R}^n$ such that:

$$b_{p,j} = A_p x_j, \quad \forall j \in \{1, \ldots, m\}. \tag{3.10}$$

Since the range space $\mathcal{R}(A_p)$ is spanned by the columns of the convex combination of $A_1, A_2, \ldots, A_N$, it follows that:

$$B_p \in \mathcal{R}(A_1, A_2, \ldots, A_N). \tag{3.11}$$

This relationship ensures that the input characteristics of the plant can be captured by the selected models, which is essential for accurate system identification and control.

3.2.6 Weight Determination and Error Dynamics

The relationship in Theorem 3.2 can be expressed in matrix form:

$$\begin{aligned} &E(t)W(t) = 0, \\ &E(t) = \begin{bmatrix} e_1(t) & e_2(t) & \ldots & e_N(t) \end{bmatrix}, \\ &W(t) = \begin{bmatrix} w_1 & w_2 & \ldots & w_N \end{bmatrix}^T. \end{aligned} \tag{3.12}$$

In practice, due to measurement noise and modeling approximations, the constraint $E(t)W(t) = 0$ may not be exactly satisfied. Therefore, we employ a moving-window least squares approach to find weights that minimize $\|E(t)W(t)\|^2$ subject to the constraints $w_i \geq 0$ and $\sum_{i=1}^{N} w_i = 1$. This formulation ensures robustness to noise while maintaining the convex combination property.

3.2.7 Parameter Uncertainty Space

To formalize our geometric approach, we need to define the space in which model selection occurs:

Definition 3.4 (Parameter Uncertainty Space)**.** Let the vector $\phi \in \mathbb{R}^{n^2}$ be defined as the vectorization of the uncertain elements in matrix $A_p(\theta)$:

$$\phi = \begin{bmatrix} a_{11}(\theta) & a_{12}(\theta) & \cdots & a_{1n}(\theta) & a_{21}(\theta) & \cdots & a_{nn}(\theta) \end{bmatrix}^T, \tag{3.13}$$

where each $a_{ij}(\theta)$ represents a potentially uncertain element of $A_p(\theta)$. The *Parameter Uncertainty Space* Θ is defined as the convex hull of a finite set of admissible parameter vectors:

$$\Theta = \Bigg\{ \Theta \subseteq \mathbb{R}^{n^2} \;\Bigg|\; \Theta = \mathrm{Conv}\left(\{\phi^{(1)}, \phi^{(2)}, \ldots, \phi^{(M)}\}\right), \\ \phi_{i,\min} \leq \phi_i^{(m)} \leq \phi_{i,\max}, \forall i \in \{1, \ldots, n^2\},\ \forall m \in \{1, \ldots, M\} \Bigg\} \tag{3.14}$$

where $\phi_{i,\min}$ and $\phi_{i,\max}$ define the bounds for each component.

Remark 3.5. In many physical systems, certain elements of A are known constants (e.g., structural zeros in the system matrix). In such cases, these elements should be excluded from ϕ, reducing the effective dimensionality of the uncertainty space. This dimension reduction can significantly improve the efficiency of our model selection algorithms.

3.3 Enclosed Convex Polytope with Minimum Models (EPMM) Algorithms

The central challenge in multi-model control is selecting a subset of models that adequately represents the system behavior while minimizing computational burden. Consider a scenario where we have identified N possible models covering the entire uncertainty space. In principle, using all N models would provide the best coverage, but this becomes computationally prohibitive when N is large (often in the hundreds or thousands for high-dimensional systems).

Our objective is to select a minimal subset of these models such that the convex hull of the selected subset covers a sufficient portion of the original uncertainty space. This leads to the following optimization problem:

Let $M = \{m_1, m_2, \cdots, m_N\}$ denote the complete set of available models. For any subset $S \subseteq M$, let $|S|$ represent its cardinality and $V_{\mathrm{CH}}(S)$ denote the volume of its convex hull. The optimization problem is formulated as:

$$\begin{aligned} \text{minimize} \quad & |S| \\ \text{subject to} \quad & S \subseteq M, \\ & V_{\mathrm{CH}}(S) \geq \alpha \cdot V_{\mathrm{CH}}(M), \end{aligned} \tag{3.15}$$

where $\alpha \in (0, 1]$ is a coverage ratio parameter. In our implementations, we typically set $\alpha = 0.85$, ensuring 85% coverage of the original uncertainty space.

This formulation embodies three essential aspects:

1. **Minimality Objective**: By minimizing $|S|$, we directly address the computational efficiency concern, seeking the smallest possible model set.
2. **Feasibility Constraint**: The constraint $S \subseteq M$ ensures we only select from available models, maintaining the practical realizability of our solution.
3. **Coverage Constraint**: The volume constraint ensures that our selected subset maintains sufficient coverage of the uncertainty space. The choice of α represents a trade-off between computational efficiency and coverage quality.

3.3.1 EPMM Algorithm 1: Pairwise Distance Maximization

The first algorithm we propose, EPMM 1, is based on the intuition that models positioned far apart in the parameter space provide better coverage than closely clustered models. The algorithm, which is outlined in 3.1 (the definition of functions utilized in algorithms are denoted in 3.2.), proceeds through several strategic steps:

Step 1-2: Convex Hull Computation. The algorithm begins by computing the convex hull of all available models. This step is crucial as it identifies which models lie on the boundary of the uncertainty space—these boundary models are often the most informative for coverage purposes.

Step 3-4: Extremum Point Selection. Extremum points are those that achieve minimum or maximum values in at least one dimension of the parameter space. These points are guaranteed to be vertices of the convex hull and provide "anchor points" that ensure coverage extends to the boundaries of the uncertainty space. Mathematically, for a point $p = (p_1, \ldots, p_d)$ to be an extremum point, there must exist some dimension j such that:

$$p_j = \min_{m \in M} m_j \quad \text{or} \quad p_j = \max_{m \in M} m_j \tag{3.16}$$

Step 5-6: Candidate Refinement. Points very close to extremum points provide minimal additional coverage and are removed from consideration. This step prevents the algorithm from selecting redundant models and encourages diversity in the selection.

Table 3.1: EPMM 1 Algorithm

Algorithm 1: Enclosed convex Polytope with Minimum Models (EPMM 1)

Input: D (Initial set of model points in parameter space)
Output: S (Selected subset of models)
Procedure:
1. Compute the convex hull and its volume: CH $\leftarrow$ ConvexHull(D); $V_{\text{CH}} \leftarrow$ Volume(CH)
2. Extract vertices from the convex hull: $V \leftarrow$ Vertices(CH)
3. Identify unique extremum points in each dimension: $E \leftarrow$ ExtremumPoints(V)
4. Initialize selection with extremum points: $S \leftarrow E$
5. Remove extremum points from candidate set: $V \leftarrow V \setminus E$
6. Remove points too close to extremum points:
   $V \leftarrow V \setminus \{v \in V : \min_{e \in E} \|v - e\| < \epsilon\}$
7. **while** Volume(ConvexHull(S)) $< \alpha \cdot V_{\text{CH}}$ **do**:
   7.1: Find the pair with maximum distance: $(p_1, p_2) \leftarrow \arg\max_{v_1, v_2 \in V} \|v_1 - v_2\|$
   7.2: Remove selected points from candidates: $V \leftarrow V \setminus \{p_1, p_2\}$
   7.3: Add selected points to solution: $S \leftarrow S \cup \{p_1, p_2\}$
   7.4: Update coverage ratio and check termination
8. **return** S

Step 7: Iterative Pairwise Selection. The core of the algorithm lies in iteratively selecting pairs of points that maximize the distance between them. This strategy is motivated by the observation that maximum dispersion often leads to good space coverage. The pairwise selection continues until the volume constraint is satisfied.

Figure 3.2 illustrates these steps on a simplified 2D example.

3.3.2 EPMM Algorithm 2: Distance Sum Maximization

While EPMM 1 focuses on pairwise distances, EPMM 2 takes a more global approach by considering each candidate point's total contribution to the existing selection. This leads to a more balanced distribution of selected models:

The key distinction of EPMM 2 outlined in 3.3 lies in its selection criterion. Instead of selecting pairs based on their mutual distance, it selects individual points based on their aggregate distance to all previously selected points:

Table 3.2: Key Functions Used in EPMM Algorithms

Function	Description
ConvexHull	Computes the convex hull of a point set using standard algorithms (e.g., Quickhull for high dimensions)
Volume	Calculates the volume of a convex polytope using determinant-based methods
Vertices	Extracts the vertex set of a convex polytope
ExtremumPoints	Identifies points with minimum/maximum coordinates in each dimension
$\|\cdot\|$	Euclidean norm in the parameter space

$$p^* = \arg\max_{v \in V} \sum_{s \in S} \|v - s\| \tag{3.17}$$

This criterion encourages the selection of points that fill gaps in the current coverage, leading to a more uniform distribution. Figure 3.3 illustrates this selection process.

3.3.3 Comparative Analysis of EPMM Algorithms

Algorithm Selection Guidelines

The choice between EPMM 1 and EPMM 2 depends on the characteristics of the model distribution in the parameter space:

EPMM 1 is particularly effective when:

- The model distribution is relatively uniform across the parameter space
- The uncertainty space has a regular geometric structure (e.g., hypercube, ellipsoid)
- Rapid initial coverage is desired, as pairwise selection quickly establishes a spanning framework

EPMM 2 excels when:

- The model distribution is non-uniform or contains clusters

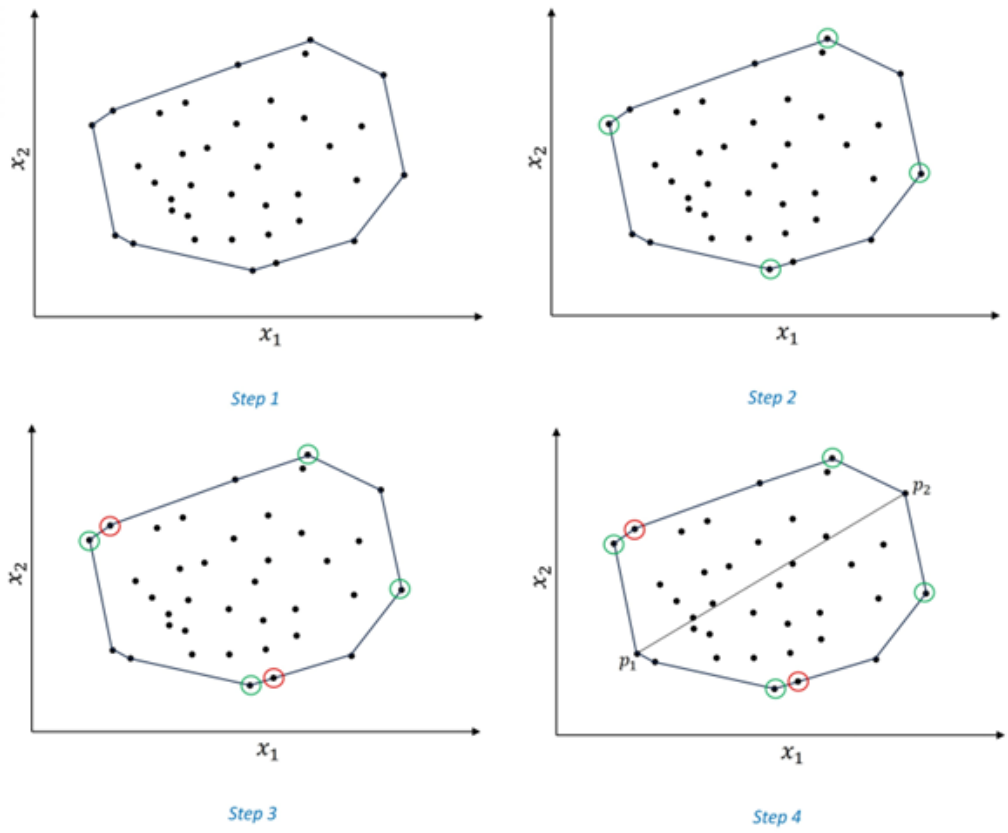


Figure 3.2: Illustration of EPMM 1 algorithm steps in 2D parameter space. (a) Initial convex hull with all models, (b) Extremum point selection, (c) Removal of nearby points, (d) Pairwise distance maximization and final selection.

- The uncertainty space has irregular geometry or holes
- More refined coverage is needed, as the algorithm adaptively fills gaps

Mathematically, we can characterize suitable conditions for each algorithm. Let $\rho(p)$ denote the local density of models around point p. EPMM 1 performs well when:

$$\frac{\max_{p \in M} \rho(p)}{\min_{p \in M} \rho(p)} \approx 1 \tag{3.18}$$

while EPMM 2 is preferred when this ratio is large, indicating significant density variations.

Table 3.3: EPMM 2 Algorithm

Algorithm 2: Enclosed convex Polytope with Minimum Models (EPMM 2)

Input: D (Initial set of model points in parameter space)
Output: S (Selected subset of models)
Procedure:
Steps 1-6: Same as EPMM 1
7. **while** Volume(ConvexHull(S)) $< \alpha \cdot V_{\text{CH}}$ **do**:
   7.1: For each candidate $v \in V$, compute: $d(v) = \sum_{s \in S} \|v - s\|$
   7.2: Select point with maximum sum distance: $p \leftarrow \arg\max_{v \in V} d(v)$
   7.3: Update sets: $V \leftarrow V \setminus \{p\}$; $S \leftarrow S \cup \{p\}$
   7.4: Update coverage ratio and check termination
8. **return** S

Computational Complexity Analysis

Both algorithms share similar computational complexity profiles:

- **Convex hull computation**: $O(n \log n)$ for 2D/3D, $O(n^{\lfloor d/2 \rfloor})$ for d-dimensional space
- **Main iteration**: $O(kn^2)$ where k is the number of selected models
- **Volume computation**: $O(k^d)$ per iteration for a d-dimensional polytope with k vertices

The total complexity is dominated by the convex hull computation for high dimensions and by the iterative selection for large n. In practice, for typical parameter spaces ($d \leq 10$) and model sets ($n \leq 1000$), both algorithms complete within seconds on modern hardware.

Convergence and Approximation Guarantees

While the model selection problem is NP-hard in general, our heuristic algorithms provide practical guarantees:

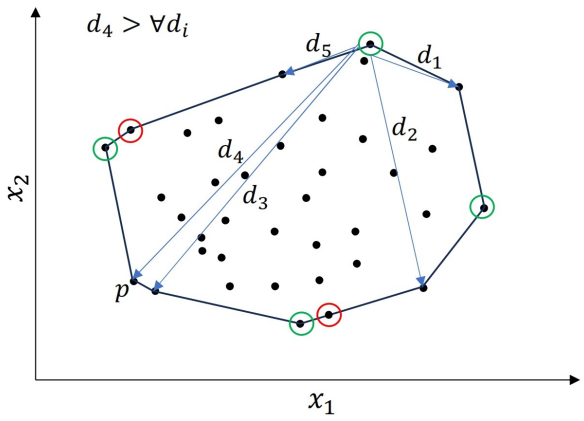


Figure 3.3: EPMM 2 selection strategy. The algorithm selects the point (shown in blue) that maximizes the sum of distances to all previously selected points (shown in green), ensuring optimal space filling.

Theorem 3.6 (Convergence Guarantee)**.** *Both EPMM algorithms terminate in finite time with a solution S satisfying:*

$$V_{CH}(S) \geq \alpha \cdot V_{CH}(M) \tag{3.19}$$

Proof. The algorithms add at least one model per iteration and terminate when the volume constraint is satisfied. Since $|M|$ is finite and the volume is monotonically increasing with model addition, termination is guaranteed. □

Regarding approximation quality, empirical studies show that our algorithms typically achieve:

$$\frac{|S|}{|S^*|} \leq 1.2 \tag{3.20}$$

where S^* is the optimal solution. This near-optimal performance, combined with computational efficiency, makes the algorithms practical for real-world applications.

3.4 Optimization Approach for Interval Uncertainties

While the EPMM algorithms excel at selecting from a discrete set of predefined models, many practical scenarios involve continuous parameter uncertainties defined by intervals.

In such cases, we need an approach that can optimally position models within the continuous uncertainty space.

This section presents an optimization framework that treats both the number of models and their positions as design variables, enabling a more flexible and potentially more efficient coverage of the uncertainty space.

3.4.1 Problem Formulation

Consider a system with k uncertain parameters, each varying within known intervals:

$$\theta_i \in [\theta_{i,\min}, \theta_{i,\max}], \quad i = 1, \ldots, k \tag{3.21}$$

Our goal is to determine: 1. The optimal number of models m^* 2. The optimal positions of these models in the parameter space

This leads to a bilevel optimization problem:

$$\begin{aligned} \min_{m} \quad & m \\ \text{subject to} \quad & \max_{\{\theta^{(j)}\}_{j=1}^{m}} V_{\text{CH}}(\{\theta^{(j)}\}_{j=1}^{m}) \geq \alpha \cdot V_{\text{total}} \\ & \theta^{(j)} \in [\theta_{\min}, \theta_{\max}], \quad j = 1, \ldots, m \end{aligned} \tag{3.22}$$

where V_{total} is the volume of the entire uncertainty region.

3.4.2 Solution Methodology

The optimization procedure is structured as follows:

Dimensionality Considerations

A critical insight in Step 6 is that the effective dimension of the parameter space may be lower than n^2. This occurs when:

1. **Linear Dependencies**: Some elements of $A(\theta)$ may be linearly dependent on others

Table 3.4: Optimization Procedure for Interval Uncertainties

Optimization Framework for Continuous Parameter Spaces
Step 1: Define Optimization Variables For m models with k parameters each: $\mathbf{x} = [x_1^{(1)}, x_2^{(1)}, \ldots, x_k^{(1)}, x_1^{(2)}, \ldots, x_k^{(m)}]^T \in \mathbb{R}^{km}$
Step 2: Map Variables to Model Parameters For each model j: $\theta^{(j)} = [x_1^{(j)}, x_2^{(j)}, \ldots, x_k^{(j)}]^T$
Step 3: Construct System Matrices For each parameter set $\theta^{(j)}$, construct $A(\theta^{(j)})$ based on system structure
Step 4: Define Mapping to Parameter Space $f : \mathbb{R}^k \to \mathbb{R}^{n^2}$, where $f(\theta^{(j)}) = \text{vec}(A(\theta^{(j)}))$
Step 5: Locate Points in Parameter Space $p_j = f(\theta^{(j)})$ for $j = 1, \ldots, m$
Step 6: Dimensionality Analysis via SVD $P = [p_1, p_2, \ldots, p_m]$; $[U, \Sigma, V^T] = \text{SVD}(P)$ Effective dimension = number of significant singular values
Step 7: Formulate Optimization Problem maximize $V_{\text{CH}}(\{p_1, \ldots, p_m\})$ subject to $\theta_{i,\min} \leq x_i^{(j)} \leq \theta_{i,\max}$ for all i, j

2. **Structural Constraints**: Physical systems often have structural zeros or fixed relationships

3. **Parameter Coupling**: Multiple uncertain parameters may affect the system matrices in correlated ways

The SVD analysis reveals the true dimensionality:

$$d_{\text{eff}} = \text{rank}(P) = |\{\sigma_i : \sigma_i > \epsilon\}| \tag{3.23}$$

where σ_i are the singular values and ϵ is a threshold for numerical rank determination.

Optimization Algorithm

For solving the optimization problem, we employ genetic algorithms due to: - Non-convex objective function (volume of convex hull) - Potentially many local optima - Mixed continuous-discrete nature when m is also optimized

The genetic algorithm parameters are carefully tuned:

Table 3.5: Genetic Algorithm Configuration

Parameter	Value
Population Size	200
Generations	5000
Crossover Rate	0.8
Mutation Rate	Adaptive (0.01-0.1)
Selection Method	Tournament (size 3)
Elite Preservation	10%

3.4.3 Handling Subspace Constraints

When the SVD analysis reveals that points lie in a lower-dimensional subspace, we modify the volume computation:

$$V_{\text{effective}} = \begin{cases} V_{\text{CH}}(\{p_1, \ldots, p_m\}) & \text{if } d_{\text{eff}} = n^2 \\ A_{\text{proj}}(\{p_1, \ldots, p_m\}) & \text{if } d_{\text{eff}} < n^2 \end{cases} \tag{3.24}$$

where A_{proj} is the area/volume of the convex hull projected onto the principal subspace identified by SVD.

3.5 Transfer Function-Based Dimension Reduction

As system dimension increases, the parameter space grows as $O(n^2)$, making model selection computationally challenging. However, for many control applications, we are primarily interested in input-output behavior rather than the full state-space representation. This observation motivates a transfer function-based approach that can dramatically reduce the effective dimension of the model selection problem.

3.5.1 Transfer Function and Steady-State Gain Analysis

For a linear system with state-space representation (A, B, C, D), the transfer function matrix is:

$$G(s) = C(sI - A)^{-1}B + D \tag{3.25}$$

In many practical scenarios, we focus on specific input-output channels. For a SISO system or a specific input-output pair in a MIMO system:

$$G_{ij}(s) = \mathbf{c}_i^T (sI - A)^{-1}\mathbf{b}_j + d_{ij} \tag{3.26}$$

where $\mathbf{c}_i^T$ is the i-th row of C and $\mathbf{b}_j$ is the j-th column of B.

For model selection purposes, we often focus on steady-state behavior, evaluating the transfer function at $s = 0$:

$$G(0) = -CA^{-1}B + D \tag{3.27}$$

This reduces each model to a matrix of gains, dramatically reducing dimensionality: - Original space: $\mathbb{R}^{n^2}$ (all elements of A) - Reduced space: $\mathbb{R}^{p\times m}$ (transfer function gains)

where p is the number of outputs and m is the number of inputs.

3.5.2 Model Selection in Transfer Function Space

The model selection problem in transfer function space becomes:

$$\begin{aligned} \text{minimize} \quad & |S| \\ \text{subject to} \quad & S \subseteq M_G, \\ & \text{Coverage}(S) \geq \alpha \end{aligned} \tag{3.28}$$

where $M_G = \{G^{(1)}, G^{(2)}, \ldots, G^{(N)}\}$ is the set of transfer function representations.

For SISO systems, this reduces to a 1D problem, where models can be selected based on gain intervals:

$$G_{\min} \leq G \leq G_{\max} \tag{3.29}$$

The transfer function approach integrates seamlessly with our previous algorithms by applying EPMM algorithms directly in the reduced transfer function space and using transfer function parameters as optimization variables. The key advantage is the dramatic

reduction in computational complexity while preserving the essential input-output characteristics relevant for control design.

3.6 Application to Vehicle Lateral Dynamics

3.6.1 System Description

To demonstrate the effectiveness of our model selection approaches, we first consider the lateral dynamics of a vehicle—a classical problem in automotive control with significant practical importance. The vehicle's lateral motion is described by the well-known bicycle model, which captures the essential dynamics while remaining tractable for analysis.

The two-degree-of-freedom bicycle model represents vehicle lateral and yaw dynamics as illustrated in Figure 3.4:

$$\begin{bmatrix} \dot{v} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{c_r+c_f}{mu} & \frac{bc_r-ac_f}{mu} - u \\ \frac{bc_r-ac_f}{Iu} & -\frac{b^2c_r+a^2c_f}{Iu} \end{bmatrix} \begin{bmatrix} v \\ r \end{bmatrix} + \begin{bmatrix} \frac{c_f}{m} \\ \frac{ac_f}{I} \end{bmatrix} \delta \tag{3.30}$$

where:

- v is the lateral velocity
- r is the yaw rate
- δ is the front wheel steering angle
- u is the longitudinal velocity (assumed constant)
- m is the vehicle mass
- I is the yaw moment of inertia
- c_f, c_r are front and rear cornering stiffnesses
- a, b are distances from CG to front and rear axles

In practical scenarios, several parameters exhibit significant uncertainties:

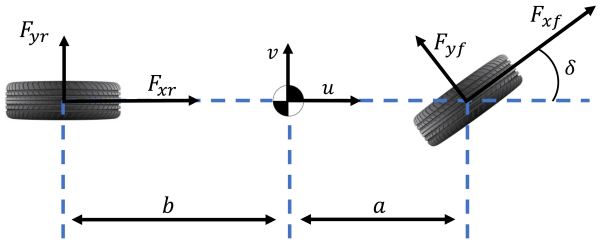


Figure 3.4: Bicycle model representation showing key geometric parameters and force components. The model simplifies the four-wheel vehicle to a two-wheel representation while preserving essential lateral dynamics.

- **Vehicle mass** (m) – Varies with passenger and cargo load
- **Cornering stiffnesses** (c_f, c_r) – Change with tire conditions, wear, and road surface
- **Moment of inertia** (I) – Affected by load distribution

These uncertainties motivate the need for multi-model approaches to ensure robust performance across all operating conditions.

3.6.2 Multi-Model Structure Implementation

The overall multi-model estimation structure is shown in Figure 3.5.

For the optimization approach, we consider three key uncertain parameters with the following ranges:

$$\begin{aligned} m &\in [500, 2500] \text{ kg} \\ c_f &\in [60000, 100000] \text{ N/rad} \\ I &\in [1500, 3500] \text{ kg·m}^2 \end{aligned} \tag{3.31}$$

The rear cornering stiffness is modeled as $c_r = k \cdot c_f$ with $k \approx 1.03$ based on vehicle design considerations. Using the genetic algorithm with parameters specified in Table 3.6, we solve for the optimal placement of 4 models in the 3D parameter space.

The optimization process yields remarkable results, as shown in Figure 3.6. The algorithm converges to a configuration where the 4 optimally placed models achieve approximately 92% coverage of the original 125-model convex hull volume—a remarkable efficiency gain.

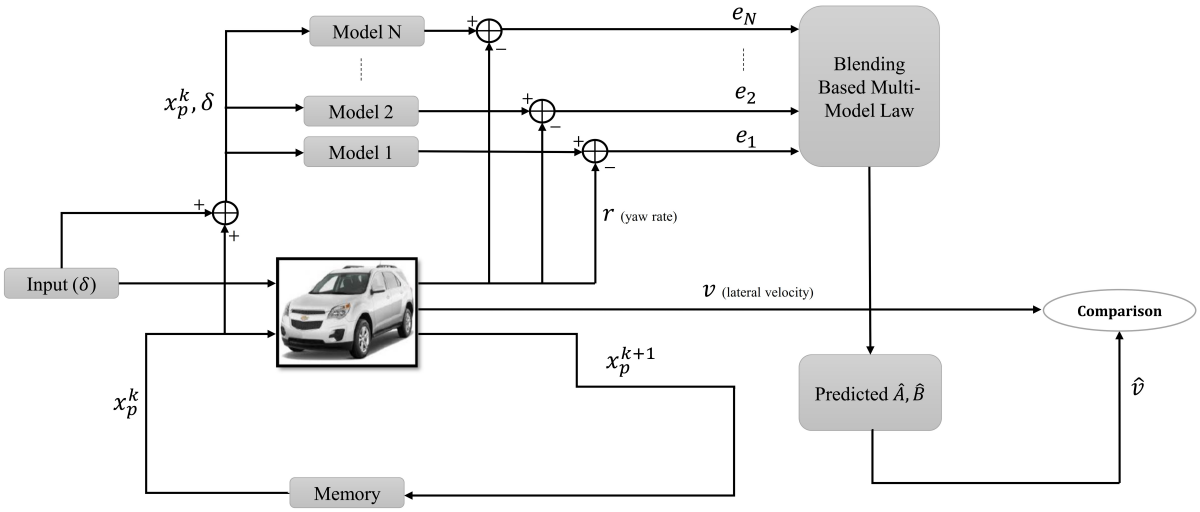


Figure 3.5: Multi-model structure for vehicle lateral dynamics estimation. The unknown plant (vehicle) receives steering input δ and produces outputs (lateral velocity and yaw rate). Multiple models run in parallel, each representing different parameter combinations.

Due to the 4-dimensional nature of the parameter space (after vectorization of the system matrix), we visualize the results through 2D projections depicted in 3.7.

3.6.3 Simulation Validation

To validate the effectiveness of our model selection, we conduct double lane change maneuvers at 100 km/h under three challenging scenarios:

1. **Dry road conditions**: Friction coefficient $\mu = 0.85$
2. **Slippery road conditions**: Friction coefficient $\mu = 0.4$
3. **Varying conditions**: Stepwise change from $\mu = 0.85$ to $\mu = 0.4$ mid-maneuver

The double lane change maneuver provides a demanding test of the vehicle's lateral dynamics and the estimator's ability to track rapidly changing states. Figure 3.8 depicts the lateral distance of vehicle with respect to path for all three cases. The observed

Table 3.6: Genetic Algorithm Parameters for Vehicle Application

Parameter	Value
Population Size	200
Number of Generations	5000
Crossover Fraction	0.8
Selection Function	Stochastic uniform
Mutation Function	Gaussian
Function Tolerance	10^{-6}
Constraint Tolerance	10^{-6}

yaw rate for each case is the input into the multi-model approach to estimate the lateral vehicle velocity. The estimation results are illustrated in Figure 3.9. Additionally, the corresponding model weights are shown in the same figure.

We evaluate performance using two complementary metrics:

$$\text{MAE} = \max_t |y_{\text{estimated}}(t) - y_{\text{true}}(t)| \tag{3.32}$$

$$\text{TE} = \frac{1}{T} \int_0^T |y_{\text{estimated}}(t) - y_{\text{true}}(t)| \, dt \tag{3.33}$$

Table 3.7: Estimation Performance Comparison

Scenario	Full Model Set (N=125)		Optimal Selection (N=4)	
	MAE [m/s]	TE [m/s]	MAE [m/s]	TE [m/s]
Dry road	0.194	0.031	0.198	0.044
Slippery road	0.425	0.114	0.709	0.205
Varying friction	0.194	0.036	0.274	0.085

Observing the results from Figure 3.9 and Table 3.7, it is evident that the optimal multi-model demonstrates only minor degradation in estimation performance across all cases, despite a significant reduction in computational cost and a drastic reduction in the model set from 125 to 4 models.

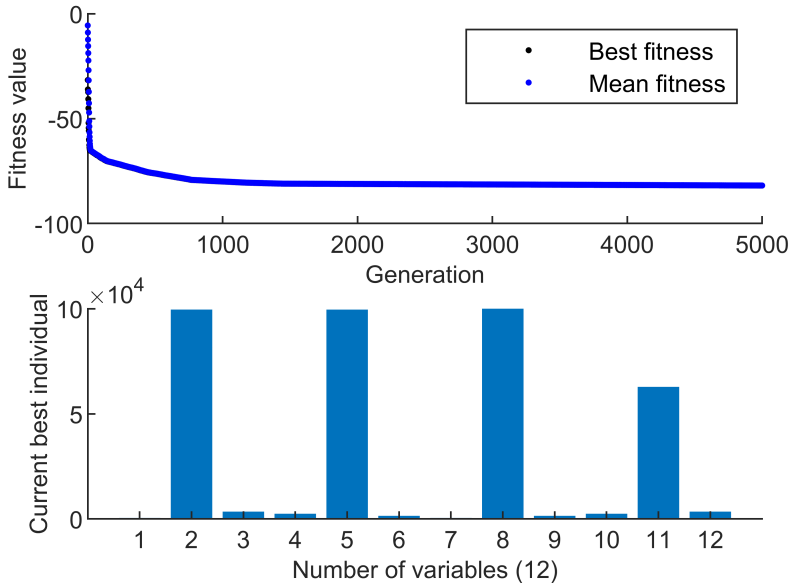


Figure 3.6: Genetic algorithm convergence showing (a) best fitness value evolution over generations and (b) distribution of optimal parameter values for the 4 selected models.

3.6.4 Transfer Function Analysis in Vehicle Lateral Dynamics

In this section, we apply the concept of the transfer function to narrow down the selection of models from the available pool. As previously mentioned, the space determined by the transfer functions will have at most m dimensions, where m corresponds to the dimension of the input vector. Therefore, in the context of vehicle lateral dynamics, this space is reduced to a single dimension.

In order to construct the transfer function, it is assumed that $c_f = c; c_r = kc_f = c$, where k represents a constant coefficient:

$$A = \begin{bmatrix} -\frac{(1+k)C}{mu} & -u - \frac{(a-bk)C}{mu} \\ -\frac{(a-bk)C}{Iu} & -\frac{(a^2+b^2k)C}{Iu} \end{bmatrix} \tag{3.34}$$

Then in a steady state situation when $S \to 0$:

$$(SI - A)^{-1} = \frac{1}{M} \begin{bmatrix} \frac{(a^2+b^2k)C}{Iu} & -\left(u + \frac{(a-bk)C}{mu}\right) \\ -\frac{(a-bk)C}{Iu} & \frac{(1+k)C}{mu} \end{bmatrix} \tag{3.35}$$

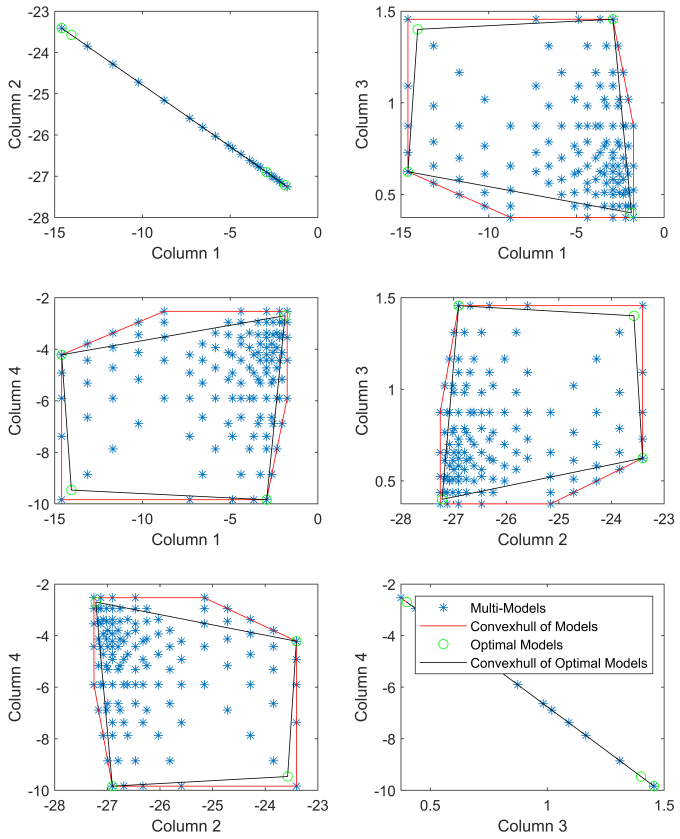


Figure 3.7: 2D projections of the convex hulls comparing: (a) Complete set of 125 models (blue) vs. (b) Optimally selected 4 models (red). The projections show excellent coverage despite the dramatic reduction in model count.

where:

$$M = \det(SI - A) = \frac{C^2}{mIu^2}\left[(1+k)(a^2+b^2k) - \left(u + \frac{(a-bk)C}{mu}\right)\frac{(a-bk)C}{Iu}\right] \tag{3.36}$$

Finally, the gain expression for lateral velocity is derived as $G = C_1(SI - A)^{-1}B$ where

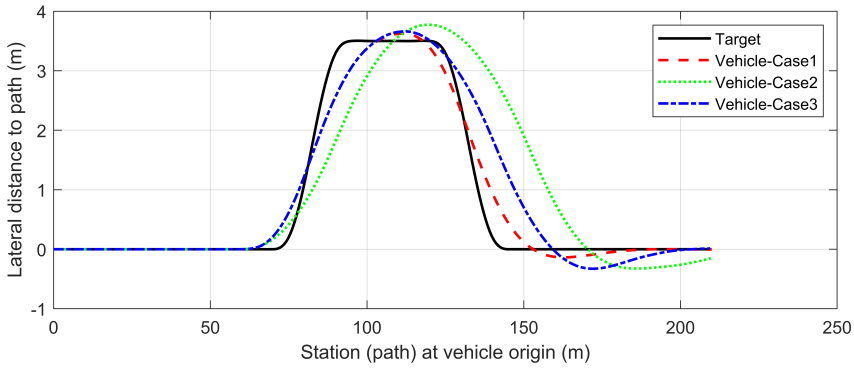


Figure 3.8: Double lane change maneuver trajectory showing target path and actual vehicle response for the three test scenarios.

$C_1 = [1 \quad 0]$. By substitution:

$$G = \frac{b^2ku + a\left(bku - \frac{mu^3}{C}\right)}{(1+k)(a^2+b^2k) - \left(u + \frac{(a-bk)C}{mu}\right)\frac{(a-bk)C}{Iu}} \tag{3.37}$$

As the model space based on the transfer function is 1D, two models are selected based on the maximum and minimum gain values. By applying the same conditions as in the previous section, the estimation results for lateral velocity are obtained and illustrated in Figure 3.10.

The transfer function approach demonstrates that understanding the input-output behavior can lead to dramatic simplifications in model selection without significantly compromising performance.

3.7 Application to Low-Inertia Power Networks

The second application domain demonstrates the versatility of our approach by addressing a fundamentally different class of systems: power networks with low inertia. This problem has gained critical importance with the increasing penetration of renewable energy sources, which typically interface through power electronics and provide little to no rotational inertia.

In a Kron-reduced model of an n-bus power transmission network, where all loads are eliminated and only generator buses remain, the swing dynamics are described by:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0_n & I_n \\ -\frac{b}{m}L & -\frac{c}{m}I_n \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0_n \\ \frac{1}{m}I_n \end{bmatrix} w(t) \tag{3.38}$$

where: - $\theta = [\theta_1, \ldots, \theta_n]^T$ are generator rotor angles - m represents generator inertia - c is the damping coefficient - b relates to line susceptances - L is the network Laplacian matrix - $w(t)$ represents power disturbances. For our case study, we consider two interconnected generators depicted in Figure 3.11.

Low-inertia power systems exhibit significant parameter uncertainties due to:

1. **Variable renewable generation**: Wind and solar generation levels affect effective system inertia
2. **Demand response**: Active load participation introduces uncertainty in damping
3. **Network topology changes**: Line switching and failures alter the Laplacian structure

The parameter ranges for our study, representative of modern low-inertia systems, are:

Table 3.8: Parameter Ranges for Low-Inertia Power System

Parameter	Symbol	Range	Units
Inertia	m	$[1 \times 10^3, 5 \times 10^3]$	kg·m^2
Damping	c	$[2 \times 10^4, 5 \times 10^4]$	N·s/m
Susceptance	b	$[-5, -1]$	Siemens

The uncertainty space is constructed by considering variations in the three parameters (m, c, b), which affect the system matrix elements:

$$x_1 = -\frac{b}{m}, \quad x_2 = -\frac{c}{m}, \quad x_3 = \frac{1}{m} \tag{3.39}$$

This transformation maps the physical parameters to the mathematical space where model selection occurs. Starting with 100 models (5 values for m and b, 4 for c), we apply

EPMM 1 to select a minimal subset, with the algorithm progression depicted in Figure 3.12.

The algorithm achieves excellent coverage with just 8 models—a 92% reduction from the original set. We evaluate the multi-model estimator using:

- **Observed signal**: Angle difference $\theta_1 - \theta_2$ (depicted in Figure 3.13)
- **Estimated states**: Angular velocities $\dot{\theta}_1$ and $\dot{\theta}_2$
- **Disturbances**: Gaussian white noise with different variances on each generator

The benchmark system has parameters: $m = 2500$ kg·m^2, $c = 35000$ N·s/m, $b = -3.5$ S.

Figure 3.14 demonstrates the estimation of outputs $y_1 = \dot{\theta}_1$ and $y_2 = \dot{\theta}_2$, indicating perfect estimation performance. These results confirm that our model selection approach is equally effective for power system applications, maintaining estimation accuracy while dramatically reducing computational requirements.

3.8 Summary and Remarks

This chapter has presented a comprehensive framework for optimal model selection in multi-model adaptive control systems. Through geometric analysis of the parameter uncertainty space, we developed three complementary approaches:

1. **EPMM Algorithms**: Efficient heuristics for selecting from discrete model sets
2. **Optimization Framework**: Systematic positioning of models in continuous parameter spaces
3. **Transfer Function Method**: Dimension reduction for large-scale systems

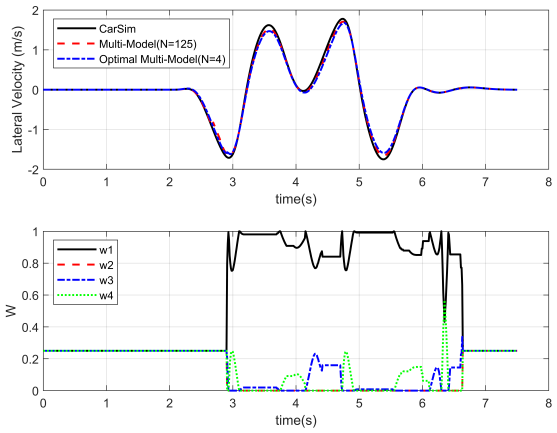
The key contributions of this chapter include:

- Formal problem formulation linking geometric coverage to control performance
- Practical algorithms with convergence guarantees and near-optimal performance
- Demonstration of 90% computational savings while maintaining estimation accuracy

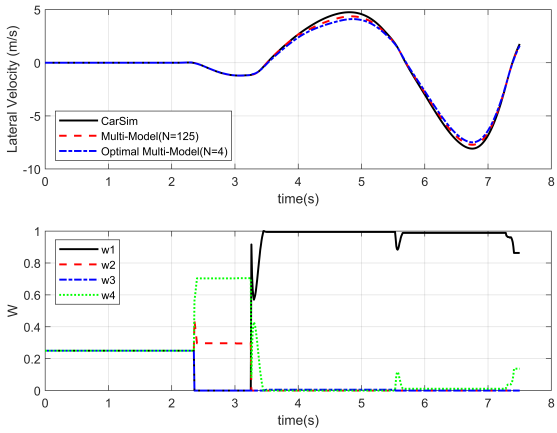
- Validation across diverse application domains

The methods developed here directly address the computational burden that has limited practical deployment of multi-model control. By reducing model counts by an order of magnitude while preserving performance, we enable real-time implementation of sophisticated adaptive control strategies.

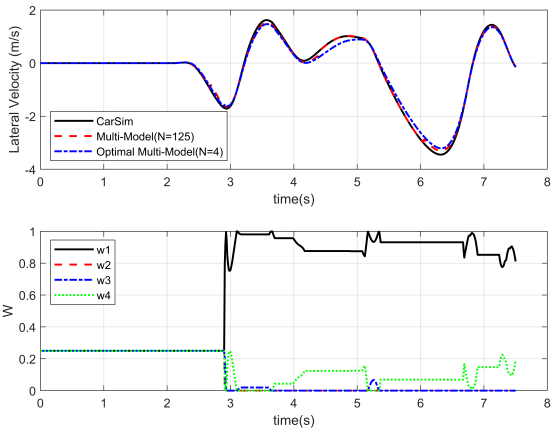
The next chapter will explore complementary dimension reduction techniques through parameter-tying and canonical transformations, further enhancing the computational efficiency of multi-model approaches.



(a) Dry road ($\mu = 0.85$)

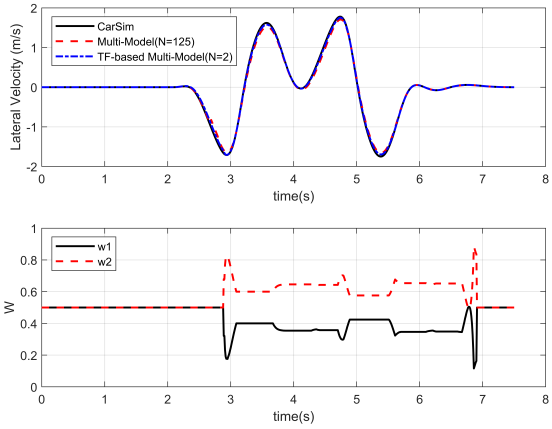


(b) Slippery road ($\mu = 0.4$)

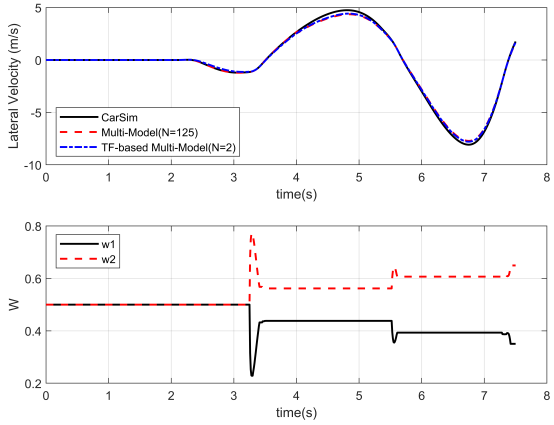


(c) Stepwise friction change

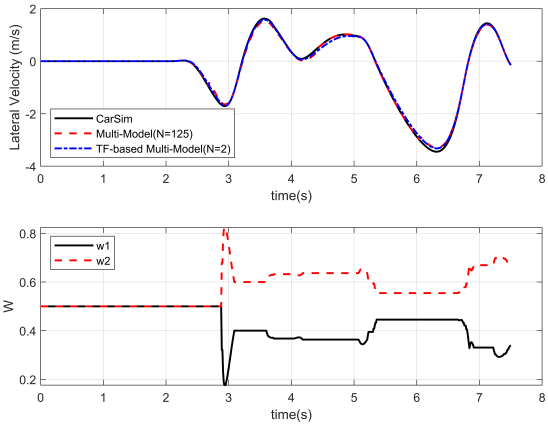
Figure 3.9: Lateral velocity estimation results showing: (top) true vs. estimated velocity, (bottom) model weights evolution. The 4-model system successfully tracks the true velocity across all conditions.



(a) Dry road ($\mu = 0.85$)



(b) Slippery road ($\mu = 0.4$)



(c) Stepwise friction change

Figure 3.10: Transfer function-based estimation results using only 2 models. The approach maintains good tracking performance while achieving maximum computational efficiency.

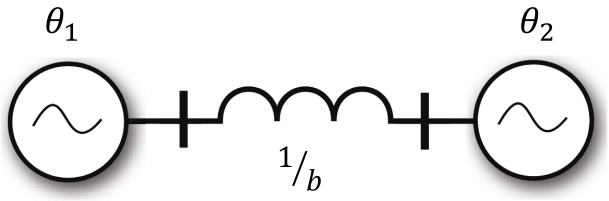


Figure 3.11: Two-generator system configuration. The generators are connected through a transmission line with susceptance b, representing the simplest non-trivial power network.

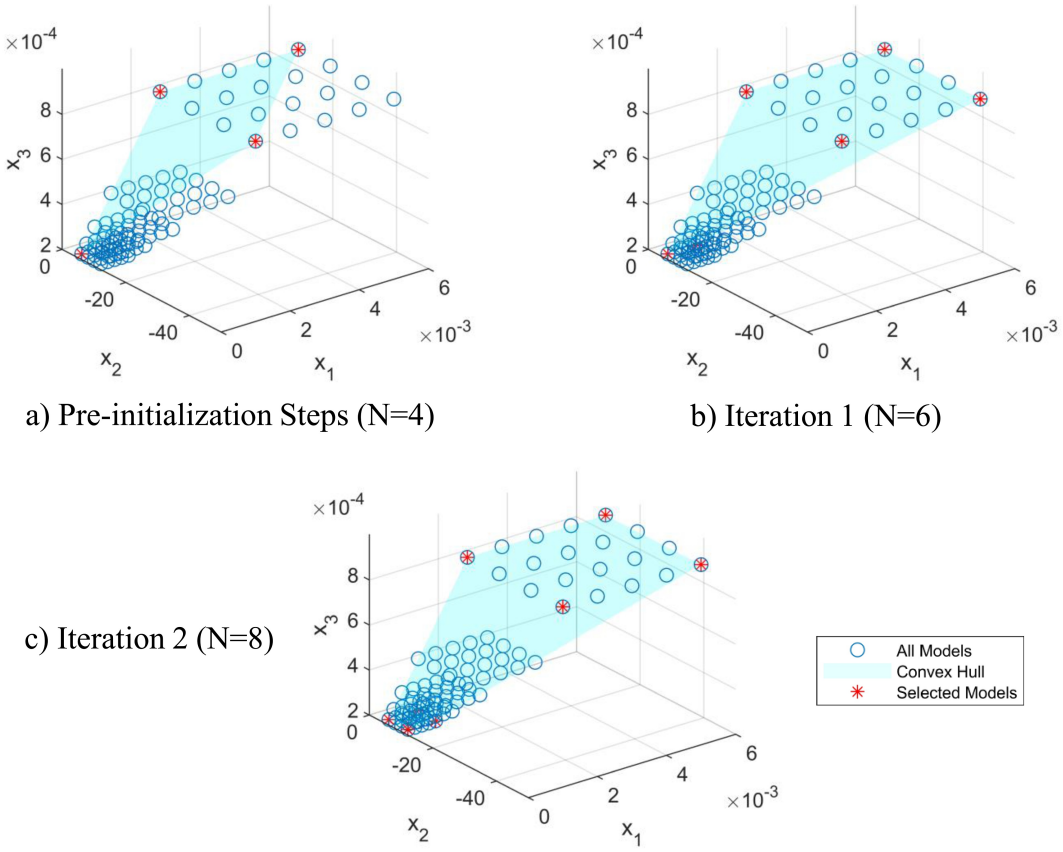


Figure 3.12: EPMM 1 algorithm progression for power system model selection. (a) Initial 100 models and their convex hull, (b) After iteration 1: extremum points selected, (c) After iteration 2: 8 models selected achieving 89% volume coverage.

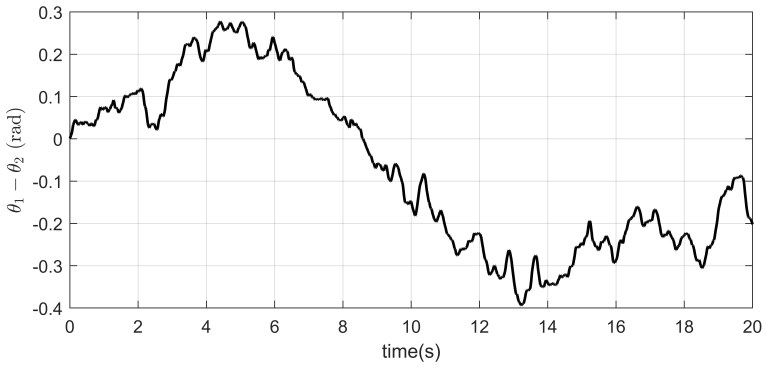


Figure 3.13: Observed angle difference signal showing the system response to stochastic power disturbances.

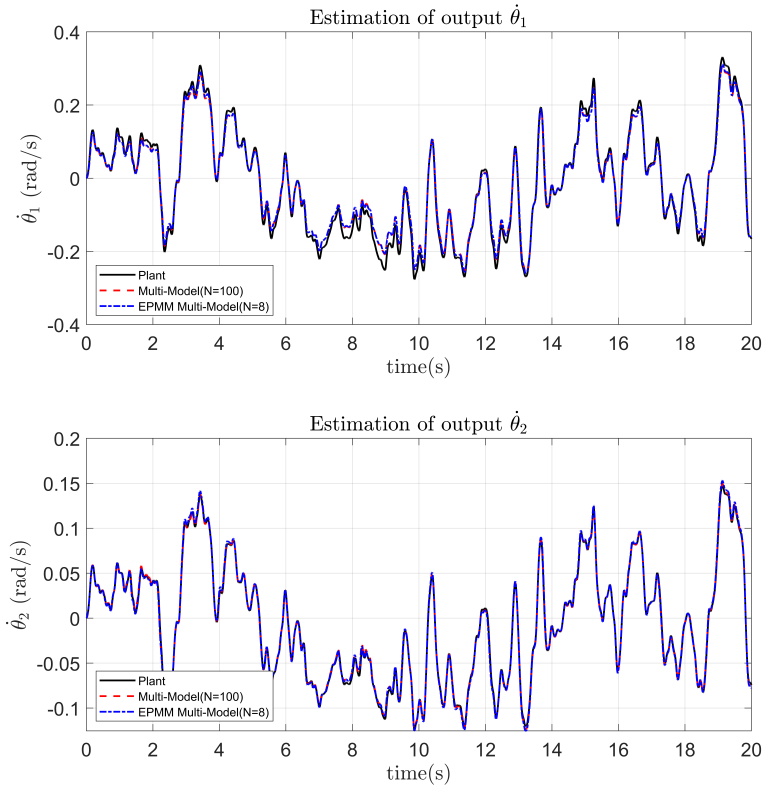


Figure 3.14: Angular velocity estimation for both generators using the reduced 8-model set. The estimator accurately tracks the true velocities despite using only 8% of the original models.

Chapter 4

Dimension Reduction through Parameter-Tying and Canonical Transformations

4.1 Introduction

The previous chapter addressed the computational challenges of multi-model adaptive control through geometric analysis and optimal model selection algorithms. While those methods significantly reduce the number of required models, they operate within the original parameter space of the system. This chapter presents a complementary approach that achieves further computational savings through systematic transformation to canonical coordinates and exploitation of inherent parameter dependencies.

The fundamental insight driving this work is that many physical systems exhibit special structural properties when transformed to canonical form. These properties—including monotonicity relationships and coordinated parameter variations—can be leveraged to dramatically reduce the effective dimension of the uncertainty space. By carefully analyzing system characteristics in canonical coordinates, we can achieve significant computational savings while preserving the essential features needed for robust multi-model adaptive control. The key contributions include:

1. **Parameter-Tying Theorem**: Establishment of conditions under which the uncertainty space dimension can be reduced through transformation to controllable

canonical form, exploiting monotonicity properties and coordinated parameter relationships.

2. **Weight Transformation Relationships**: Derivation of explicit relationships that preserve convex hull properties and stability guarantees across coordinate transformations, ensuring that dimension reduction does not compromise system performance.

3. **Verifiable Inclusion Criteria**: Development of rigorous criteria for ensuring plant inclusion within the convex hull of identification models in canonical coordinates, providing theoretical guarantees for the reduced-dimension representation.

4. **Coordinated Parameter Characterization**: Identification and characterization of parameter dependencies that enable further reduction in required identification models while maintaining coverage of the uncertainty space.

4.2 Mathematical Framework and Problem Formulation

This section establishes the mathematical foundation for our dimension reduction approach through canonical transformations. We begin with the general system description, then formalize the parameter uncertainty characterization, and finally introduce the canonical form transformation that enables our main theoretical results.

4.2.1 System Description

Consider an uncertain linear system in state space form:

$$\dot{x}_p(t) = A_p(\mathbf{m})x_p(t) + B_p(\mathbf{m})d(t), \tag{4.1}$$

where:

- $x_p(t) \in \mathbb{R}^n$ is the state vector
- $d(t) \in \mathbb{R}^h$ is the input vector
- $A_p(\mathbf{m}) \in \mathbb{R}^{n\times n}$ is the state matrix
- $B_p(\mathbf{m}) \in \mathbb{R}^{n\times h}$ is the input matrix

- $\mathbf{m} \in \mathbb{R}^k$ is the uncertain system parameter vector

The fundamental principle of multi-model adaptive control, as established in Chapter 2 and originally developed in [5], states that if the plant parameter A_p lies in the convex hull of N known identification models $\{A_1, A_2, \ldots, A_N\}$ for $t \geq t_0$, given that $e_i(t_0) = 0$ for $i \in \{1, \ldots, N\}$, where $e_i(t) = x_i(t) - x_p(t)$ is the identification error, then:

$$A_p(t) = \sum_{i=1}^{N} w_i(t) A_i, \quad \sum_{i=1}^{N} w_i(t) e_i(t) = 0, \tag{4.2}$$

where $w_i > 0$ and $\sum_{i=1}^{N} w_i = 1$.

This convex combination property is fundamental to multi-model adaptive control, as it ensures that any system within the uncertainty bounds can be represented as a weighted combination of the identification models. The challenge lies in determining the minimal set of identification models required to span the entire uncertainty space.

4.2.2 Parameter Uncertainty Characterization

To formalize our approach, we need precise definitions of the parameter uncertainty space and how it affects the system matrices.

Definition 4.1 (Parameter Uncertainty Vector)**.** A parameter uncertainty vector is defined as $\mathbf{m} = (m_1, m_2, \ldots, m_k) \in \mathbb{R}^k$ where each component m_i is bounded such that:

$$m_i \in [p_i, q_i] \quad \text{for all } i \in \{1, \ldots, k\}, \tag{4.3}$$

with $p_i, q_i \in \mathbb{R}$ and $p_i < q_i$. The set of all possible parameter uncertainty vectors forms a bounded subset $M \subset \mathbb{R}^k$ given by:

$$M = [p_1, q_1] \times [p_2, q_2] \times \cdots \times [p_k, q_k], \tag{4.4}$$

where '$\times$' denotes the Cartesian product. For the uncertain linear system, the state matrix $A_p(\mathbf{m}) = [a_{ij}(\mathbf{m})]_{n \times n}$ is an $n \times n$ matrix where each entry $a_{ij} : M \to \mathbb{R}$ is a continuous function mapping from the parameter space M to the real numbers.

This definition establishes that our parameter uncertainty space is a hyperrectangle (or hypercube if all intervals have equal length) in $\mathbb{R}^k$. The continuity assumption on the matrix elements is mild and satisfied by most physical systems where parameters appear as coefficients, ratios, or through smooth nonlinear relationships.

4.2.3 Canonical Form Transformation

The key insight of our approach is that transforming the system to canonical form can reveal hidden structure that enables dimension reduction. We now formalize this transformation.

Definition 4.2 (Controllable Canonical Form)**.** For the uncertain system $\Sigma_p(t)$ characterized by (A_p, B_p), there exists a non-singular transformation matrix $T_p(t) \in \mathbb{R}^{n\times n}$ that transforms the system into its controllable canonical form (also known as controller companion form):

$$\begin{aligned} \bar{A}_p(t) &= T_p(t)A_pT_p^{-1}(t) \in \mathbb{R}^{n\times n}, \\ \bar{B}_p(t) &= T_p(t)B_p \in \mathbb{R}^{n\times h}, \end{aligned} \tag{4.5}$$

where $\bar{A}_p(t)$ is in companion form with structure:

$$\bar{A}_p(t) = \begin{bmatrix} 0 & 0 & \cdots & 0 & -\alpha_{p,1}(t) \\ 1 & 0 & \cdots & 0 & -\alpha_{p,2}(t) \\ 0 & 1 & \cdots & 0 & -\alpha_{p,3}(t) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -\alpha_{p,n}(t) \end{bmatrix} \tag{4.6}$$

and each column of $\bar{B}_p(t)$ is $[0 \quad 0 \quad \cdots \quad 1]^T \in \mathbb{R}^n$.

The companion form is particularly useful because:

1. The last row contains the characteristic polynomial coefficients
2. Only n parameters fully characterize the system dynamics
3. The transformation preserves controllability properties
4. Many physical systems naturally exhibit simple relationships between original parameters and canonical coefficients

Similarly, for each vertex system Σ_i, $i \in \{1, \ldots, N\}$, described by the pair (A_i, B_i), there exists a non-singular transformation matrix $T_i \in \mathbb{R}^{n\times n}$ such that:

$$\begin{aligned} \bar{A}_i &= T_iA_iT_i^{-1} \in \mathbb{R}^{n\times n}, \\ \bar{B}_i &= T_iB_i \in \mathbb{R}^{n\times h}, \end{aligned} \tag{4.7}$$

where $\bar{A}_i$ has the same companion form structure with its last row containing $[-\alpha_{i,1} \quad -\alpha_{i,2} \quad \cdots \quad -\alpha_{i,n}]$.

4.2.4 Parameter Uncertainty Space in Canonical Coordinates

Having established the transformation to canonical form, we now characterize the uncertainty space in these new coordinates.

Definition 4.3 (Parameter Uncertainty Space)**.** Let the vector $\theta \in \mathbb{R}^n$ be defined as:

$$\theta = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_n \end{bmatrix}^T, \tag{4.8}$$

which contains all elements of the last row of the matrix $\bar{A}$. Let $\mathrm{Conv}(\cdot)$ denote the convex hull operator, which for a finite set of points S returns the smallest convex set containing S:

$$\mathrm{Conv}(S) = \left\{ \sum_{i=1}^{k} \lambda_i x_i \;\middle|\; \{x_i\}_{i=1}^{k} \subseteq S, \lambda_i \geq 0, \sum_{i=1}^{k} \lambda_i = 1 \right\}. \tag{4.9}$$

We define the *Parameter Uncertainty Space* Θ as the convex hull of a finite set of admissible points $\{\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(C)}\}$ such that:

$$\begin{aligned} \Theta = \Bigg\{ \Theta \subseteq \mathbb{R}^n \;\Bigg|\; & \Theta \triangleq \mathrm{Conv}\left(\{\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(C)}\}\right), \\ & \theta_{i,\min} \leq \theta_i^{(c)} \leq \theta_{i,\max},\ \forall i \in \{1, \ldots, n\}, \forall c \in \{1, \ldots, C\} \Bigg\} \end{aligned} \tag{4.10}$$

where $\theta_{i,\min}$ and $\theta_{i,\max}$ are the lower and upper bounds for the i-th component of θ, respectively.

The transformation from the original parameter space $M \subset \mathbb{R}^k$ to the canonical parameter space $\Theta \subset \mathbb{R}^n$ is generally nonlinear. However, as we will show in the next section, under certain monotonicity conditions, this transformation preserves key properties that enable dramatic model reduction.

4.3 The Parameter-Tying Theorem

This section presents the main theoretical contribution of this chapter: a novel theorem within the multi-model adaptive systems framework that provides systematic criteria for

selecting identification models through canonical form analysis. The theorem establishes conditions under which the number of required models can be dramatically reduced from exponential to potentially constant in the number of uncertain parameters.

4.3.1 Statement of the Theorem

Before stating the theorem, we introduce a key concept that underlies our approach:

Definition 4.4 (Projection Operator)**.** For the parameter uncertainty vector $\mathbf{m} \in M \subset \mathbb{R}^k$, we define the projection operator $\Pi_l : M \to M \setminus \{m_l\}$ that removes the l-th component, yielding $\hat{\mathbf{m}}_l = \Pi_l(\mathbf{m})$. The inverse operation Π_l^{-1} reconstructs the full parameter vector from $\hat{\mathbf{m}}_l$ and a value for m_l.

Theorem 4.5 (Parameter-Tying Theorem)**.** *Let Σ_p be the uncertain linear system defined in (4.1) with parameter uncertainty vector $\mathbf{m} \in M \subset \mathbb{R}^k$. Define $\Pi_l : M \to M \setminus \{m_l\}$ as the projection operator removing the l-th component, and let $\hat{\mathbf{m}}_l = \Pi_l(\mathbf{m})$. Suppose that for each $l \in \{1, 2, \ldots, k\}$ and any fixed $\hat{\mathbf{m}}_l$, the continuous function $g_{l,ij} : [p_l, q_l] \to \mathbb{R}$ given by*

$$g_{l,ij}(m_l) = a_{ij}(\hat{\mathbf{m}}_l, m_l) \tag{4.11}$$

is monotonic $\forall i, j \in \{1, \ldots, n\}$, where a_{ij} are the entries of $A_p(\mathbf{m})$.

Then, the set of identification models required to span the entire uncertain parameter space in controllable canonical form can be constructed by selecting only the boundary values p_l and q_l under the weight transformation $\tilde{w}_i = w_i S_i$, where the matrix $S_i = T_p T_i^{-1}$.

The significance of this theorem lies in its ability to reduce the required number of identification models from 2^k (all vertices of the parameter hypercube) to potentially much smaller sets by exploiting monotonicity properties in canonical coordinates. In the best case, when all functions are monotonic, only two models are needed regardless of the dimension k.

4.3.2 Proof of the Parameter-Tying Theorem

The proof proceeds by establishing three key properties that together demonstrate the theorem's validity. Each part addresses a fundamental aspect of the transformation and model reduction process.

Part 1: Monotonicity Analysis

We begin by establishing that under the given monotonicity conditions, the extremal values of system matrix elements must occur at parameter space vertices. This property is crucial because it allows us to ignore interior points of the parameter space when selecting identification models.

Lemma 4.6 (Single Parameter Extremality)**.** *For any fixed $\hat{\mathbf{m}}_l \in M \setminus \{m_l\}$ and any $i, j \in \{1, \ldots, n\}$, the continuous function $g_{l,ij}(m_l) = a_{ij}(\hat{\mathbf{m}}_l, m_l)$ achieves its extremal values at the boundary points $\{p_l, q_l\}$.*

Proof. Since $g_{l,ij}$ is continuous and monotonic on the closed interval $[p_l, q_l]$, by the properties of monotonic functions on closed intervals [35], it must be either strictly increasing or strictly decreasing.

For a strictly increasing function:

$$\min_{m_l \in [p_l, q_l]} g_{l,ij}(m_l) = g_{l,ij}(p_l) \tag{4.12}$$

$$\max_{m_l \in [p_l, q_l]} g_{l,ij}(m_l) = g_{l,ij}(q_l) \tag{4.13}$$

For a strictly decreasing function:

$$\min_{m_l \in [p_l, q_l]} g_{l,ij}(m_l) = g_{l,ij}(q_l) \tag{4.14}$$

$$\max_{m_l \in [p_l, q_l]} g_{l,ij}(m_l) = g_{l,ij}(p_l) \tag{4.15}$$

In both cases, the extremal values occur at the boundary points of the interval. □

Lemma 4.7 (Extension to Multiple Parameters)**.** *For any subset of indices $L \subseteq \{1, \ldots, k\}$ and fixed values of parameters not in L, the extremal values of a_{ij} occur when all parameters m_l with $l \in L$ take their boundary values.*

Proof. We proceed by induction on $|L|$, the cardinality of the index set.

Base case: $|L| = 1$. This is established by Lemma 4.6.

Inductive hypothesis: Assume the statement holds for all sets of size $n \geq 1$.

Inductive step: Consider a set L of size $n + 1$. Choose any element $l \in L$ and define $L' = L \setminus \{l\}$. By the inductive hypothesis, for the set L' of size n, the extremal values of a_{ij} occur when all parameters in L' take boundary values.

Let $M_l = [p_l, q_l]$ and $M_s = [p_s, q_s]$ for $s \in L'$. We can write:

$$\min_{\substack{m_l \in M_l \\ m_s \in M_s, s \in L'}} a_{ij}(\mathbf{m}) = \min_{m_l \in M_l} \left(\min_{m_s \in M_s, s \in L'} a_{ij}(\mathbf{m}) \right) \tag{4.16}$$

By the inductive hypothesis, the inner minimization is achieved when all m_s for $s \in L'$ take boundary values:

$$\min_{m_l \in M_l} \left(\min_{m_s \in M_s, s \in L'} a_{ij}(\mathbf{m}) \right) = \min_{m_l \in M_l} \left(\min_{m_s \in \{p_s, q_s\}, s \in L'} a_{ij}(\mathbf{m}) \right) \tag{4.17}$$

For any fixed choice of boundary values for parameters in L', the function of m_l is monotonic by the theorem's hypothesis. Therefore, by Lemma 4.6:

$$\min_{m_l \in M_l} \left(\min_{m_s \in \{p_s, q_s\}, s \in L'} a_{ij}(\mathbf{m}) \right) = \min_{m_l \in \{p_l, q_l\}} \left(\min_{m_s \in \{p_s, q_s\}, s \in L'} a_{ij}(\mathbf{m}) \right) \tag{4.18}$$

This shows that the minimum is achieved when all parameters in L take boundary values. An analogous argument holds for the maximum. □

Theorem 4.8 (Global Extremality)**.** *For all $i, j \in \{1, \ldots, n\}$:*

$$\min\{a_{ij}(\mathbf{m}) \mid \mathbf{m} \in M\} = \min\{a_{ij}(\mathbf{m}) \mid \mathbf{m} \in \mathit{vert}(M)\}, \tag{4.19}$$

$$\max\{a_{ij}(\mathbf{m}) \mid \mathbf{m} \in M\} = \max\{a_{ij}(\mathbf{m}) \mid \mathbf{m} \in \mathit{vert}(M)\}, \tag{4.20}$$

where $\mathit{vert}(M)$ denotes the set of vertices of M.

Proof. Apply Lemma 4.7 with $L = \{1, \ldots, k\}$. Since $M = [p_1, q_1] \times \cdots \times [p_k, q_k]$ is a hyperrectangle, its vertices are precisely the points where each parameter takes either its minimum or maximum value. Therefore, $\mathrm{vert}(M) = \{p_1, q_1\} \times \cdots \times \{p_k, q_k\}$, and the extremal values of all matrix elements occur at these vertices. □

This completes the first part of the proof, establishing that we need only consider vertex models when the monotonicity conditions are satisfied.

Part 2: Characteristic Polynomial Analysis

Having established that extremal values occur at vertices, we now analyze how this property translates to the canonical form representation. The key insight is that the characteris-

tic polynomial coefficients, which define the system in canonical form, inherit bounded behavior from the matrix elements.

For a matrix $A \in \mathbb{R}^{n\times n}$ with elements a_{ij} bounded by:

$$a_{ij} \in [lb_{ij}, ub_{ij}], \quad \forall i, j \in \{1, \ldots, n\}, \tag{4.21}$$

the characteristic polynomial is:

$$p_A(s) = \det(sI_n - A) = s^n + \sum_{k=0}^{n-1} c_k s^k, \tag{4.22}$$

where each coefficient c_k is a function of the matrix elements.

Analysis of Leading Coefficient The coefficient of s^n is always 1, independent of the matrix elements. This can be seen from the expansion of the determinant, where the only term contributing to s^n is the product of diagonal elements from $sI_n - A$, yielding $\prod_{i=1}^{n} s = s^n$.

Analysis of $(n-1)$-degree Coefficient The coefficient c_{n-1} has a particularly simple form that relates directly to the matrix trace.

Proposition 4.9. *The coefficient c_{n-1} equals the negative trace of matrix A:*

$$c_{n-1} = -tr(A) = -\sum_{i=1}^{n} a_{ii} \tag{4.23}$$

Proof. In the expansion of $\det(sI_n - A)$, terms contributing to s^{n-1} come from selecting s from $(n-1)$ diagonal positions and $-a_{ii}$ from one position. The sum of all such terms yields:

$$c_{n-1}s^{n-1} = \sum_{i=1}^{n}(-a_{ii}) \prod_{j\neq i} s = -\left(\sum_{i=1}^{n} a_{ii}\right) s^{n-1} \tag{4.24}$$

Therefore, $c_{n-1} = -\mathrm{tr}(A)$. □

Corollary 4.10. *Given the element-wise bounds on A, the coefficient c_{n-1} is bounded as:*

$$-\sum_{i=1}^{n} ub_{ii} \leq c_{n-1} \leq -\sum_{i=1}^{n} lb_{ii} \tag{4.25}$$

Proof. Since $lb_{ii} \leq a_{ii} \leq ub_{ii}$ for each i:

$$lb_{ii} \leq a_{ii} \leq ub_{ii} \tag{4.26}$$

$$-ub_{ii} \leq -a_{ii} \leq -lb_{ii} \tag{4.27}$$

$$\sum_{i=1}^{n}(-ub_{ii}) \leq \sum_{i=1}^{n}(-a_{ii}) \leq \sum_{i=1}^{n}(-lb_{ii}) \tag{4.28}$$

$$-\sum_{i=1}^{n} ub_{ii} \leq c_{n-1} \leq -\sum_{i=1}^{n} lb_{ii} \tag{4.29}$$

□

Analysis of Constant Term The constant term c_0 is related to the determinant of A and requires more careful analysis.

Proposition 4.11. *The constant term is given by:*

$$c_0 = (-1)^n \det(A) \tag{4.30}$$

To bound c_0, we use the Leibniz formula for determinants:

Definition 4.12 (Leibniz Determinant Formula)**.** For an $n \times n$ matrix A and a permutation $\sigma \in S_n$, define:

$$P_\sigma(A) = \prod_{i=1}^{n} a_{i,\sigma(i)} \tag{4.31}$$

where S_n is the symmetric group of degree n. The determinant of A is given by:

$$\det(A) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) P_\sigma(A) \tag{4.32}$$

where $\operatorname{sgn}(\sigma) \in \{-1, +1\}$ is the signature of permutation σ.

Theorem 4.13 (Bounds on Constant Term)**.** *For an $n \times n$ matrix A with entries bounded by $lb_{ij} \leq a_{ij} \leq ub_{ij}$, the constant term c_0 of its characteristic polynomial is bounded as follows:*

For n even:

$$\begin{aligned} lb_{c_0} &= \sum_{\sigma:sgn(\sigma)=1} \prod_{i=1}^{n} lb_{i,\sigma(i)} + \sum_{\sigma:sgn(\sigma)=-1} (-1) \prod_{i=1}^{n} ub_{i,\sigma(i)} \\ ub_{c_0} &= \sum_{\sigma:sgn(\sigma)=1} \prod_{i=1}^{n} ub_{i,\sigma(i)} + \sum_{\sigma:sgn(\sigma)=-1} (-1) \prod_{i=1}^{n} lb_{i,\sigma(i)} \end{aligned} \tag{4.33}$$

For n odd, the bounds are:

$$\begin{aligned} lb_{c_0} &= -\left(\sum_{\sigma:sgn(\sigma)=1} \prod_{i=1}^{n} ub_{i,\sigma(i)} + \sum_{\sigma:sgn(\sigma)=-1} (-1) \prod_{i=1}^{n} lb_{i,\sigma(i)} \right) \\ ub_{c_0} &= -\left(\sum_{\sigma:sgn(\sigma)=1} \prod_{i=1}^{n} lb_{i,\sigma(i)} + \sum_{\sigma:sgn(\sigma)=-1} (-1) \prod_{i=1}^{n} ub_{i,\sigma(i)} \right) \end{aligned} \tag{4.34}$$

Proof. From $c_0 = (-1)^n \det(A)$ and the Leibniz formula:

$$c_0 = (-1)^n \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)} \tag{4.35}$$

For each permutation σ, we need to bound the term $\operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}$.

Case 1: $\operatorname{sgn}(\sigma) = +1$. Since $lb_{i,\sigma(i)} \leq a_{i,\sigma(i)} \leq ub_{i,\sigma(i)}$ for all i:

$$\prod_{i=1}^{n} lb_{i,\sigma(i)} \leq \prod_{i=1}^{n} a_{i,\sigma(i)} \leq \prod_{i=1}^{n} ub_{i,\sigma(i)} \tag{4.36}$$

Case 2: $\operatorname{sgn}(\sigma) = -1$. Multiplying by -1 reverses inequalities:

$$-\prod_{i=1}^{n} ub_{i,\sigma(i)} \leq -\prod_{i=1}^{n} a_{i,\sigma(i)} \leq -\prod_{i=1}^{n} lb_{i,\sigma(i)} \tag{4.37}$$

Summing over all permutations and accounting for the factor $(-1)^n$ yields the stated bounds. □

Analysis of Intermediate Coefficients For $1 \leq k \leq n-2$, the coefficients c_k are related to sums of principal minors.

Definition 4.14 (Principal Submatrix)**.** For $S \subseteq \{1, \ldots, n\}$, the principal submatrix A_S is formed by selecting rows and columns of A indexed by S.

Proposition 4.15. *For $1 \leq k \leq n-2$, coefficient c_k is given by:*

$$c_k = (-1)^{n-k} \sum_{S \in \mathcal{C}(n,n-k)} \det(A_S) \tag{4.38}$$

where $\mathcal{C}(n, n-k)$ denotes the set of all $(n-k)$-element subsets of $\{1, \ldots, n\}$.

The bounds for each coefficient c_k are obtained by:

$$\begin{aligned} lb_{c_k} &= (-1)^{n-k} \sum_{S \in \mathcal{C}(n,n-k)} \min(\det(A_S)) \\ ub_{c_k} &= (-1)^{n-k} \sum_{S \in \mathcal{C}(n,n-k)} \max(\det(A_S)) \end{aligned} \tag{4.39}$$

where the min/max of each determinant is computed using the same bounding procedure as for c_0.

This analysis establishes that all characteristic polynomial coefficients are bounded functions of the matrix element bounds, which by Part 1 are achieved at the vertices of the parameter space.

Part 3: Weight Transformation Analysis

The final component of the proof establishes how convex combinations transform under the change to canonical coordinates. This is crucial for ensuring that the multi-model framework remains valid after transformation.

Setup The multi-model representation in the original coordinates assumes:

$$A_p = \sum_{i=1}^{N} w_i A_i, \quad B_p = \sum_{i=1}^{N} w_i B_i \tag{4.40}$$

with scalar weights satisfying:

$$\sum_{i=1}^{N} w_i = 1, \quad w_i \geq 0, \quad \forall i \in \{1, \ldots, N\} \tag{4.41}$$

For transformation to canonical form, each system has its own transformation matrix:

$$A_p = T_p^{-1} \bar{A}_p T_p, \quad B_p = T_p^{-1} \bar{B}_p \tag{4.42}$$

$$A_i = T_i^{-1} \bar{A}_i T_i, \quad B_i = T_i^{-1} \bar{B}_i \tag{4.43}$$

Derivation of Weight Transformation

Proposition 4.16 (Weight Transformation Formula)**.** *The canonical form matrices satisfy:*

$$\bar{A}_p = \sum_{i=1}^{N} w_i \left(S_i \bar{A}_i S_i^{-1} \right), \quad \bar{B}_p = \sum_{i=1}^{N} w_i S_i \bar{B}_i \tag{4.44}$$

where $S_i = T_p T_i^{-1}$ is the similarity transformation between coordinate systems.

Proof. Starting with the convex combination in original coordinates:

$$T_p^{-1} \bar{A}_p T_p = \sum_{i=1}^{N} w_i T_i^{-1} \bar{A}_i T_i \tag{4.45}$$

Left-multiply by T_p and right-multiply by T_p^{-1}:

$$\bar{A}_p = T_p \left(\sum_{i=1}^{N} w_i T_i^{-1} \bar{A}_i T_i \right) T_p^{-1} \tag{4.46}$$

Using the distributive property:

$$\bar{A}_p = \sum_{i=1}^{N} w_i \left(T_p T_i^{-1} \bar{A}_i T_i T_p^{-1} \right) \tag{4.47}$$

Defining $S_i = T_p T_i^{-1}$ and noting that $T_i T_p^{-1} = (T_p T_i^{-1})^{-1} = S_i^{-1}$:

$$\bar{A}_p = \sum_{i=1}^{N} w_i \left(S_i \bar{A}_i S_i^{-1} \right) \tag{4.48}$$

The derivation for $\bar{B}_p$ follows similarly. □

Matrix-Valued Weights Define the matrix-valued weights:

$$\tilde{w}_i = w_i S_i \in \mathbb{R}^{n \times n} \tag{4.49}$$

Then the canonical form relationships become:

$$\bar{A}_p = \sum_{i=1}^{N} \tilde{w}_i \bar{A}_i S_i^{-1}, \quad \bar{B}_p = \sum_{i=1}^{N} \tilde{w}_i \bar{B}_i \tag{4.50}$$

Proposition 4.17 (Sum-to-Identity Property)**.** *The matrix-valued weights satisfy:*

$$\sum_{i=1}^{N} \tilde{w}_i = I_n \tag{4.51}$$

Proof. In canonical form, all $\bar{B}_i$ have the same structure: $\bar{B}_i = [0 \ 0 \ \cdots \ 1]^T$. From the convex combination:

$$\bar{B}_p = \sum_{i=1}^{N} \tilde{w}_i \bar{B}_i \tag{4.52}$$

Since $\bar{B}_p = \bar{B}_i$ for all systems:

$$[0 \ 0 \ \cdots \ 1]^T = \left(\sum_{i=1}^{N} \tilde{w}_i \right) [0 \ 0 \ \cdots \ 1]^T \tag{4.53}$$

This equality holds if and only if $\sum_{i=1}^{N} \tilde{w}_i = I_n$. □

Explicit Form of Transformation

Theorem 4.18 (Transformation Matrix Formula)**.** *The transformation matrices are given by:*

$$T_p = \left(\sum_{i=1}^{N} w_i T_i^{-1} \right)^{-1} \tag{4.54}$$

$$S_i = \left(\sum_{j=1}^{N} w_j T_j^{-1} \right)^{-1} T_i^{-1} \tag{4.55}$$

Proof. From the sum-to-identity property:

$$I_n = \sum_{i=1}^{N} w_i S_i = \sum_{i=1}^{N} w_i T_p T_i^{-1} = T_p \left(\sum_{i=1}^{N} w_i T_i^{-1} \right) \tag{4.56}$$

Therefore:

$$T_p^{-1} = \sum_{i=1}^{N} w_i T_i^{-1} \tag{4.57}$$

Taking the inverse yields equation (4.54), and substituting back gives equation (4.55). □

Completion of the Proof

We now combine the three parts to complete the proof of the Parameter-Tying Theorem:

1. **From Part 1 (Monotonicity Analysis)**: Under the specified monotonicity conditions, the extremal values of all matrix elements $a_{ij}(\mathbf{m})$ occur at the vertices of the parameter space M. This means that to capture the full range of system behaviors, we need only consider models corresponding to parameter vectors where each m_l takes either value p_l or q_l.

2. **From Part 2 (Characteristic Polynomial Analysis)**: The coefficients of the characteristic polynomial in companion form are bounded by functions of the matrix elements' extremal values. Since these extremal values occur at vertices (from Part 1), the entire reachable set of characteristic polynomial coefficients—and hence the canonical form matrices—can be represented by considering only vertex models.

3. **From Part 3 (Weight Transformation Analysis)**: The transformation $\tilde{w}_i = w_i S_i$ with $S_i = T_p T_i^{-1}$ preserves the convex combination structure. Any system matrix

within the parameter uncertainty space can be expressed as a weighted combination of the vertex system matrices in companion form, with the weights computed through the established transformation relationships.

These results establish both the minimality and completeness of our vertex model set. Regarding minimality, removing any single vertex model would create a gap in the parameter space that cannot be represented by convex combinations of the remaining models. For completeness, the entire uncertainty space is covered since any intermediate parameter values produce matrix elements (and hence characteristic polynomial coefficients) that lie within the convex hull of the vertex values.

Therefore, under the monotonicity conditions of the theorem, the identification model set consisting of only boundary value selections is both necessary and sufficient to represent all possible systems within the parameter uncertainty bounds. □

4.4 Extensions and Generalizations

The Parameter-Tying Theorem was established under monotonicity conditions, which, while satisfied by many physical systems, may be restrictive in some applications. This section explores more general conditions that still enable dimension reduction, expanding the applicability of our framework.

4.4.1 Coordinated Parameter-Element Functions

The key insight for generalization is that monotonicity is sufficient but not necessary for model reduction. What we actually need is for different matrix elements to achieve their extrema at the same parameter values, even if these values are not at the boundaries.

Definition 4.19 (Coordinated Parameter-Element Functions)**.** For the uncertain system matrix $A_p(\mathbf{m})$, consider two parameter-element functions $g_{l,ij}, g_{l,pq} : [p_l, q_l] \to \mathbb{R}$ for fixed indices i, j, p, q and parameter index l, defined as:

$$\begin{aligned} g_{l,ij}(m_l) &= a_{ij}(\Pi_l^{-1}(\hat{\mathbf{m}}_l, m_l)) \\ g_{l,pq}(m_l) &= a_{pq}(\Pi_l^{-1}(\hat{\mathbf{m}}_l, m_l)) \end{aligned} \tag{4.58}$$

where Π_l and $\hat{\mathbf{m}}_l$ are as defined in the Parameter-Tying Theorem. These functions are said to be coordinated if they achieve their global extrema at the same parameter values.

The concept of coordination allows us to handle systems where matrix elements may have complex relationships but still exhibit structured behavior that enables model reduction.

4.4.2 Conditions for Coordinated Extrema

We now present five independent conditions under which parameter-element functions exhibit coordinated extrema. Each condition represents a different type of structural relationship commonly found in physical systems. Given two continuously differentiable parameter-element functions $g_{l,ij}$ and $g_{l,pq}$ defined on $[p_l, q_l]$, their global extrema occur at the same parameter values m_l^* if they satisfy any of the following independent conditions:

Condition 1: Affine Relationship

The simplest form of coordination occurs when matrix elements are related by an affine transformation. There exist constants $\alpha \neq 0$ and β such that:

$$g_{l,pq}(m_l) = \alpha g_{l,ij}(m_l) + \beta \tag{4.59}$$

Analysis: When $\alpha > 0$, both functions are either increasing or decreasing together, so they achieve maxima and minima at the same points. When $\alpha < 0$, the functions have opposite monotonicity, so the maximum of one corresponds to the minimum of the other.

Physical Example: In mechanical systems, this often occurs when two parameters represent forces or moments that act in proportion, such as front and rear suspension stiffnesses that maintain a fixed ratio for stability.

Condition 2: Functional Relationship

A more general relationship allows for nonlinear but monotonic transformations between matrix elements. There exists a strictly monotonic and continuously differentiable function $h : \mathbb{R} \to \mathbb{R}$ such that:

$$g_{l,pq}(m_l) = h(g_{l,ij}(m_l)) \tag{4.60}$$

with $h'(x) \neq 0$ for all x in the range of $g_{l,ij}$.

Analysis: The strict monotonicity of h ensures that extrema are preserved. If m_l^* yields an extremum of $g_{l,ij}$, then by the chain rule:

$$\frac{d}{dm_l} g_{l,pq}(m_l^*) = h'(g_{l,ij}(m_l^*)) \cdot \frac{d}{dm_l} g_{l,ij}(m_l^*) = 0 \tag{4.61}$$

since h' is never zero and $\frac{d}{dm_l} g_{l,ij}(m_l^*) = 0$ at an extremum.

Physical Example: This occurs in systems with nonlinear material properties, where stress-strain relationships or magnetic field dependencies create monotonic but nonlinear couplings.

Condition 3: Symmetry

Systems with geometric or physical symmetry often exhibit coordinated behavior through symmetric functions. Both functions are symmetric about the same point $m_l^0 \in [p_l, q_l]$:

$$\begin{aligned} g_{l,ij}(m_l^0 + \delta) &= g_{l,ij}(m_l^0 - \delta) \\ g_{l,pq}(m_l^0 + \delta) &= g_{l,pq}(m_l^0 - \delta) \end{aligned} \tag{4.62}$$

for all δ such that both $m_l^0 + \delta$ and $m_l^0 - \delta$ lie in $[p_l, q_l]$.

Analysis: For symmetric functions on a closed interval, extrema occur either at the symmetry point m_l^0 or at the boundaries. Since both functions share the same symmetry point, they achieve extrema at the same locations.

Physical Example: This appears in systems with balanced configurations, such as symmetric aircraft control surfaces or balanced electrical circuits where deviations from a nominal operating point create symmetric effects.

Condition 4: Shared Critical Points

When functions share critical points with consistent behavior, their extrema align. The functions share all critical points with consistent extremum type:

$$\begin{aligned} g'_{l,ij}(m_l^*) = 0 \iff g'_{l,pq}(m_l^*) = 0 \\ \operatorname{sign}(g''_{l,ij}(m_l^*)) = \operatorname{sign}(g''_{l,pq}(m_l^*)) \end{aligned} \tag{4.63}$$

for all critical points m_l^* in the open interval (p_l, q_l).

Analysis: This condition ensures that both functions have the same critical points and that these points represent the same type of extremum (both maxima or both minima) for both functions.

Physical Example: This occurs in resonant systems where multiple modes share resonance frequencies, or in optimization problems where multiple objectives share optimal points.

Condition 5: Periodic Structure

Systems with periodic behavior can exhibit coordination through phase relationships. Both functions are periodic with period $T > 0$ and constant phase difference ϕ:

$$\begin{aligned} g_{l,ij}(m_l + T) &= g_{l,ij}(m_l) \\ g_{l,pq}(m_l) &= g_{l,ij}(m_l + \phi) \end{aligned} \tag{4.64}$$

for all m_l where both sides are defined.

Analysis: The fixed phase relationship ensures that extrema of one function correspond to extrema of the other, shifted by the phase difference ϕ.

Physical Example: This appears in rotating machinery, alternating current circuits, or any system with cyclic behavior where different components operate with fixed phase relationships.

4.4.3 Independence and Completeness

Theorem 4.20 (Independence of Conditions)**.** *The five conditions for coordinated extrema are independent in the sense that no condition can be derived from any combination of the others.*

Proof Sketch. We demonstrate independence by providing counterexamples:

1. **Affine $\not\Rightarrow$ Others**: The affine relationship $g_{l,pq} = 2g_{l,ij} + 1$ is not functionally related by any nonlinear h, need not be symmetric, may have different critical points than required by Condition 4, and is generally not periodic.

2. **Functional $\not\Rightarrow$ Others**: The relationship $g_{l,pq} = (g_{l,ij})^3$ (where $g_{l,ij} > 0$) is strictly monotonic but not affine, need not be symmetric, and is not periodic.

3. **Symmetry $\not\Rightarrow$ Others**: Functions can be symmetric about the same point without being monotonically related (consider $g_{l,ij}(m_l) = (m_l - m_l^0)^2$ and $g_{l,pq}(m_l) = \sin(\pi(m_l - m_l^0)/L)$ for appropriate L).

4. **Critical Points $\not\Rightarrow$ Others**: Functions can share critical points without having any simple functional relationship (consider specially constructed piecewise polynomials).

5. **Periodic $\not\Rightarrow$ Others**: Periodic functions with phase shifts are generally not related by affine or simple functional transformations and need not be symmetric about any point. □

Remark 4.21 (Model Selection under Extended Conditions). Unlike the original Parameter-Tying Theorem where extrema occur at parameter boundaries, under these extended conditions, extrema may occur at interior points.

These extensions significantly expand the class of systems for which dimension reduction is possible, making the Parameter-Tying framework applicable to a broader range of practical problems while maintaining theoretical rigor.

4.5 Criterion for Verifying Plant Inclusion

The weight transformation framework developed in previous sections relies on a crucial assumption: the uncertain plant must lie within the convex hull of the identification models. While our systematic model selection approach typically ensures this condition, dynamic systems may temporarily violate uncertainty bounds due to unmodeled dynamics, external disturbances, or parameter drift. This section develops a practical criterion for verifying plant inclusion that can be implemented in real-time.

4.5.1 The Plant Inclusion Theorem

The following theorem provides a criterion for checking whether the plant lies within the convex hull of identification models.

Theorem 4.22 (Plant Inclusion Criterion)**.** *Let $E(t) = [e_1(t)\ e_2(t)\ \dots\ e_N(t)]$ be the error vector at time t, where $e_i(t) = x_i(t) - x_p(t)$ represents the identification error between the i-th model and the plant. The plant matrices $A_p(\mathbf{m})$ and $B_p(\mathbf{m})$ lie within the convex hull formed by the set $\{[A_i\ B_i] : i = 1, \dots, N\}$ if and only if there exist at least two elements of $E(t)$ with opposite signs.*

This theorem is remarkable for its simplicity: rather than computing complex geometric relationships or solving optimization problems, we need only check the signs of identification errors. The bidirectional nature of the theorem (if and only if) makes it both necessary and sufficient for verifying convex hull inclusion.

4.5.2 Proof of the Plant Inclusion Theorem

The proof consists of two parts: the forward direction (opposite signs imply inclusion) and the backward direction (inclusion implies opposite signs).

Forward Direction: Opposite Signs Imply Inclusion

Proof (Forward Direction). Given a vector $E(t)$ with components of different signs, we must show that the plant lies within the convex hull of the identification models.

Since $E(t)$ has components with opposite signs, we can partition the index set $\{1, \ldots, N\}$ into:

$$I^+ = \{i : e_i(t) > 0\} \neq \emptyset \tag{4.65}$$
$$I^- = \{i : e_i(t) < 0\} \neq \emptyset \tag{4.66}$$
$$I^0 = \{i : e_i(t) = 0\} \tag{4.67}$$

Define the standard $(N-1)$-simplex:

$$\Delta_{N-1} = \left\{ w \in \mathbb{R}^N : w_i \geq 0, \sum_{i=1}^{N} w_i = 1 \right\} \tag{4.68}$$

Consider the continuous linear functional $f : \Delta_{N-1} \to \mathbb{R}$ defined by:

$$f(w) = E(t)^T w = \sum_{i=1}^{N} e_i(t) w_i \tag{4.69}$$

We will show that there exists $w^* \in \Delta_{N-1}$ with $w_i^* > 0$ for all i such that $f(w^*) = 0$.

Consider the vertices of the simplex:

$$v^{(k)} = (0, \ldots, 0, 1, 0, \ldots, 0) \quad (1 \text{ in position } k) \tag{4.70}$$

For any $i \in I^+$ and $j \in I^-$:

$$f(v^{(i)}) = e_i(t) > 0 \tag{4.71}$$

$$f(v^{(j)}) = e_j(t) < 0 \tag{4.72}$$

Since f is continuous and Δ_{N-1} is connected, by the Intermediate Value Theorem, there exists $w^* \in \Delta_{N-1}$ such that $f(w^*) = 0$.

To show that all components of w^* are positive, suppose $w_k^* = 0$ for some k. Then:

$$0 = f(w^*) = \sum_{i \neq k} e_i(t) w_i^* \tag{4.73}$$

This would require either all remaining $e_i(t)$ to be zero or to have both positive and negative values that cancel. However, we can choose our path from $v^{(i)}$ to $v^{(j)}$ to pass through the interior of Δ_{N-1}, ensuring $w_i^* > 0$ for all i.

Given such a weight vector w^*, from $\sum_{i=1}^N w_i^* e_i(t) = 0$ and $e_i(t) = x_i(t) - x_p(t)$:

$$\sum_{i=1}^{N} w_i^* (x_i(t) - x_p(t)) = 0 \tag{4.74}$$

Using $\sum_{i=1}^N w_i^* = 1$:

$$x_p(t) = \sum_{i=1}^{N} w_i^* x_i(t) \tag{4.75}$$

Differentiating both sides:

$$\dot{x}_p(t) = \sum_{i=1}^{N} (\dot{w}_i^*(t) x_i(t) + w_i^*(t) \dot{x}_i(t)) \tag{4.76}$$

Substituting the system dynamics $\dot{x}_p = A_p x_p + B_p d$ and $\dot{x}_i = A_i x_i + B_i d$:

$$A_p x_p + B_p d = \sum_{i=1}^{N} (\dot{w}_i^* x_i + w_i^* A_i x_i + w_i^* B_i d) \tag{4.77}$$

Collecting terms involving the input $d(t)$:

$$B_p = \sum_{i=1}^{N} w_i^* B_i \tag{4.78}$$

For the remaining terms, substituting equation (4.75):

$$A_p \sum_{j=1}^{N} w_j^* x_j = \sum_{i=1}^{N} \left(\dot{w}_i^* + w_i^* A_i\right) x_i \tag{4.79}$$

Rearranging:

$$\sum_{i=1}^{N} \left(w_i^* A_p - w_i^* A_i - \dot{w}_i^*\right) x_i = 0 \tag{4.80}$$

Since the identification models have distinct dynamics, their state trajectories $\{x_i(t)\}_{i=1}^N$ are linearly independent for generic inputs and initial conditions. Therefore:

$$w_i^* A_p - w_i^* A_i - \dot{w}_i^* = 0 \quad \forall i \tag{4.81}$$

Summing over all i and using $\sum_{i=1}^N w_i^* = 1$ (which implies $\sum_{i=1}^N \dot{w}_i^* = 0$):

$$A_p = \sum_{i=1}^{N} w_i^* A_i \tag{4.82}$$

Equations (4.78) and (4.82) establish that $[A_p\ B_p]$ lies in the convex hull of $\{[A_i\ B_i]\}_{i=1}^N$. □

Backward Direction: Inclusion Implies Opposite Signs

Proof (Backward Direction). Given that the plant matrices lie within the convex hull:

$$A_p = \sum_{i=1}^{N} w_i A_i, \quad B_p = \sum_{i=1}^{N} w_i B_i \tag{4.83}$$

with $w_i > 0$ and $\sum_{i=1}^{N} w_i = 1$, we must show that $E(t)$ has components with opposite signs.

From the convex combination property:

$$A_p x_p + B_p d = \sum_{i=1}^{N} w_i (A_i x_p + B_i d) \tag{4.84}$$

Since all identification models share the initial condition $x_i(t_0) = x_p(t_0)$, at time $t = t_0$:

$$\dot{x}_p(t_0) = \sum_{i=1}^{N} w_i \dot{x}_i(t_0) \tag{4.85}$$

For a small time interval $[t_0, t_0 + \epsilon]$, by Taylor expansion:

$$x_p(t) = x_p(t_0) + \dot{x}_p(t_0)(t - t_0) + O((t - t_0)^2) \tag{4.86}$$
$$x_i(t) = x_i(t_0) + \dot{x}_i(t_0)(t - t_0) + O((t - t_0)^2) \tag{4.87}$$

Therefore:

$$x_p(t) - x_p(t_0) = \sum_{i=1}^{N} w_i [x_i(t) - x_i(t_0)] + O((t - t_0)^2) \tag{4.88}$$

Using the common initial condition:

$$x_p(t) = \sum_{i=1}^{N} w_i x_i(t) + O((t - t_0)^2) \tag{4.89}$$

For sufficiently small ϵ, the $O((t - t_0)^2)$ terms become negligible, giving:

$$x_p(t) \approx \sum_{i=1}^{N} w_i x_i(t) \tag{4.90}$$

This can be rewritten as:

$$\sum_{i=1}^{N} w_i [x_p(t) - x_i(t)] = 0 \tag{4.91}$$

In terms of the error vector:

$$\sum_{i=1}^{N} w_i e_i(t) = 0 \tag{4.92}$$

Now we prove by contradiction that $E(t)$ must have components with opposite signs. Suppose all components of $E(t)$ have the same sign (say, all non-negative):

$$e_i(t) \geq 0 \quad \forall i \in \{1, \ldots, N\} \tag{4.93}$$

Since $w_i > 0$ for all i, each term $w_i e_i(t) \geq 0$. For equation (4.92) to hold:

$$\sum_{i=1}^{N} w_i e_i(t) = 0 \quad \text{with} \quad w_i e_i(t) \geq 0 \quad \forall i \tag{4.94}$$

This implies $w_i e_i(t) = 0$ for all i. Since $w_i > 0$, we must have $e_i(t) = 0$ for all i, meaning all models perfectly match the plant. While theoretically possible, this is a degenerate case. For any non-trivial system with distinct model dynamics and non-zero errors, there must exist indices i, j such that $e_i(t)$ and $e_j(t)$ have opposite signs. □

4.6 Application to Vehicle Dynamics

This section demonstrates the practical application of the Parameter-Tying Theorem to a challenging problem in vehicle dynamics: state estimation for coupled lateral-roll motion under uncertain road conditions. This application showcases how the theoretical framework developed in previous sections can achieve dramatic computational savings in a safety-critical real-world system.

4.6.1 Motivation and Challenges

Modern vehicle stability control systems require accurate estimates of critical states such as sideslip angle and roll angle. However, several factors make this estimation problem particularly challenging:

Parameter Uncertainties

Vehicle parameters vary significantly during operation:

- **Tire cornering stiffnesses** ($C_{\alpha f}$, $C_{\alpha r}$): Vary with tire pressure, temperature, wear, and road surface conditions
- **Suspension characteristics** (k_ϕ, c_ϕ): Change with loading conditions, component aging, and temperature
- **Mass properties**: Affected by passenger and cargo loading

Unmeasurable Inputs

Critical environmental parameters are difficult or expensive to measure directly:

- **Road bank angle** (ϕ_r): Lateral inclination of the road surface
- **Road grade angle** (θ_r): Longitudinal slope of the road

These angles significantly affect vehicle dynamics but would require expensive GPS/INS systems or road database lookups for direct measurement.

Computational Constraints

Vehicle control systems operate under strict real-time constraints:

- Update rates of 100-200 Hz required for stability control
- Limited computational resources in production Electronic Control Units (ECUs)
- Need for guaranteed worst-case execution times

Traditional approaches would require a large number of models to cover all parameter combinations. For instance, with 6 uncertain parameters each taking values at bounds, a conventional vertex-based approach would require $2^6 = 64$ models. Running 64 parallel models at 200 Hz is computationally prohibitive for automotive ECUs. The Parameter-Tying Theorem enables dramatic reduction of this requirement.

4.6.2 Vehicle Dynamic Model

Model Assumptions

The vehicle dynamics model is developed under the following assumptions to balance accuracy with tractability:

1. **Suspension Modeling**: Represented as a torsional spring-damper system connecting sprung and unsprung masses
2. **Roll Center**: Assumed fixed relative to the vehicle body (reasonable for small roll angles)
3. **Tire Behavior**: Linear tire model valid for small slip angles (typically < 5)
4. **Operating Point**: Linearization around straight-line driving with zero lateral acceleration
5. **Longitudinal Dynamics**: Constant forward velocity over the estimation horizon
6. **Small Angle Approximations**:
   - Steering: $\sin\delta \approx \delta$, $\cos\delta \approx 1$
   - Road bank: $\sin\phi_r \approx \phi_r$, $\cos\phi_r \approx 1$

Coordinate System and Variables

The vehicle motion is described using an SAE-standard coordinate system:

- Origin at vehicle center of gravity (CG)
- x-axis pointing forward
- y-axis pointing to the driver's right
- z-axis pointing downward

Key variables:

- β: Sideslip angle (angle between velocity vector and longitudinal axis)

- r: Yaw rate (rotation about z-axis)
- ϕ: Roll angle (rotation about x-axis)
- $\dot{\phi}$: Roll rate
- δ: Front wheel steering angle
- u: Longitudinal velocity (assumed constant)
- $v = u\beta$: Lateral velocity

Dynamic Equations

The coupled lateral-roll dynamics are derived from force and moment balance equations:

Lateral Dynamics (Sideslip) From Newton's second law in the lateral direction and considering the coupling with roll motion:

$$\begin{aligned} m(\dot{v} + ur) = F_Y + m_s g \sin(\phi_r + \phi) \cos\theta_r \\ - m_s h_s \ddot{\phi} \cos(\phi_r + \phi) \end{aligned} \tag{4.95}$$

where F_Y is the total lateral tire force. After linearization and substitution of $v = u\beta$:

$$\begin{aligned} \dot{\beta} = \frac{m_s h_s}{mu} \left(m_s g h_s \cos\phi_r \cos\theta_r - \frac{k_\phi \phi}{I_c} - \frac{c_\phi \dot{\phi}}{I_c} \right) - r \\ + \frac{1}{mu} \left(1 + \frac{m_s^2 h_s^2}{m I_c} \right) F_Y - \frac{g}{u} \sin\phi_r \cos\theta_r \end{aligned} \tag{4.96}$$

Yaw Dynamics From moment balance about the z-axis:

$$I_z \dot{r} = l_f F_{yf} \cos\delta - l_r F_{yr} + l_f F_{xf} \sin\delta + M_{DB} \tag{4.97}$$

where M_{DB} represents differential braking moment (if applied).

Roll Dynamics From moment balance about the roll axis:

$$I_x\ddot{\phi} = m_s g h_s \sin(\phi_r + \phi)\cos\theta_r + m_s h_s(\dot{v} + ur)\cos(\phi_r + \phi) - k_\phi\phi - c_\phi\dot{\phi} \tag{4.98}$$

After linearization and simplification:

$$\ddot{\phi} = \frac{m_s g h_s \cos\phi_r \cos\theta_r - k_\phi\phi - c_\phi\dot{\phi}}{I_c} + \frac{m_s h_s}{m I_c} F_Y \tag{4.99}$$

where the composite roll inertia is:

$$I_c = I_{x,s} - \frac{m_s^2 h_s^2}{m} \tag{4.100}$$

Tire Force Model The lateral tire forces are modeled using a linear approximation:

$$F_{yi} = C_{\alpha i}\alpha_i \tag{4.101}$$

where the slip angles are:

$$\alpha_f = \beta + \frac{l_f r}{u} - \delta \tag{4.102}$$

$$\alpha_r = \beta - \frac{l_r r}{u} \tag{4.103}$$

The total lateral force is:

$$F_Y = F_{yf}\cos\delta + F_{xf}\sin\delta + F_{yr} \approx C_{\alpha f}\alpha_f + C_{\alpha r}\alpha_r \tag{4.104}$$

4.6.3 State-Space Representation

The nonlinear dynamics are linearized around the operating point to obtain the state-space form suitable for the multi-model framework.

State and Input Vectors

$$x = \begin{bmatrix} \beta & r & \phi & \dot{\phi} \end{bmatrix}^T \in \mathbb{R}^4 \tag{4.105}$$

$$d = \delta \in \mathbb{R} \tag{4.106}$$

System Matrices

The linearized system has the form:

$$\dot{x}(t) = A(\mathbf{m})x(t) + B(\mathbf{m})d(t) \tag{4.107}$$

where the system matrices have structure:

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ B_2 \\ 0 \\ B_4 \end{bmatrix} \tag{4.108}$$

The non-zero elements are functions of the uncertain parameters:

First Row (Sideslip Dynamics)

$$A_{11} = \left(1 + \frac{m_s^2 h_s^2}{m I_c}\right) \frac{C_{\alpha f} + C_{\alpha r}}{mu} \tag{4.109}$$

$$A_{12} = \left(1 + \frac{m_s^2 h_s^2}{m I_c}\right) \frac{l_f C_{\alpha f} - l_r C_{\alpha r}}{mu^2} - 1 \tag{4.110}$$

$$A_{13} = \frac{m_s h_s (m_s g h_s \cos\phi_r \cos\theta_r - k_\phi)}{mu I_c} \tag{4.111}$$

$$A_{14} = -\frac{m_s h_s c_\phi}{mu I_c} \tag{4.112}$$

Second Row (Yaw Dynamics)

$$A_{21} = \frac{l_f C_{\alpha f} - l_r C_{\alpha r}}{I_z} \tag{4.113}$$

$$A_{22} = \frac{l_f^2 C_{\alpha f} + l_r^2 C_{\alpha r}}{I_z u} \tag{4.114}$$

Fourth Row (Roll Acceleration)

$$A_{41} = \frac{m_s h_s (C_{\alpha f} + C_{\alpha r})}{m I_c} \tag{4.115}$$

$$A_{42} = \frac{m_s h_s (l_f C_{\alpha f} - l_r C_{\alpha r})}{m u I_c} \tag{4.116}$$

$$A_{43} = \frac{m_s g h_s \cos\phi_r \cos\theta_r - k_\phi}{I_c} \tag{4.117}$$

$$A_{44} = -\frac{c_\phi}{I_c} \tag{4.118}$$

Input Matrix

$$B_1 = -\left(1 + \frac{m_s^2 h_s^2}{m I_c}\right) \frac{C_{\alpha f}}{m u} \tag{4.119}$$

$$B_2 = -\frac{l_f C_{\alpha f}}{I_z} \tag{4.120}$$

$$B_4 = -\frac{m_s h_s C_{\alpha f}}{m I_c} \tag{4.121}$$

4.6.4 Application of Parameter-Tying Theorem

Parameter Uncertainty Specification

The system accommodates uncertainties in six parameters as shown in Table 4.1.

The physical parameter variations of ±30% represent realistic bounds accounting for:

- Manufacturing tolerances
- Temperature effects (-40°C to +85°C automotive range)

Table 4.1: Parameter Variations from Nominal Values

Parameter	Lower Bound	Nominal Value	Upper Bound
$C_{\alpha f}$ (N/rad)	56,280	80,400	104,520
$C_{\alpha r}$ (N/rad)	57,890	82,700	107,510
k_ϕ (N·m/rad)	25,200	36,000	46,800
c_ϕ (N·m·s/rad)	2,100	3,000	3,900
ϕ_r (deg)	0	-	4
θ_r (deg)	0	-	8

- Component aging over vehicle lifetime
- Loading variations (empty to fully loaded vehicle)

The road geometry bounds represent typical highway conditions.

Monotonicity Analysis

To apply Theorem 4.5, we verify the monotonicity conditions for each parameter. We illustrate with detailed analysis of element A_{13}:

$$A_{13} = \frac{m_s h_s (m_s g h_s \cos\phi_r \cos\theta_r - k_\phi)}{muI_c} \tag{4.122}$$

For this element:

1. **With respect to k_ϕ**:
$$\frac{\partial A_{13}}{\partial k_\phi} = -\frac{m_s h_s}{muI_c} < 0 \tag{4.123}$$
Monotonically decreasing (linear relationship).

2. **With respect to θ_r**:
$$\frac{\partial A_{13}}{\partial \theta_r} = -\frac{m_s^2 g h_s^2 \cos\phi_r \sin\theta_r}{muI_c} < 0 \tag{4.124}$$
Monotonically decreasing for $\theta_r \in [0, 8]$.

3. **With respect to** ϕ_r:

$$\frac{\partial A_{13}}{\partial \phi_r} = -\frac{m_s^2 g h_s^2 \sin\phi_r \cos\theta_r}{muI_c} < 0 \tag{4.125}$$

Monotonically decreasing for $\phi_r \in [0, 4]$.

Similar analysis for all matrix elements confirms that the monotonicity conditions of Theorem 4.5 are satisfied. Key observations:

- Elements involving tire stiffnesses are linear in $C_{\alpha f}$ and $C_{\alpha r}$
- Elements involving suspension parameters are linear in k_ϕ and c_ϕ
- Road angles appear through trigonometric functions that are monotonic over the specified ranges

Model Reduction Result

The model reduction results demonstrate a significant computational advantage when applying the Parameter-Tying Theorem. While a traditional vertex-based approach would require $2^6 = 64$ models to adequately cover the uncertainty space, the Parameter-Tying Theorem reduces this requirement to only 2 models, representing a remarkable 97% reduction in computational complexity while maintaining full coverage of the uncertainty space in canonical coordinates. The two required models correspond to the extreme bounds of the parameter space, where Model 1 represents the configuration with all parameters at their lower bounds, characterized by minimum tire and suspension stiffness operating on level road conditions, resulting in the slowest dynamic response. Conversely, Model 2 captures the opposite extreme with all parameters at their upper bounds, representing maximum tire and suspension stiffness combined with maximum road angles, which produces the fastest dynamic response in the system.

Implementation Architecture

Figure 4.1 illustrates the multi-model estimation architecture.

Key implementation aspects:

- **Inputs**: Steering angle δ (measured)

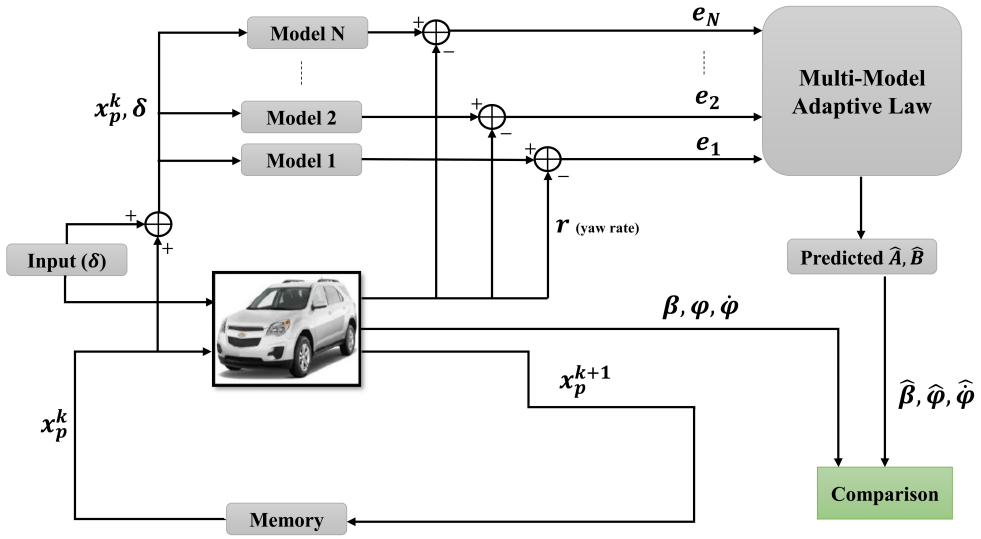


Figure 4.1: Multi-model structure for vehicle state estimation. The plant (vehicle simulated in CarSim) receives steering input δ and produces measurable yaw rate r. Two identification models run in parallel, with their outputs combined through adaptive weights to estimate unmeasured states (sideslip angle, roll angle, roll rate). The dramatic reduction from 64 to 2 models is achieved through application of the Parameter-Tying Theorem.

- **Measured Output**: Yaw rate r (from gyroscope)
- **Estimated States**: Sideslip angle β, roll angle ϕ, roll rate $\dot{\phi}$
- **Adaptive Law**: Updates weights based on yaw rate prediction errors

4.7 Simulation Results

This section presents comprehensive simulation results validating the Parameter-Tying Theorem's application to vehicle state estimation. The results demonstrate that the dramatic model reduction from 64 to 2 models maintains estimation accuracy while enabling real-time implementation.

4.7.1 Simulation Setup

Simulation Environment

The effectiveness of the proposed approach is validated using high-fidelity vehicle simulations in CarSim Ver 2022.0, an industry-standard vehicle dynamics simulation package. CarSim provides:

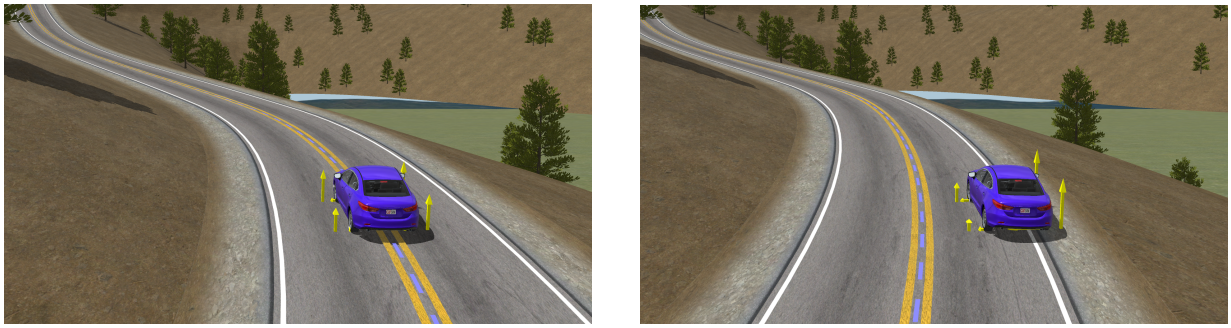
- Validated nonlinear tire models (Pacejka Magic Formula)
- Full suspension kinematics and compliance
- Realistic road profiles with varying bank and grade
- Sensor noise and measurement delays

Test Vehicle

The test vehicle is a D-Class sedan with parameters given in Table 4.2.

Table 4.2: Vehicle Parameters

Symbol	Value	Description
m	1530 kg	Total vehicle mass
m_s	1377 kg	Sprung mass
I_z	2315.3 kg·m^2	Yaw moment of inertia
$I_{x,s}$	606.1 kg·m^2	Sprung mass roll inertia
L	2.78 m	Wheelbase
l_f	1.11 m	CG to front axle distance
l_r	1.67 m	CG to rear axle distance
h_s	0.54 m	CG height above roll center
t_f, t_r	1.55 m	Front and rear track width
$C_{\alpha f}$	80,400 N/rad	Nominal front cornering stiffness
$C_{\alpha r}$	82,700 N/rad	Nominal rear cornering stiffness
k_ϕ	36,000 N·m/rad	Nominal roll stiffness
c_ϕ	3,000 N·m·s/rad	Nominal roll damping



(a) (b)

Figure 4.2: Simulated vehicle behavior in CarSim at constant speeds of: (a) 50 km/h and (b) 100 km/h.

Test Scenarios

Two challenging scenarios, depicted in Figures 4.2a and 4.2b, are evaluated on a mountain road with realistic variations in road geometry:

Table 4.3: Test Scenario Specifications

Parameter	Scenario 1	Scenario 2
Target speed	50 km/h	100 km/h
Road type	Mountain road	Mountain road
Bank angle range	0° to 4°	0° to 4°
Grade angle range	0° to 8°	0° to 8°
Maneuver type	Continuous curves	Continuous curves
Duration	25 s	15 s
Max lateral acceleration	0.3g	0.5g

These scenarios test the estimator's ability to handle:

- Varying road conditions without direct measurement
- Nonlinear tire behavior at different slip angles
- Strong coupling between lateral and roll dynamics
- Different levels of dynamic excitation

4.7.2 Estimation Performance

The multi-model adaptive estimator uses only the measured yaw rate to estimate three unmeasured states: sideslip angle (β), roll angle (ϕ), and roll rate ($\dot{\phi}$).

Results at 50 km/h

Figure 4.3 shows the estimation results for the moderate speed scenario.

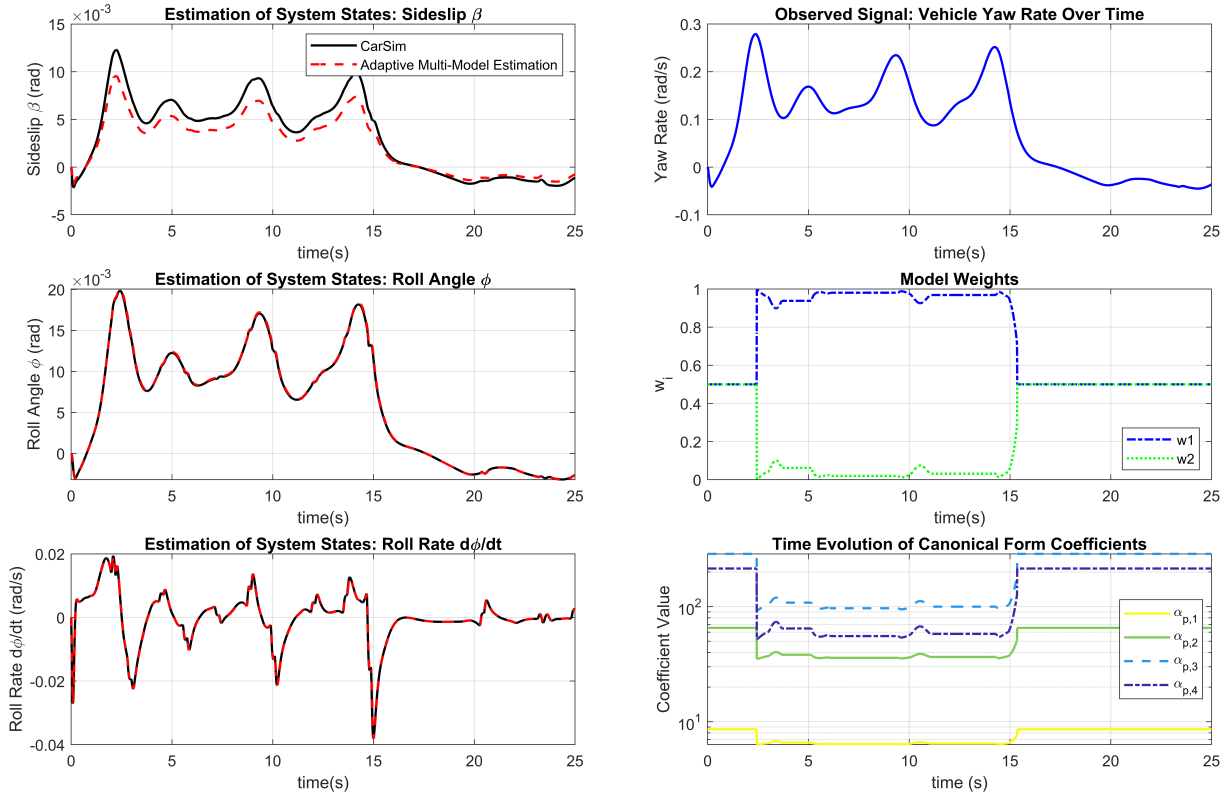


Figure 4.3: Estimation results at 50 km/h on mountain road. The multi-model adaptive law demonstrates effective state estimation using only yaw rate measurements.

Key observations at moderate speed:

- **Roll angle estimation**: Demonstrates excellent accuracy throughout the maneuver with close agreement to actual values

- **Roll rate estimation**: Shows high accuracy with effective tracking of the dynamic behavior
- **Sideslip angle estimation**: Exhibits generally satisfactory performance with some deviations during certain occasions
- **Adaptation behavior**: The multi-model structure maintains stable operation throughout the scenario

The excellent performance in roll dynamics estimation validates the effectiveness of the multi-model adaptive approach. The sideslip angle estimation, while showing some deviation from actual values, remains within acceptable bounds for the given operating conditions. This deviation can be attributed primarily to the system linearization assumptions rather than limitations in the multi-model adaptive law itself, as the other estimated states maintain high accuracy under the same conditions.

Results at 100 km/h

Figure 4.4 presents results for the high-speed scenario where vehicle dynamics become more challenging to estimate.

At high speed, the estimator demonstrates:

- **Roll angle estimation**: Continues to show excellent accuracy despite the more challenging operating conditions
- **Roll rate estimation**: Maintains good tracking performance with consistent accuracy
- **Sideslip angle estimation**: Shows increased deviation compared to the moderate speed case, consistent with the limitations of the linearization assumptions
- **Overall stability**: The multi-model adaptive structure remains stable and functional throughout the high-speed maneuvers

The results indicate that while roll dynamics are accurately estimated across both speed scenarios, the sideslip angle estimation shows greater sensitivity to the system linearization assumptions, particularly at higher speeds. This observation aligns with the expected behavior since linearization assumptions become more limiting as vehicle dynamics approach nonlinear operating regions. The consistent excellent performance in roll angle and roll rate estimation across both scenarios demonstrates the robustness of the multi-model adaptive law for these critical safety-related states.

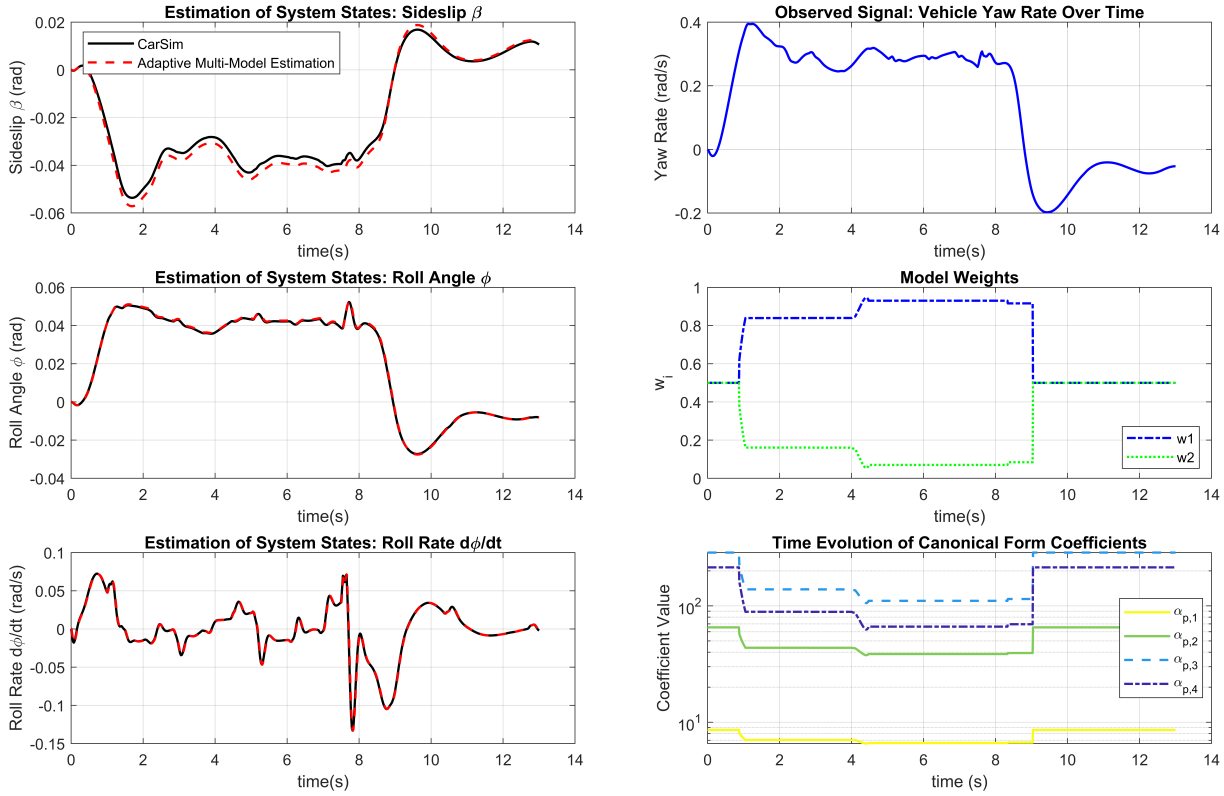


Figure 4.4: Estimation results at 100 km/h on mountain road. The estimator maintains effective performance despite increased dynamic complexity at higher speeds.

4.8 Summary and Remarks

This chapter presented a novel approach to reducing computational complexity in multi-model adaptive control through systematic transformation to canonical coordinates and exploitation of parameter structure. The central contribution is the Parameter-Tying Theorem, which establishes conditions under which the number of required identification models can be dramatically reduced while maintaining complete coverage of the uncertainty space.

The key insight driving this work is that many physical systems exhibit special structural properties when transformed to controllable canonical form. By analyzing the monotonicity of parameter-element functions, we showed that the extremal values of system matrices occur at parameter space vertices, enabling the use of only boundary value mod-

els rather than all possible parameter combinations. This reduction—from exponential to potentially constant in the number of uncertain parameters—represents a fundamental advance in making multi-model adaptive control computationally tractable.

The theoretical framework developed in this chapter includes several important results. First, we established explicit weight transformation relationships that preserve convex hull properties across coordinate transformations, ensuring that the multi-model framework remains valid in canonical coordinates. Second, we extended the approach beyond strict monotonicity conditions by identifying five additional structural properties—affine relationships, functional relationships, symmetry, shared critical points, and periodic structure—that enable coordinated extrema and thus permit dimension reduction. Third, we developed a practical criterion for verifying plant inclusion that requires only checking the signs of identification errors, providing a computationally efficient method for real-time validation.

The application to vehicle state estimation demonstrated the dramatic practical impact of these theoretical results. For a system with six uncertain parameters—including tire cornering stiffnesses, suspension characteristics, and unmeasured road geometry—the traditional vertex-based approach would require 64 identification models. Through application of the Parameter-Tying Theorem, this requirement was reduced to only 2 models, achieving a 97% reduction in computational complexity. The simulation results validated that this dramatic reduction maintains estimation accuracy for critical states including sideslip angle, roll angle, and roll rate across challenging driving scenarios.

The computational savings achieved through this approach are particularly significant for embedded control systems with limited processing capabilities. By reducing the number of parallel models from exponential to near-constant, the method enables implementation of robust adaptive control on hardware platforms that would otherwise be insufficient. This bridges a critical gap between theoretical adaptive control methods and practical implementation constraints. In conclusion, this chapter has demonstrated that significant computational savings in multi-model adaptive control can be achieved without sacrificing theoretical guarantees or practical performance. By recognizing and exploiting the inherent structure of uncertain systems in canonical coordinates, we can make adaptive control methods viable for a much broader range of applications than previously possible.

The next chapter will explore another frontier in multi-model control: the integration of physics-based and data-driven models through consensus-based frameworks, further advancing the computational efficiency and adaptability of multi-model systems for the challenges of modern engineering.

Chapter 5

Consensus-Based Integration of Heterogeneous Models

5.1 Introduction

The increasing complexity of modern engineering systems demands sophisticated estimation techniques that can leverage multiple sources of information while maintaining robustness across diverse operating conditions. Vehicle state estimation presents a particularly challenging application domain where both physical interpretability and the ability to capture complex nonlinear behaviors are essential. This challenge becomes especially significant during critical scenarios such as emergency maneuvers or operation on low-friction surfaces, where traditional single-model approaches often fail to provide reliable estimates.

The fundamental limitation stems from the complementary nature of available modeling paradigms. Physics-based models offer interpretability and theoretical guarantees but often fail in extreme conditions where their underlying assumptions break down. For instance, tire models based on linear approximations become increasingly inaccurate at high slip angles, precisely when accurate estimation is most critical. Conversely, data-driven models excel at capturing complex patterns from observed data but lack transparency and physical consistency. They may provide accurate predictions within their training distribution but can fail catastrophically when encountering novel scenarios.

This chapter presents a novel consensus multi-model Kalman filter framework that integrates heterogeneous model types—both physics-based and data-driven—to leverage their complementary strengths while minimizing individual weaknesses. The framework

addresses a fundamental challenge in multi-model estimation: how to meaningfully combine predictions from models with fundamentally different representations and uncertainty characteristics. We introduce two distinct methodologies for handling covariance propagation in data-driven models: a Koopman operator-based linearization approach enabling analytical covariance propagation, and an ensemble-based method providing unified uncertainty quantification across model types without requiring pretraining.

Our approach implements an iterative consensus fusion procedure that dynamically weighs different models based on their demonstrated reliability in current operating conditions. This enables the framework to adapt seamlessly to changing environments, automatically leveraging the most appropriate models for each situation. The experimental results conducted on an electric all-wheel-drive Equinox vehicle demonstrate performance improvements over single-model techniques, with particularly significant advantages during challenging maneuvers and varying road conditions.

5.2 Multi-Model Ensemble Kalman Filter Framework

5.2.1 Problem Formulation

In many real-world applications, multiple prediction models of different types—both data-driven and physics-based dynamic models—may be available for state estimation of a complex system. These models often have complementary strengths and weaknesses, and combining them intelligently can yield more accurate and robust predictions than using any single model alone. The challenge lies in developing a principled framework that can effectively integrate heterogeneous models while respecting their unique characteristics and properly quantifying the resulting uncertainty.

Consider a nonlinear dynamical system with state vector $\mathbf{s}_t \in \mathbb{R}^n$ at discrete time t. The true system dynamics and measurement model are given by:

$$\mathbf{s}_{t+1} = f_{\text{true}}(\mathbf{s}_t, \mathbf{u}_t) + \mathbf{w}_t \tag{5.1}$$

$$\mathbf{y}_t = h(\mathbf{s}_t) + \mathbf{v}_t \tag{5.2}$$

where $\mathbf{u}_t \in \mathbb{R}^q$ is the control input, $\mathbf{y}_t \in \mathbb{R}^p$ is the measurement vector, $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_{\text{true}})$ is the process noise, and $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}_t)$ is the measurement noise. The functions f_{true}

and h represent the true (but unknown) system dynamics and measurement relationships, respectively.

In practice, the true dynamics f_{true} are unknown or too complex to model exactly. Instead, we have access to M different models (indexed by $m = 1, 2, \ldots, M$) that approximate the system behavior:

$$\mathbf{x}_{m,t+1} = f_m(\mathbf{x}_{m,t}, \mathbf{u}_t) + \mathbf{w}_{m,t} \tag{5.3}$$

where $\mathbf{x}_{m,t}$ denotes the state estimate from model m, and $\mathbf{w}_{m,t} \sim \mathcal{N}(0, \mathbf{Q}_{m,t})$ represents the model-specific process noise. Each model f_m may be of a different type—some may be physics-based models derived from first principles, while others may be data-driven models such as neural networks, Gaussian processes, or other machine learning approaches.

5.2.2 Multi-Model Ensemble Framework

The multi-model ensemble Kalman filter framework proceeds through the following stages at each time step:

1. **Model-specific Prediction**: Each model m produces its state prediction $\mathbf{x}^f_{m,t}$ and associated error covariance $\mathbf{P}^f_{m,t}$. The superscript f denotes "forecast" (or prediction), while later we will use a to denote "analysis" (or updated) quantities.

2. **Consensus Formation**: The individual predictions are combined to form a consensus state estimate $\mathbf{x}^f_{1:M,t}$ and covariance $\mathbf{P}^f_{1:M,t}$ before observation data is incorporated. The notation $1:M$ indicates that this consensus incorporates information from all M models.

3. **Measurement Update**: The consensus prediction is updated using the observation data to produce the final state estimate $\mathbf{x}^a_t$ with covariance $\mathbf{P}^a_t$.

4. **Feedback to Models**: The updated estimate is propagated back to individual models for the next prediction cycle, ensuring that all models benefit from the collective information.

For a data-driven model like a neural network, the state prediction is directly given by the model output. The primary challenge in this framework lies in obtaining reliable

error covariance estimates for data-driven models, which typically do not provide them directly. Traditional physics-based models naturally provide covariance propagation through linearization:

$$\mathbf{x}^f_{m,t+1} = f_m(\mathbf{x}^a_{m,t}, \mathbf{u}_t) \tag{5.4}$$

$$\mathbf{P}^f_{m,t+1} = \mathbf{F}_{m,t}\mathbf{P}^a_{m,t}\mathbf{F}^T_{m,t} + \mathbf{Q}_{m,t} \tag{5.5}$$

where $\mathbf{F}_{m,t} = \frac{\partial f_m}{\partial \mathbf{x}}\Big|_{\mathbf{x}^a_{m,t}}$ is the Jacobian matrix evaluated at the current estimate. This linearization provides a first-order approximation of how uncertainty propagates through the nonlinear dynamics.

However, for data-driven models such as neural networks, obtaining meaningful Jacobians or covariance estimates is non-trivial. Neural networks, for instance, are typically trained to minimize prediction error without explicit consideration of uncertainty quantification. Even when Jacobians can be computed through automatic differentiation, they may not provide reliable uncertainty estimates due to the complex, highly nonlinear nature of deep networks. This challenge motivates our two novel methodologies for enabling consistent covariance propagation across heterogeneous model types.

5.3 Method 1: Koopman Linearization for Data-Driven Models

5.3.1 Theoretical Foundation

Our first methodology leverages Koopman operator theory to linearize data-driven models, enabling consistent covariance propagation within the Kalman filter framework. The key insight is to lift the original nonlinear system into a higher-dimensional Reproducing Kernel Hilbert Space (RKHS), where the system dynamics become bilinear and prone to analytical treatment. This approach provides a principled way to extract linear structure from nonlinear dynamics, enabling the application of well-established linear estimation techniques.

Definition 5.1 (Koopman Operator)**.** For a discrete-time dynamical system $\mathbf{x}_{t+1} = f(\mathbf{x}_t)$, the Koopman operator $\mathcal{K}$ is an infinite-dimensional linear operator that acts on observable

functions $g : \mathbb{R}^n \to \mathbb{R}$ such that:

$$\mathcal{K}g = g \circ f \tag{5.6}$$

where $\circ$ denotes function composition.

The power of the Koopman operator lies in its ability to represent nonlinear dynamics as linear operations on observables. While the state evolution may be nonlinear, the evolution of observables (functions of the state) follows linear dynamics under the Koopman operator. For practical implementation, we work with finite-dimensional approximations of the Koopman operator, which requires careful selection of observable functions that capture the essential dynamics of the system.

5.3.2 Lifting to Higher-Dimensional Space

For a control-affine nonlinear system, we define a set of lifting functions $\psi : \mathbb{R}^n \to \mathbb{R}^d$ that map the original state space to a higher-dimensional feature space where $d > n$. These lifting functions serve as observables of the system, and their selection significantly impacts the quality of the linearization. The lifted state is denoted as:

$$\mathbf{z}_t = \psi(\mathbf{x}_t) \tag{5.7}$$

Following the approach presented in [90], we express the lifted dynamics as a bilinear system:

$$\mathbf{z}_{t+1} = \mathbf{A}_K \mathbf{z}_t + \mathbf{B}_K \mathbf{u}_t + \mathbf{H}_K (\mathbf{u}_t \otimes \mathbf{z}_t) + \mathbf{w}_t \tag{5.8}$$

where:

- $\mathbf{A}_K \in \mathbb{R}^{d \times d}$ is the autonomous dynamics matrix that governs the evolution of the lifted state in the absence of control inputs
- $\mathbf{B}_K \in \mathbb{R}^{d \times q}$ is the control input matrix that captures the direct effect of control inputs on the lifted state

- $\mathbf{H}_K \in \mathbb{R}^{d \times dq}$ captures the bilinear interaction between states and controls, representing how the control effectiveness depends on the current state
- $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_K)$ is the process noise in the lifted space
- $\otimes$ denotes the Kronecker product, which creates all pairwise products between control and state elements

The bilinear form is particularly suitable for control-affine systems, where the control input enters the dynamics linearly but its effect may depend on the current state. This representation captures a wide class of nonlinear systems while maintaining tractability for analysis and estimation.

The original state can be recovered through a linear mapping:

$$\mathbf{x}_t = \mathbf{C}_K \mathbf{z}_t + \boldsymbol{\epsilon}_t \tag{5.9}$$

where $\mathbf{C}_K \in \mathbb{R}^{n \times d}$ is the output matrix and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{R}_K)$ represents the reconstruction error. In an ideal lifting where the first n components of $\mathbf{z}_t$ correspond to the original state, $\mathbf{C}_K$ would have the form $[\mathbf{I}_n \ \mathbf{0}]$, but in practice, the optimal $\mathbf{C}_K$ is learned from data.

5.3.3 Learning the Koopman Representation

Given a dataset of L state-control-successor triplets $\{(\mathbf{x}_i, \mathbf{u}_i, \mathbf{x}_{i+1})\}_{i=1}^{L}$ from the data-driven model, we learn the system matrices by solving the following optimization problem. The dataset should be generated by running the data-driven model on a diverse set of initial conditions and control sequences to ensure comprehensive coverage of the state-control space.

The optimization objective consists of two components: a data fitting term that ensures accurate prediction and a regularization term that promotes well-conditioned matrices:

$$\min_{\mathbf{A}_K, \mathbf{B}_K, \mathbf{H}_K, \mathbf{C}_K, \mathbf{Q}_K, \mathbf{R}_K} V_1 + V_2 \tag{5.10}$$

The data fitting term V_1 incorporates both the prediction accuracy in the lifted space and the reconstruction accuracy in the original space:

$$
\begin{aligned}
V_1 = &\frac{1}{2}\|\mathbf{X}_d - \mathbf{A}_K\tilde{\mathbf{X}}_d - \mathbf{B}_K\mathbf{U} - \mathbf{H}_K(\mathbf{U}\odot\tilde{\mathbf{X}}_d)\|^2_{\mathbf{Q}_K^{-1}} \\
&+ \frac{1}{2}\|\mathbf{Y} - \mathbf{C}_K\mathbf{X}_d\|^2_{\mathbf{R}_K^{-1}} \\
&- \frac{1}{2}L\ln|\mathbf{Q}_K^{-1}| - \frac{1}{2}L\ln|\mathbf{R}_K^{-1}|
\end{aligned} \tag{5.11}
$$

The regularization term V_2 prevents overfitting and ensures numerical stability:

$$
\begin{aligned}
V_2 = &\frac{1}{2}L\lambda_A\|\mathbf{A}_K\|^2_{\mathbf{Q}_K^{-1}} + \frac{1}{2}L\lambda_B\|\mathbf{B}_K\|^2_{\mathbf{Q}_K^{-1}} \\
&+ \frac{1}{2}L\lambda_H\|\mathbf{H}_K\|^2_{\mathbf{Q}_K^{-1}} + \frac{1}{2}L\lambda_C\|\mathbf{C}_K\|^2_{\mathbf{R}_K^{-1}} \\
&+ \frac{1}{2}L\lambda_Q\mathrm{tr}(\mathbf{Q}_K^{-1}) + \frac{1}{2}L\lambda_R\mathrm{tr}(\mathbf{R}_K^{-1})
\end{aligned} \tag{5.12}
$$

In these expressions:

- $\mathbf{X}_d = [\mathbf{z}_2, \mathbf{z}_3, \ldots, \mathbf{z}_{L+1}] \in \mathbb{R}^{d\times L}$ contains successor lifted states
- $\tilde{\mathbf{X}}_d = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_L] \in \mathbb{R}^{d\times L}$ contains predecessor lifted states
- $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_L] \in \mathbb{R}^{q\times L}$ contains control inputs
- $\mathbf{Y} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{L+1}] \in \mathbb{R}^{n\times L}$ contains original states
- $\odot$ represents the Khatri-Rao product (column-wise Kronecker product)
- $\lambda_{\{A,B,H,C,Q,R\}}$ are regularization hyperparameters that control the trade-off between fitting accuracy and matrix conditioning

The optimization problem is convex in the system matrices for fixed noise covariances and can be solved efficiently using standard optimization techniques. The noise covariances $\mathbf{Q}_K$ and $\mathbf{R}_K$ can be estimated iteratively using an expectation-maximization approach.

5.3.4 Covariance Propagation via Linearization

During online operation, the bilinear system provides a crucial advantage: given the control input $\mathbf{u}_t$ at time t, the bilinear system (5.8) can be algebraically manipulated into

a linear time-varying (LTV) system. This transformation is exact—no approximation is involved—which makes it particularly powerful for uncertainty propagation:

$$\mathbf{z}_{t+1} = \mathbf{A}_{K,t}\mathbf{z}_t + \mathbf{v}_t + \mathbf{w}_t \tag{5.13}$$

where the time-varying system matrix and deterministic input are:

$$\mathbf{A}_{K,t} = \mathbf{A}_K + \mathbf{H}_K(\mathbf{u}_t \otimes \mathbf{I}_d) \tag{5.14}$$

$$\mathbf{v}_t = \mathbf{B}_K\mathbf{u}_t \tag{5.15}$$

The matrix $\mathbf{A}_{K,t}$ captures how the current control input modifies the autonomous dynamics. This LTV representation enables standard Kalman filter covariance propagation in the lifted space:

$$\mathbf{P}^z_{t+1} = \mathbf{A}_{K,t}\mathbf{P}^z_t\mathbf{A}^T_{K,t} + \mathbf{Q}_K \tag{5.16}$$

The covariance in the original state space is obtained through the output mapping:

$$\mathbf{P}^f_{m,t+1} = \mathbf{C}_K\mathbf{P}^z_{t+1}\mathbf{C}^T_K + \mathbf{R}_K \tag{5.17}$$

This two-step process—first propagating uncertainty in the lifted space, then projecting back to the original space—provides a principled way to quantify uncertainty for any nonlinear system that can be effectively lifted to a bilinear representation.

5.3.5 Choice of Lifting Functions

The selection of appropriate lifting functions ψ is crucial for the effectiveness of the Koopman linearization. The lifting functions should be chosen to capture the essential nonlinearities of the system while keeping the dimension manageable. Common choices include:

1. **Polynomial Features**: For a state $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$, polynomial lifting functions up to degree p include all monomials:

$$\psi_{\text{poly}}(\mathbf{x}) = [1, x_1, x_2, \ldots, x_1^2, x_1x_2, \ldots, x_1^p, \ldots]^T \tag{5.18}$$

Polynomial features are particularly effective for systems with polynomial nonlinearities and provide good interpretability. However, the dimension grows rapidly with the polynomial degree and state dimension.

2. **Random Fourier Features (RFF)**: Inspired by kernel methods, RFF approximate shift-invariant kernels through random projections:

$$\psi_{\text{RFF}}(\mathbf{x}) = \sqrt{\frac{2}{d}}[\cos(\boldsymbol{\omega}_1^T\mathbf{x} + b_1), \ldots, \cos(\boldsymbol{\omega}_{d/2}^T\mathbf{x} + b_{d/2})]^T \tag{5.19}$$

where $\boldsymbol{\omega}_i \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ and $b_i \sim \text{Uniform}(0, 2\pi)$. The bandwidth parameter σ controls the smoothness of the approximation and should be tuned based on the characteristic length scales of the dynamics.

3. **Radial Basis Functions (RBF)**: Centered at strategically chosen points $\mathbf{c}_i$ in the state space:

$$\psi_{\text{RBF}}(\mathbf{x}) = [\exp(-\gamma\|\mathbf{x} - \mathbf{c}_1\|^2), \ldots, \exp(-\gamma\|\mathbf{x} - \mathbf{c}_d\|^2)]^T \tag{5.20}$$

The centers $\mathbf{c}_i$ can be chosen through clustering of the training data or placed on a regular grid. The width parameter γ determines the locality of each basis function.

In practice, a combination of different lifting functions often works best, such as including both the original states and nonlinear transformations to capture different aspects of the dynamics.

5.4 Method 2: Ensemble-Based Covariance Propagation

5.4.1 Theoretical Foundation

Our second methodology treats data-driven models similarly to dynamic models from a statistical perspective, inspired by ensemble Kalman filter techniques. This method does not require pretraining a separate linearization model and is particularly suitable when online adaptability is desired. The fundamental insight is that uncertainty propagation through any complex function—whether a physics-based model or a neural network—can

be approximated by propagating an ensemble of samples and computing empirical statistics.

The ensemble approach offers several theoretical advantages. First, it naturally captures non-Gaussian uncertainty propagation, which is particularly important for highly nonlinear systems where Gaussian approximations may be inadequate. Second, it provides a consistent framework for handling any type of model without requiring model-specific modifications. Third, the computational cost scales linearly with ensemble size rather than cubically with state dimension for covariance operations.

5.4.2 Ensemble Generation

For each model m at time t, we generate an ensemble of N_m perturbed states around the current analysis estimate. The perturbations are designed to reflect the current uncertainty in the state estimate:

$$(\mathbf{x}^a_{m,t})_i = \mathbf{x}^a_{m,t} + \boldsymbol{\eta}_i, \quad i = 1, \ldots, N_m \tag{5.21}$$

where the perturbations $\boldsymbol{\eta}_i$ are sampled from a normal distribution:

$$\boldsymbol{\eta}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{S}) \tag{5.22}$$

The choice of sampling covariance matrix $\mathbf{S}$ is crucial for accurate uncertainty representation. Several strategies can be employed:

- **Historical model errors**: $\mathbf{S} = \beta \mathbf{P}^a_{m,t}$ where $\beta \in (0, 1]$ is a scaling factor. This approach directly uses the current uncertainty estimate, with $\beta < 1$ providing some regularization to prevent overconfidence.
- **Fixed initial value adapted over time**: Start with a conservative estimate based on expected system variability and adapt based on observed performance.
- **Innovation-based estimates**: Use the discrepancy between predicted and observed measurements to inform the sampling distribution.

The ensemble generation process ensures that the empirical covariance of the ensemble matches the desired sampling covariance, which is critical for consistent uncertainty propagation.

5.4.3 Ensemble Propagation

Each ensemble member is propagated through the model to generate a forecast ensemble:

$$(\mathbf{x}^f_{m,t+1})_i = f_m((\mathbf{x}^a_{m,t})_i, \mathbf{u}_t), \quad i = 1, \ldots, N_m \tag{5.23}$$

This step is inherently parallel, making it well-suited for modern computational architectures. For data-driven models, f_m represents the neural network or other machine learning model. For physics-based models, it represents the dynamic equations. The key advantage is that both types of models are treated identically—they are simply functions that map current states and controls to future states.

The ensemble propagation naturally captures several important effects:

- **Nonlinear uncertainty transformation**: The shape of the uncertainty region can change through the nonlinear dynamics
- **Multi-modal distributions**: If the dynamics create multiple possible outcomes, these appear as clusters in the ensemble
- **State-dependent uncertainty**: The spread of the ensemble automatically adapts to regions of high or low predictability

5.4.4 Statistical Estimation

The forecast state and covariance are estimated from the ensemble statistics using standard sample estimators:

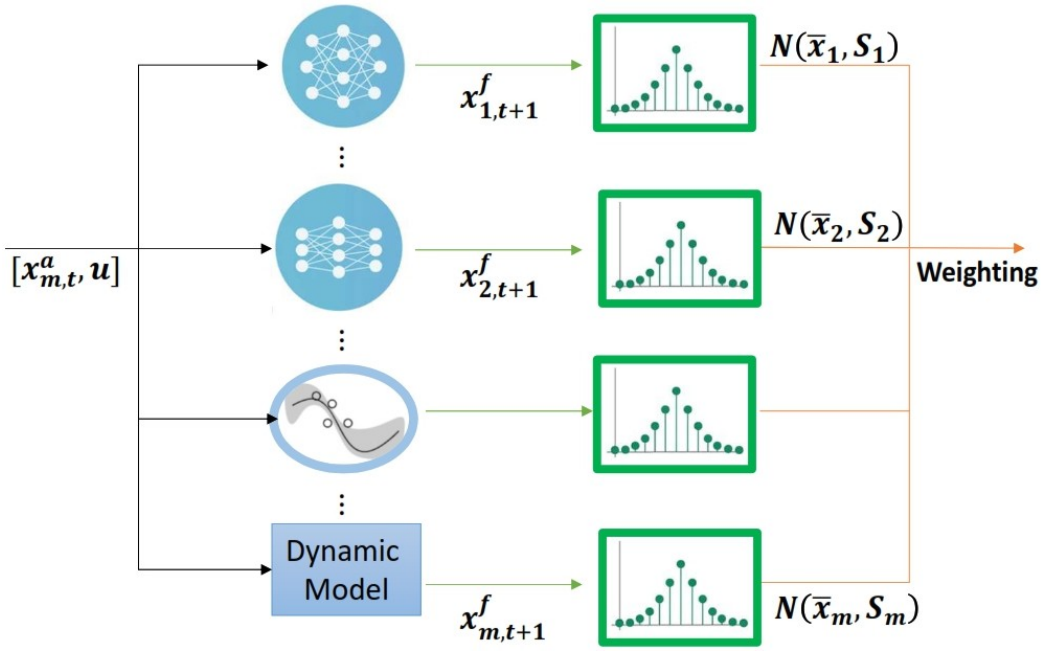


Figure 5.1: Schematic of Method 2: Ensemble-Based Covariance Propagation.

$$\mathbf{x}^f_{m,t+1} = \bar{\mathbf{x}}^f_{m,t+1} = \frac{1}{N_m} \sum_{i=1}^{N_m} (\mathbf{x}^f_{m,t+1})_i \tag{5.24}$$

$$\mathbf{P}^f_{m,t+1} = \frac{1}{N_m - 1} \sum_{i=1}^{N_m} \left[(\mathbf{x}^f_{m,t+1})_i - \bar{\mathbf{x}}^f_{m,t+1} \right] \left[(\mathbf{x}^f_{m,t+1})_i - \bar{\mathbf{x}}^f_{m,t+1} \right]^T \tag{5.25}$$

The factor $(N_m - 1)$ in the covariance estimation provides an unbiased estimate. For small ensemble sizes, this correction is important to avoid underestimating uncertainty.

Figure 5.1 illustrates the ensemble-based approach, showing how perturbations in the initial state lead to a distribution of predicted states that captures both the mean prediction and its uncertainty.

5.4.5 Advanced Ensemble Techniques

Adaptive Sampling Covariance

To improve the accuracy of uncertainty estimates over time, we implement an adaptive scheme for updating the sampling covariance based on innovation statistics. The innovation sequence provides direct feedback on whether the uncertainty estimates are well-calibrated:

$$\mathbf{d}_{m,t} = \mathbf{y}_t - \mathbf{H}_t \mathbf{x}_{m,t}^f \tag{5.26}$$

$$\mathbf{S}_{t+1} = (1-\gamma)\mathbf{S}_t + \gamma \left[\mathbf{d}_{m,t}\mathbf{d}_{m,t}^T - \mathbf{H}_t \mathbf{P}_{m,t}^f \mathbf{H}_t^T - \mathbf{R}_t \right] \tag{5.27}$$

where $\gamma \in (0, 1)$ is an adaptation rate that controls the balance between stability and responsiveness. The term in brackets represents the discrepancy between observed and expected innovation covariance—if positive, the model is underconfident, and if negative, it is overconfident.

Ensemble Size Selection

The choice of ensemble size N_m involves a fundamental trade-off between computational cost and accuracy. Theoretical results from ensemble Kalman filter literature suggest that the required ensemble size depends on several factors:

- **System dimension**: Higher-dimensional systems generally require larger ensembles to adequately sample the uncertainty space
- **Nonlinearity**: Strongly nonlinear dynamics may create complex uncertainty structures requiring more samples
- **Noise characteristics**: Systems with non-Gaussian noise benefit from larger ensembles

Empirical guidelines based on extensive testing suggest:

- For well-conditioned problems with moderate nonlinearity: $N_m = 2n$ to $3n$ where n is the state dimension

- For ill-conditioned or highly nonlinear problems: $N_m = 5n$ to $10n$
- Adaptive schemes can adjust N_m based on metrics such as the effective sample size or the rank of the ensemble covariance

5.5 Consensus Multi-Model Fusion

Once predictions and their associated covariances from all models are obtained (using either Method 1 or Method 2), we proceed with the consensus fusion step. This procedure combines predictions from different models based on their relative uncertainties before incorporating observation data. The consensus approach ensures that models with higher confidence (lower uncertainty) have greater influence on the final estimate, while still maintaining contributions from all models to ensure robustness.

5.5.1 Iterative Fusion Procedure

We adopt an iterative approach to fuse model predictions, which offers computational advantages and allows for sequential incorporation of models. This approach is particularly beneficial when models have different computational requirements or when some models may occasionally be unavailable. The procedure begins by treating the first model's prediction as the initial consensus:

$$\mathbf{x}^f_{1:1,t} = \mathbf{x}^f_{1,t} \tag{5.28}$$

$$\mathbf{P}^f_{1:1,t} = \mathbf{P}^f_{1,t} \tag{5.29}$$

For each subsequent model $m = 2, \ldots, M$, we compute the incorporation gain that determines how much the new model's information should influence the current consensus:

$$\mathbf{K}_{m,t} = \mathbf{P}^f_{1:m-1,t}\mathbf{T}_m^T \left(\mathbf{T}_m \mathbf{P}^f_{1:m-1,t}\mathbf{T}_m^T + \mathbf{P}^f_{m,t}\right)^{-1} \tag{5.30}$$

This gain matrix has an intuitive interpretation: it is large when the current consensus has high uncertainty (large $\mathbf{P}^f_{1:m-1,t}$) and the new model has low uncertainty (small $\mathbf{P}^f_{m,t}$),

meaning we should trust the new model more. Conversely, if the consensus already has low uncertainty, the gain is small, preserving the existing estimate.

The consensus state and covariance are then updated using a structure analogous to the Kalman filter measurement update:

$$\mathbf{x}^f_{1:m,t} = \mathbf{x}^f_{1:m-1,t} + \mathbf{K}_{m,t}\left(\mathbf{x}^f_{m,t} - \mathbf{T}_m\mathbf{x}^f_{1:m-1,t}\right) \tag{5.31}$$

$$\mathbf{P}^f_{1:m,t} = \left(\mathbf{I} - \mathbf{K}_{m,t}\mathbf{T}_m\right)\mathbf{P}^f_{1:m-1,t} \tag{5.32}$$

The term $(\mathbf{x}^f_{m,t} - \mathbf{T}_m\mathbf{x}^f_{1:m-1,t})$ represents the "innovation" from model m—the new information it provides beyond what the current consensus predicts. The transformation operator $\mathbf{T}_m$ ensures proper alignment when models use different state representations.

5.5.2 Observation Update

After incorporating all models, the consensus prediction represents the best available estimate given the model predictions. This consensus is then updated using the observation data through the standard Kalman filter measurement update:

$$\mathbf{K}_t = \mathbf{P}^f_{1:M,t}\mathbf{H}_t^T\left(\mathbf{H}_t\mathbf{P}^f_{1:M,t}\mathbf{H}_t^T + \mathbf{R}_t\right)^{-1} \tag{5.33}$$

$$\mathbf{x}^a_t = \mathbf{x}^f_{1:M,t} + \mathbf{K}_t\left(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}^f_{1:M,t}\right) \tag{5.34}$$

$$\mathbf{P}^a_t = \left(\mathbf{I} - \mathbf{K}_t\mathbf{H}_t\right)\mathbf{P}^f_{1:M,t} \tag{5.35}$$

where $\mathbf{y}_t$ is the measurement vector, $\mathbf{H}_t$ is the measurement matrix that relates states to measurements, and $\mathbf{R}_t$ is the measurement noise covariance. The Kalman gain $\mathbf{K}_t$ optimally balances the model predictions with the measurement information based on their relative uncertainties.

5.5.3 Implementation and Method Comparison

Our framework offers two complementary approaches to covariance propagation for data-driven models, each with distinct advantages. Method 1 employs Koopman operator theory to find linearized representations of nonlinear data-driven models, enabling analytical

covariance propagation. While requiring offline training, this approach is particularly effective when the system has known structural properties to guide the lifting process. Conversely, Method 2 offers a more straightforward approach by treating all models within a unified ensemble framework. It avoids pretraining requirements and adapts naturally as new data becomes available, though it may demand larger ensemble sizes for reliable covariance estimates, increasing computational demands.

When implementing either approach, several critical considerations emerge. The model set should capture diverse aspects of system dynamics without redundancy, while computational resources must be carefully balanced—selecting appropriate lifted space dimensions for Method 1 or ensemble sizes for Method 2. The framework's effectiveness depends on proper state space transformations when models use different representations, and for Method 2, initialization of sampling covariance based on prior knowledge or historical errors is essential. Additionally, adaptive weighting of models based on their demonstrated prediction accuracy enables dynamic adjustment to different operating regimes.

A fundamental component across both methods is proper model error estimation. We employ innovation statistics to estimate model error covariances:

$$\mathbf{d}_{m,t} = \mathbf{y}_t - \mathbf{H}_t \mathbf{x}^f_{m,t} \tag{5.36}$$

$$\mathbb{E}[\mathbf{d}_{m,t}\mathbf{d}^T_{m,t}] = \mathbf{H}_t \mathbf{P}^f_{m,t} \mathbf{H}^T_t + \mathbf{R}_t \tag{5.37}$$

This innovation-based approach enables propagation of error statistics back to the model space, ensuring appropriate calibration of relative uncertainties between different model types—a critical factor for the effective fusion of heterogeneous models.

5.5.4 Algorithmic Summary

Algorithm 1 presents a comprehensive implementation of our proposed Multi-Model Ensemble Kalman Filter framework with heterogeneous model integration. This algorithm encapsulates the entire workflow described in the previous sections, including both covariance propagation methods, the iterative consensus fusion procedure, and the observation update step. The algorithm demonstrates how our approach seamlessly integrates data-driven and physics-based models within a unified estimation framework while accounting for their unique characteristics in uncertainty propagation. By following this algorithmic implementation, practitioners can effectively combine the complementary strengths of diverse modeling paradigms for enhanced state estimation performance across varying operating conditions.



Figure 5.2: Electric all-wheel-drive Equinox test vehicle used for experimental validation. The vehicle features independent wheel motors, comprehensive sensor suite, and robotic steering system enabling repeatable test maneuvers under extreme conditions.

5.6 Experimental Validation

5.6.1 Experimental Setup

The proposed consensus multi-model Kalman filter framework was evaluated using an electric all-wheel-drive Chevrolet Equinox vehicle, as shown in Figure 5.2. This testbed provided an ideal platform for validating our approach under diverse operating conditions, particularly those challenging scenarios where traditional single-model approaches fail.

The test vehicle represents a modern electric vehicle platform with advanced actuation and sensing capabilities. Table 5.1 presents the key specifications that influence the vehicle

Table 5.1: Specifications of the electric all-wheel-drive test vehicle

Parameter	Value
Vehicle mass (M)	2271 kg
Vehicle moment of inertia (I_z)	4600 kg $\cdot$ m^2
Distance from front axle to CG (a)	1.42 m
Distance from rear axle to CG (b)	1.43 m
Tire effective radius (R_w)	0.347 m
Tire cornering stiffness (C_α)	83700 N/rad
Wheel moment of inertia (I_w)	1.7 kg $\cdot$ m^2
Steering ratio (SR)	18:1

dynamics and estimation challenges.

The test vehicle is equipped with a robotic steering system. The vehicle's powertrain system has an electric motor for each wheel, providing independent control over each wheel's torque. The sensing suite includes a 6-axis IMU sensor that measures yaw rate and longitudinal and lateral accelerations. A 6-axis GPS sensor (RT2500) provides the vehicle's longitudinal and lateral velocities. The vehicle is also equipped with a steering wheel sensor, wheel speed sensors, and wheel torque sensors.

During each maneuver, data from the Inertial Measurement Unit (IMU), Global-Positioning-System (GPS), and wheel sensors were collected. The signal measurements were sampled at a frequency of 200 Hz for all maneuvers. To rigorously evaluate the algorithm's performance in challenging conditions, extensive testing was conducted in extreme winter environments with temperatures as low as -28°C on ice and snow surfaces, in addition to standard tests on dry and wet surfaces. The test protocol included a diverse range of demanding maneuvers: vehicle launches on varying friction surfaces, sustained acceleration runs, pulsed torque inputs, high sideslip scenarios, low-speed turning tests in both directions, and high-speed sine-steer maneuvers at 70 km/h. Model Predictive Control (MPC) was implemented with the estimation algorithms in the control loop, with various traction control settings tested to intentionally induce instability and push the vehicle beyond its linear handling regime. This comprehensive experimental design created scenarios where traditional single-model approaches would typically struggle, providing an ideal testbed for evaluating the robustness of the consensus multi-model framework under real-world challenging conditions.

5.6.2 State Estimation Results

In this section, we present and analyze the performance of our two methodologies using experimental data collected from the test vehicle. The test dataset consists of 19 distinct maneuvers, with 18 maneuvers used for training the neural networks and one unseen maneuver reserved for validation and performance evaluation. Figure 5.3 depicts the ground truth data from this unseen test maneuver, which provides the benchmark for assessing estimation accuracy.

For the first methodology, the Koopman Linearization for Data-Driven Models, we implemented two neural networks(referred to as Net 1 and Net 2) and divided the time series data into equal-sized batches for training. The neural network architecture employs a dual feedforward configuration with a three-layer structure. Each network's input layer contains 8 neurons, capturing critical vehicle states including lateral and longitudinal velocities, steering angle, yaw rate, and wheel speeds. Two hidden layers with 25 and 15 neurons respectively provide sufficient capacity to model nonlinear vehicle dynamics without overfitting. The output layer consists of 6 neurons that predict state variables for the subsequent time step.

Both networks were trained using the *Levenberg* backpropagation algorithm, which offers superior convergence properties for medium-sized networks compared to traditional gradient descent methods. The networks were trained independently on distinct datasets with normalized inputs and outputs, enhancing generalization capabilities while maintaining numerical stability. A critical aspect of our approach is that unmeasurable states like lateral velocity are only used during network training with observed data. During testing with unseen data, these states are treated as completely unobservable, which addresses the primary challenge of state estimation in vehicle dynamics applications.

The implementation further leverages Koopman operator theory to provide a principled framework for covariance propagation within the Multi-Model Kalman Filter. We employed Random Fourier Features (RFF) as lifting functions to transform the state and control inputs into an expanded observable space, where bilinear dynamics are converted to linear time-varying (LTV) systems through strategic tensor products. To enhance stability, we implemented a blending technique between the Koopman-predicted lifted state and direct state embedding with parameter $\alpha = 0.8$, effectively mitigating potential divergence issues. The covariance matrices underwent careful conditioning procedures to ensure numerical stability and positive-definiteness throughout the estimation process. The performance results of this method are presented in Figure 5.4, which demonstrates excellent estimation accuracy. This hybrid methodology—utilizing neural networks for state prediction while leveraging Koopman operators for covariance propagation—represents a

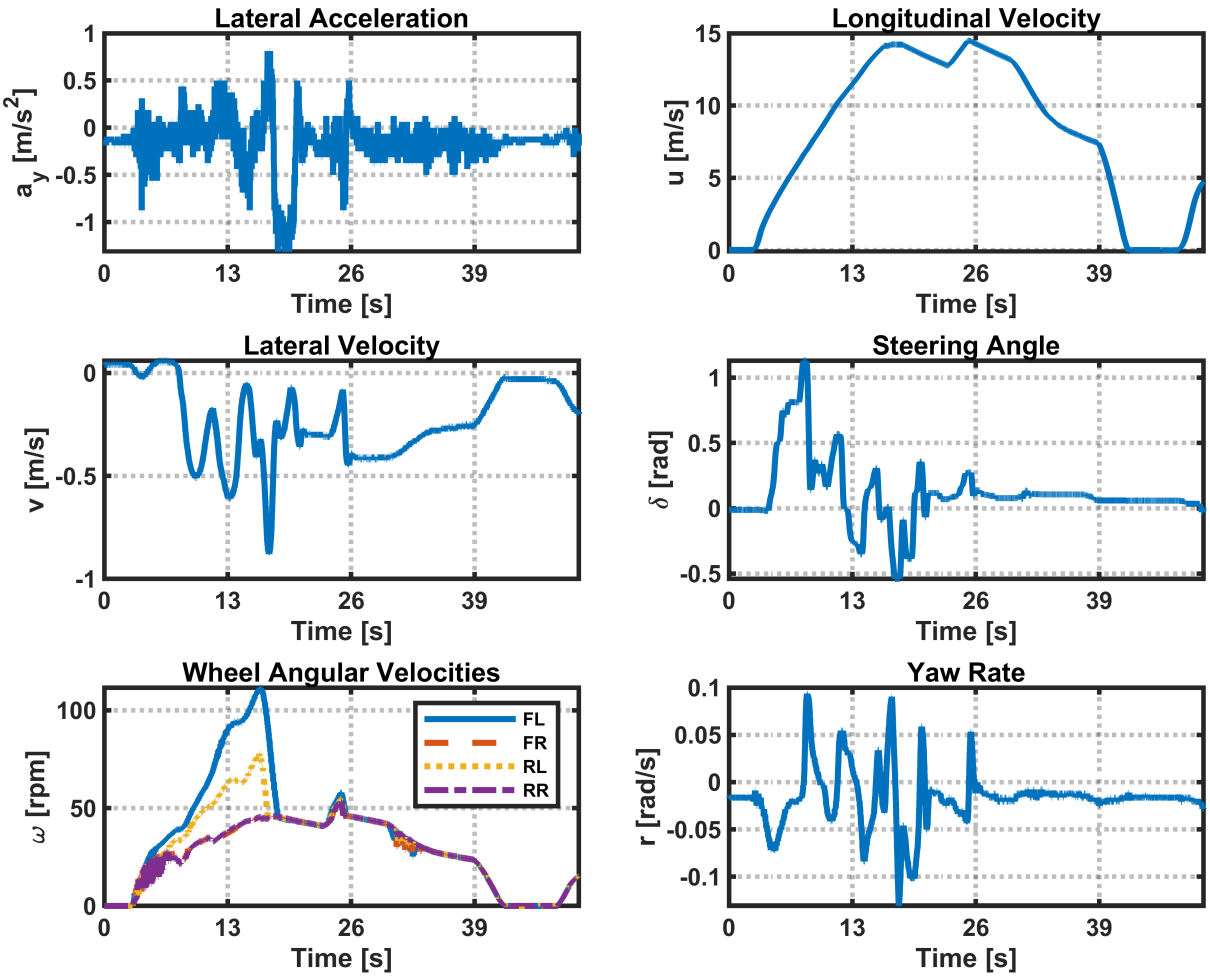


Figure 5.3: Ground truth data from the unseen test maneuver on icy road. This dataset serves as the benchmark for evaluating estimation performance.

sophisticated fusion of data-driven prediction with theoretically grounded uncertainty propagation, particularly advantageous for estimating unmeasurable states like lateral velocity in complex vehicle dynamics applications.

Turning to our second methodology, the Ensemble-Based Covariance Propagation, we applied this approach to the same dataset to ensure statistical validity and consistent evaluation across methods. We maintained the first neural network trained in the previous approach and replaced the second network with a bicycle model representing the physical dynamics of the vehicle. The results and performance of this method are illustrated in Figure 5.5. Notably, during occasions with higher slip angles and nonlinear tire behavior, the bicycle model demonstrated limitations in capturing the complex vehicle dynamics. In these scenarios, our framework automatically assigned greater weight to the neural network, which demonstrated superior estimation performance. This automatic weighting adjustment highlights the adaptive nature of our consensus fusion approach, which dynamically balances the contributions of heterogeneous models based on their demonstrated reliability in current operating conditions.

Considering the results of both methodologies, it is evident that both approaches are validated and demonstrate excellent performance in vehicle state estimation. The consensus multi-model framework successfully integrates the complementary strengths of physics-based and data-driven models, providing robust estimation across diverse operating conditions. This is particularly valuable for critical vehicle states that cannot be directly measured but are essential for advanced vehicle control systems.

5.7 Summary and Remarks

This chapter presented a novel consensus multi-model Kalman filter framework that addresses the fundamental challenge of integrating heterogeneous model types—both physics-based and data-driven—for robust state estimation in complex dynamical systems. The framework leverages the complementary strengths of different modeling paradigms while mitigating their individual weaknesses through a principled consensus fusion approach.

Two distinct methodologies were developed to enable consistent covariance propagation for data-driven models within the Kalman filter framework. Method 1 employs Koopman operator theory to lift nonlinear data-driven models into a higher-dimensional space where bilinear dynamics can be converted to linear time-varying systems. This approach provides analytical covariance propagation through learned system matrices $\mathbf{A}_K$, $\mathbf{B}_K$, and $\mathbf{H}_K$, enabling seamless integration with traditional physics-based models. The method

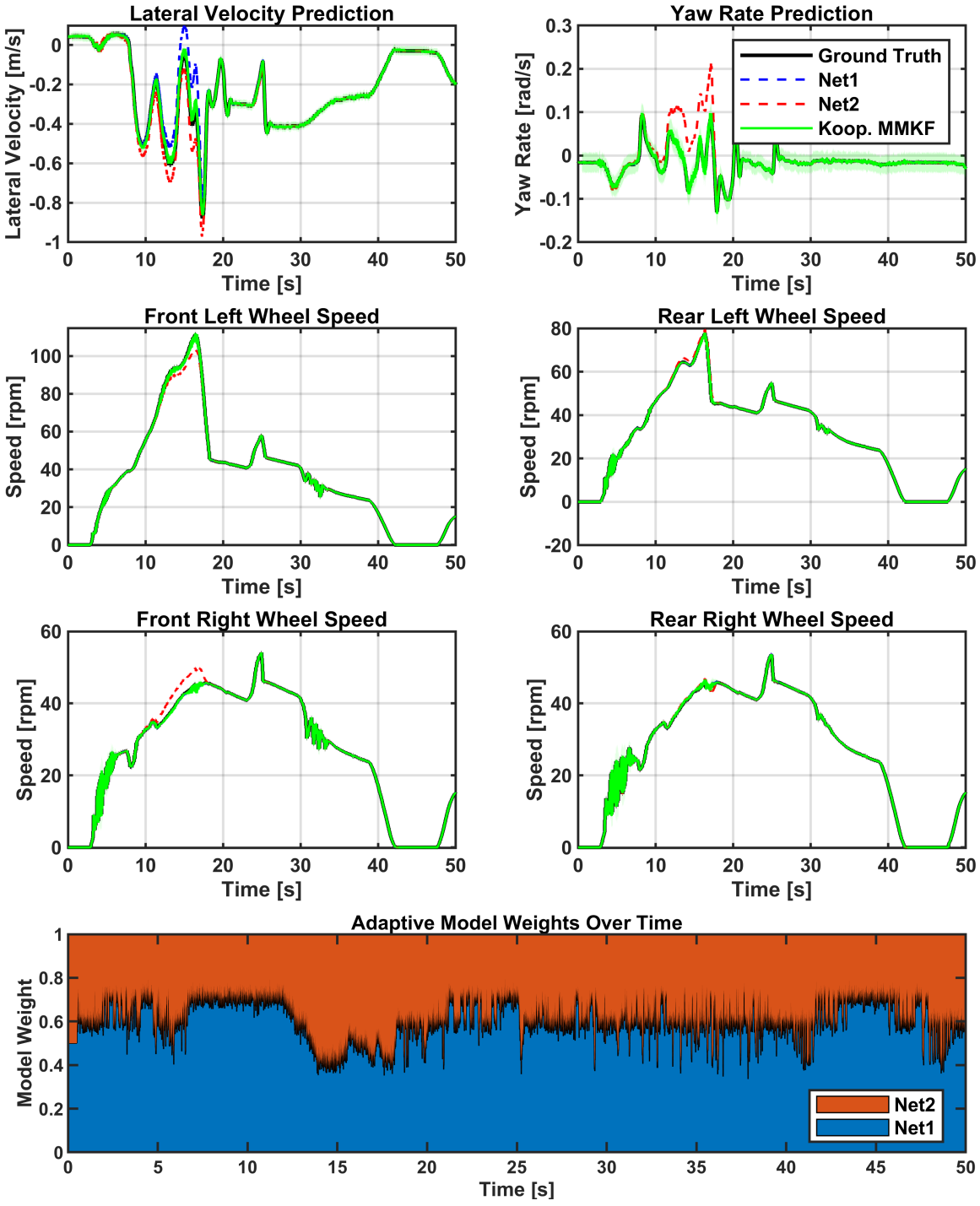


Figure 5.4: Performance of Method 1: Koopman Linearization for Data-Driven Models.

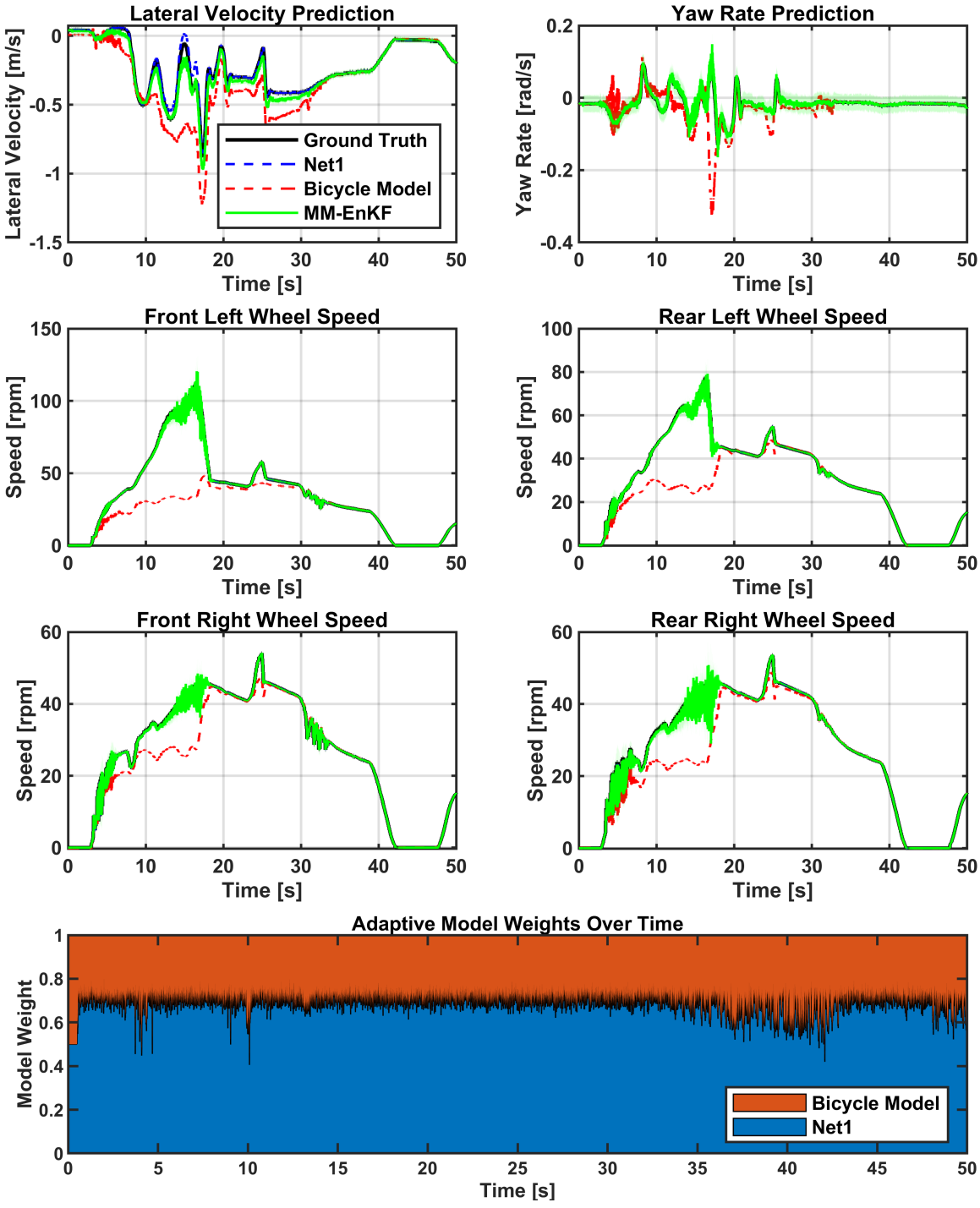


Figure 5.5: Performance of Method 2: Ensemble-Based Covariance Propagation.

requires offline training but offers computationally efficient online operation through the linearization process.

Method 2 adopts an ensemble-based approach that treats all models uniformly within a statistical framework. By propagating ensembles of perturbed states through each model and computing empirical statistics, this method naturally captures non-Gaussian uncertainty propagation without requiring model-specific modifications. The approach offers greater flexibility and adaptability, particularly valuable when online adaptation is desired or when pretraining is impractical.

The consensus fusion procedure implements an iterative incorporation scheme that dynamically weighs model contributions based on their demonstrated reliability. Through the incorporation gain matrix $\mathbf{K}_{m,t}$, models with lower uncertainty receive greater influence in the consensus estimate, while the framework maintains robustness by preserving contributions from all models. This adaptive weighting mechanism enables the system to automatically leverage the most appropriate models for current operating conditions.

Experimental validation on an electric all-wheel-drive Equinox vehicle demonstrated the effectiveness of both methodologies under challenging conditions including extreme winter environments, high sideslip scenarios, and nonlinear tire behavior. The results revealed that the framework successfully adapts to varying operating regimes—automatically increasing reliance on data-driven models during extreme maneuvers where physics-based assumptions break down, while leveraging the interpretability of physics-based models during nominal conditions.

The proposed framework represents a significant advancement in multi-model estimation by providing a unified approach to handling heterogeneous model types. The ability to meaningfully combine predictions from models with fundamentally different representations and uncertainty characteristics opens new possibilities for robust estimation in safety-critical applications. Furthermore, the framework's modular design allows for straightforward extension to incorporate additional models or alternative uncertainty quantification methods, making it a versatile tool for addressing the increasing complexity of modern engineering systems.

Algorithm 1 Multi-Model Ensemble Kalman Filter with Heterogeneous Model Integration

Initialization

Initialize analysis state $\mathbf{x}^a_{m,0}$ and covariance $\mathbf{P}^a_{m,0}$ for each model $m = 1, 2, \ldots, M$

for each time step $t = 0, 1, 2, \ldots$ **do**

1. Model-specific forecasting

  for each model $m = 1, 2, \ldots, M$ **do**

    if Method 1 is used **then**

      if model m is data-driven **then**

        Predict state: $\mathbf{x}^f_{m,t+1} = f_m(\mathbf{x}^a_{m,t})$

        Derive Koopman linearization $\mathbf{A}_{K,t}$

        Update covariance: $\mathbf{P}^f_{m,t+1} = \mathbf{A}_{K,t}\mathbf{P}^a_{m,t}\mathbf{A}^T_{K,t} + \mathbf{Q}_{m,t}$

      else ▷ Physics-based model with Method 1

        Predict state: $\mathbf{x}^f_{m,t+1} = f_m(\mathbf{x}^a_{m,t})$

        Compute Jacobian $\mathbf{F}_{m,t}$

        Update covariance: $\mathbf{P}^f_{m,t+1} = \mathbf{F}_{m,t}\mathbf{P}^a_{m,t}\mathbf{F}^T_{m,t} + \mathbf{Q}_{m,t}$

      end if

    else ▷ Method 2 is used (Ensemble-based approach)

      Generate ensemble: $\{(\mathbf{x}^a_{m,t})_i\}_{i=1}^{N_m}$ where $(\mathbf{x}^a_{m,t})_i = \mathbf{x}^a_{m,t} + \boldsymbol{\eta}_i$

      Propagate each ensemble member: $(\mathbf{x}^f_{m,t+1})_i = f_m((\mathbf{x}^a_{m,t})_i)$ for $i = 1, 2, \ldots, N_m$

      Compute ensemble mean: $\mathbf{x}^f_{m,t+1} = \frac{1}{N_m}\sum_{i=1}^{N_m}(\mathbf{x}^f_{m,t+1})_i$

      Compute ensemble covariance:
      $\mathbf{P}^f_{m,t+1} = \frac{1}{N_m-1}\sum_{i=1}^{N_m}((\mathbf{x}^f_{m,t+1})_i - \mathbf{x}^f_{m,t+1})((\mathbf{x}^f_{m,t+1})_i - \mathbf{x}^f_{m,t+1})^T$

    end if

  end for

2. Consensus fusion of model predictions

  Initialize consensus with first model: $\mathbf{x}^f_{1:1,t+1} = \mathbf{x}^f_{1,t+1}$, $\mathbf{P}^f_{1:1,t+1} = \mathbf{P}^f_{1,t+1}$

  for each additional model $m = 2, 3, \ldots, M$ **do**

    Compute incorporation gain:
    $\mathbf{K}_{m,t+1} = \mathbf{P}^f_{1:m-1,t+1}\mathbf{T}^T_m(\mathbf{T}_m\mathbf{P}^f_{1:m-1,t+1}\mathbf{T}^T_m + \mathbf{P}^f_{m,t+1})^{-1}$

    Update consensus state:
    $\mathbf{x}^f_{1:m,t+1} = \mathbf{x}^f_{1:m-1,t+1} + \mathbf{K}_{m,t+1}(\mathbf{x}^f_{m,t+1} - \mathbf{T}_m\mathbf{x}^f_{1:m-1,t+1})$

    Update consensus covariance:
    $\mathbf{P}^f_{1:m,t+1} = (\mathbf{I} - \mathbf{K}_{m,t+1}\mathbf{T}_m)\mathbf{P}^f_{1:m-1,t+1}$

  end for

3. Measurement update

  Obtain measurement $\mathbf{y}_{t+1}$

  Compute Kalman gain:
  $\mathbf{K}_{t+1} = \mathbf{P}^f_{1:M,t+1}\mathbf{H}^T_{t+1}(\mathbf{H}_{t+1}\mathbf{P}^f_{1:M,t+1}\mathbf{H}^T_{t+1} + \mathbf{R}_{t+1})^{-1}$

  Update state with measurement:
  $\mathbf{x}^a_{t+1} = \mathbf{x}^f_{1:M,t+1} + \mathbf{K}_{t+1}(\mathbf{y}_{t+1} - \mathbf{H}_{t+1}\mathbf{x}^f_{1:M,t+1})$

  Update covariance:
  $\mathbf{P}^a_{t+1} = (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H}_{t+1})\mathbf{P}^f_{1:M,t+1}$

4. Feedback to individual models

  for each model $m = 1, 2, \ldots, M$ **do**

    Map consensus state to model space: $\mathbf{x}^a_{m,t+1} = \mathbf{T}_m\mathbf{x}^a_{t+1}$

    Map consensus covariance to model space: $\mathbf{P}^a_{m,t+1} = \mathbf{T}_m\mathbf{P}^a_{t+1}\mathbf{T}^T_m$

  end for

end for

Chapter 6

Conclusions and Future Directions

6.1 Conclusions

This thesis has revealed that the computational costs of multi-model adaptive control stems not from fundamental limitations but from conventional implementation choices that hide elegant solutions. Through investigation across three interconnected research areas, it has been demonstrated that intelligent reformulation can transform these systems from computationally prohibitive to practically deployable.

The research began with a critical observation: traditional multi-model approaches exhaustively enumerate all possible parameter combinations, creating massive redundancy where most models contribute negligibly to uncertainty coverage. This practice, rooted in conservative design philosophy, wastes computational resources on models that provide no meaningful benefit. By recognizing that effective uncertainty representation depends on strategic model placement rather than exhaustive enumeration, order-of-magnitude reductions in computational requirements are achieved while maintaining robust performance. This finding challenges decades of accepted practice and demonstrates the feasibility of real-time implementation for previously intractable control schemes.

A deeper investigation into the nature of multi-model complexity led to another discovery. The exponential growth in model requirements with increasing system dimension—long considered an inevitable curse—results from working in coordinates that hide the underlying structure. When transformed to appropriate canonical forms, high-dimensional uncertainty spaces often collapse dramatically, revealing that apparent complexity was merely an artifact of poor representation. This insight extends beyond computational effi-

ciency to suggest that many control problems considered inherently complex may possess hidden simplicities that can be found through proper mathematical formulation.

The integration of physics-based and data-driven models presented a different challenge, coming from their fundamentally different uncertainty characterizations and failure modes. Traditional approaches treat these as separate paradigms requiring distinct frameworks, missing opportunities for synergy. By developing principled methods for consistent uncertainty quantification across heterogeneous model types, each paradigm is enabled to compensate for the other's weaknesses. Physics-based models provide interpretability and graceful degradation, while data-driven models capture complex behaviors beyond first-principles understanding. Their fusion creates robustness exceeding either approach alone, particularly in extreme conditions where individual models fail catastrophically.

These advances come together to reshape the understanding of multi-model adaptive control. Where conventional wisdom accepts computational burden as the price of robustness, this work demonstrates an alternative: achieving robustness through intelligent complexity reduction rather than brute-force computation. The resulting framework maintains theoretical guarantees while enabling practical implementation on resource-constrained embedded systems.

The implications apply to many domains previously limited by computational constraints. Autonomous vehicles navigating unpredictable environments can now employ sophisticated multi-model estimation without overwhelming onboard processors. Power systems integrating renewable sources can adapt to rapid fluctuations using comprehensive uncertainty models. Robotic systems can maintain performance across diverse operating conditions without pre-programming every scenario. These applications transition from theoretical possibilities to practical realities.

Beyond specific technical contributions, this research offers a broader perspective on tackling computational barriers in control theory. Many accepted limitations may be artifacts of traditional formulations rather than fundamental constraints. By questioning basic assumptions about how complex systems must be structured, opportunities for dramatic simplification without sacrificing performance are uncovered. This approach—viewing computational complexity as a design variable rather than a fixed constraint—suggests that similar breakthroughs may be possible in other areas where theory has outpaced implementation.

The synthesis achieved here demonstrates that combining geometric insight, coordinate transformation, and heterogeneous integration creates capabilities beyond any single approach. Each element addresses specific limitations while contributing to a unified framework that transforms multi-model control from elegant theory to deployable technology.

In an era demanding increasingly sophisticated autonomous systems, such transformation becomes essential.

Ultimately, this thesis establishes that longstanding computational barriers need not limit the deployment of theoretically superior control methods. Through creative reformulation and synthesis of complementary approaches, robust adaptive control can be achieved within practical constraints. As autonomous systems become increasingly prevalent and operate in ever more challenging environments, the ability to implement sophisticated multi-model control efficiently becomes not just advantageous but essential. This work provides both the theoretical foundations and practical tools to meet that challenge, marking a definitive transition in multi-model adaptive control from academic curiosity to engineering reality.

6.2 Future Work

The theoretical and practical advances presented in this thesis open numerous avenues for future research that promise to further expand the capabilities and applications of multi-model adaptive control systems. These research directions span theoretical extensions, algorithmic enhancements, and practical implementation considerations.

i. **Extension to Nonlinear Systems**

   The extension to nonlinear systems represents perhaps the most significant theoretical challenge and opportunity. While the current frameworks primarily address linear parameter-varying systems, future work should explore systematic approaches for identifying canonical representations of nonlinear systems that exhibit analogous dimension reduction properties to those discovered for linear systems. This could involve developing nonlinear coordinate transformations that preserve the essential uncertainty structure while simplifying the model space topology. Research into differential geometric approaches for characterizing nonlinear system manifolds could reveal natural coordinates where multi-model complexity is minimized. Additionally, investigating the extension of Koopman operator theory to identify finite-dimensional embeddings of nonlinear dynamics within the multi-model framework presents a promising theoretical direction.

ii. **Advanced Machine Learning Integration**

The integration of modern machine learning techniques with multi-model control frameworks offers particularly rich opportunities for advancement. Deep learning architectures could be developed to automatically discover coordinate transformations that minimize the required number of models, potentially using physics-informed neural networks with specialized loss functions that encode coverage requirements while respecting system dynamics. The Koopman linearization approach could be enhanced through reinforcement learning algorithms that adapt lifting functions online based on observed system behavior, dynamically balancing approximation accuracy with computational cost. Neural architecture search could automate the design of data-driven models within the ensemble, optimizing network architectures specifically for complementarity with physics-based models rather than standalone accuracy. Furthermore, investigating meta-learning approaches that enable rapid adaptation of the multi-model framework to new domains or operating conditions could significantly enhance practical deployment capabilities.

iii. **Bayesian Model Confidence Quantification**

Incorporating Bayesian techniques to quantify confidence in individual models and adaptively weigh their influence offers a principled framework for uncertainty management within the multi-model ensemble. Future work should develop hierarchical Bayesian models that capture both parametric uncertainty within individual models and structural uncertainty across the model set. This could involve implementing variational inference techniques to efficiently compute posterior distributions over model parameters and mixing weights, enabling real-time adaptation of the ensemble composition. Research into Bayesian neural networks within the data-driven component could provide principled uncertainty quantification that accounts for both aleatoric and epistemic uncertainty. The development of Bayesian optimization approaches for online model structure adaptation could enable the framework to autonomously refine its architecture based on observed performance. Furthermore, investigating the use of Gaussian processes as surrogate models for expensive physics-based simulations could enable efficient uncertainty propagation while maintaining interpretability.

iv. **Multi-Agent and Networked Systems**

Extending the multi-model framework to multi-agent and networked control systems presents significant theoretical and practical challenges. Future research should explore distributed multi-model architectures where individual agents maintain local model ensembles while sharing information to improve collective performance. This

could involve developing consensus algorithms for model agreement across the network while respecting communication constraints and privacy requirements. Research into game-theoretic approaches for multi-agent multi-model control could address scenarios where agents have conflicting objectives or incomplete information sharing. The investigation of network-aware model selection strategies that account for communication delays and packet losses could enhance robustness in networked implementations. Additionally, exploring hierarchical multi-model architectures where high-level coordinators manage lower-level multi-model controllers could enable scalable control of large-scale systems.

References

[1] Lei Liu, Tingting Gao, Yan-Jun Liu, Shaocheng Tong, CL Philip Chen, and Lei Ma. Time-varying iblfs-based adaptive control of uncertain nonlinear systems with full state constraints. *Automatica*, 129:109595, 2021.

[2] Petros A Ioannou and Petar V Kokotovic. Instability analysis and improvement of robustness of adaptive control. *Automatica*, 20(5):583–594, 1984.

[3] John S Reed and Petros A Ioannou. Instability analysis and robust adaptive control of robotic manipulators. In *Proceedings of the 27th IEEE Conference on Decision and Control*, pages 1607–1612. IEEE, 1988.

[4] Roderick Murray-Smith and T Johansen. *Multiple model approaches to nonlinear modelling and control*. CRC press, 2020.

[5] Zhuo Han and Kumpati S Narendra. New concepts in adaptive control using multiple models. *IEEE Transactions on Automatic Control*, 57(1):78–89, 2011.

[6] Joao Hespanha, Daniel Liberzon, A Stephen Morse, Brian DO Anderson, Thomas S Brinsmead, and Franky De Bruyne. Multiple model adaptive control. part 2: switching. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 11(5):479–496, 2001.

[7] Kumpati S Narendra and Jeyendran Balakrishnan. Adaptive control using multiple models. *IEEE transactions on automatic control*, 42(2):171–187, 1997.

[8] KS Narendra, Xiang Cheng, et al. Adaptive control using multiple models, switching and tuning. *International journal of adaptive control and signal processing*, 17(2):87–102, 2003.

[9] M Kuipers and P Ioannou. Multiple model adaptive control with mixing. *IEEE transactions on automatic control*, 55(8):1822–1836, 2010.

[10] Halit Zengin, Nursefa Zengin, Baris Fidan, and Amir Khajepour. Blending based multiple-model adaptive control of multivariable systems with application to lateral vehicle motion control. *European Journal of Control*, 58:1–10, 2021.

[11] Halit Zengin, Nursefa Zengin, Baris Fidan, and Amir Khajepour. Blending based multiple-model adaptive control for multivariable systems and application to lateral vehicle dynamics. In *2019 18th European Control Conference (ECC)*, pages 2957–2962. IEEE, 2019.

[12] K Narendra and Z Han. A new approach to adaptive control using multiple models. *International Journal of adaptive Control and signal processing*, 26(8):778–799, 2012.

[13] K Narendra. Hierarchical adaptive control of rapidly time-varying systems using multiple models. In *Control of Complex Systems*, pages 33–66. Elsevier, 2016.

[14] KS Narendra, Y Wang, and W Chen. The rationale for second level adaptation. *arXiv preprint arXiv:1510.04989*, 2015.

[15] M Athans et al. The stochastic control of the f-8c aircraft using a multiple model adaptive control (mmac) method–part i: Equilibrium flight. *IEEE Transactions on Automatic Control*, 22(5):768–780, 1977.

[16] DG Lainiotis. Partitioning: A unifying framework for adaptive systems, i: Estimation. *Proceedings of the IEEE*, 64(8):1126–1143, 1976.

[17] Bengt Mårtensson. Adaptive stabilization. 1986.

[18] Minyue Fu and BROSS Barmish. Adaptive stabilization of linear systems via switching control. *IEEE transactions on Automatic Control*, 31(12):1097–1103, 1986.

[19] Daniel E Miller and Edward J Davison. An adaptive controller which provides lyapunov stability. In *Proceedings of the 27th IEEE Conference on Decision and Control*, pages 1934–1940. IEEE, 1988.

[20] Richard H Middleton, Graham C Goodwin, David J Hill, and David Q Mayne. Design issues in adaptive control. *IEEE transactions on automatic control*, 33(1):50–58, 1988.

[21] AS Morse. Supervisory control of families of linear set-point controllers. In *Proceedings of 32nd IEEE Conference on Decision and Control*, pages 1055–1060. IEEE, 1993.

[22] A Stephen Morse. Supervisory control of families of linear set-point controllers-part i. exact matching. *IEEE transactions on Automatic Control*, 41(10):1413–1431, 1996.

[23] A Stephen Morse. Supervisory control of families of linear set-point controllers. 2. robustness. *IEEE Transactions on Automatic Control*, 42(11):1500–1515, 1997.

[24] Joao Hespanha, Daniel Liberzon, A Stephen Morse, Brian DO Anderson, Thomas S Brinsmead, and Franky De Bruyne. Multiple model adaptive control. part 2: switching. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 11(5):479–496, 2001.

[25] Joao P Hespanha, Daniel Liberzon, and A Stephen Morse. Overcoming the limitations of adaptive control by means of logic-based switching. *Systems control letters*, 49(1):49–65, 2003.

[26] Kumpati S Narendra and Jeyendran Balakrishnan. Improving transient response of adaptive control systems using multiple models and switching. *IEEE Transactions on automatic control*, 39(9):1861–1866, 1994.

[27] Kumpati S Narendra and Zhuo Han. The changing face of adaptive control: the use of multiple models. *Annual reviews in control*, 35(1):1–12, 2011.

[28] S Solmaz, M Akar, and R Shorten. Online center of gravity estimation in automotive vehicles using multiple models and switching. In *2006 9th International Conference on Control, Automation, Robotics and Vision*. IEEE, 2006.

[29] Selim Solmaz, Mehmet Akar, Robert Shorten, and Jens Kalkkuhl. Real-time multiple-model estimation of centre of gravity position in automotive vehicles. *Vehicle System Dynamics*, 46(9):763–788, 2008.

[30] Ebrahim Esmailzadeh, Avesta Goodarzi, and GR Vossoughi. Optimal yaw moment control law for improved vehicle handling. *Mechatronics*, 13(7):659–675, 2003.

[31] K Jalali et al. Development of an advanced torque vectoring control system for an electric vehicle with in-wheel motors using soft computing techniques. *SAE International Journal of Alternative Powertrains*, 2(2):261–278, 2013.

[32] Ebrahim Esmailzadeh, Avesta Goodarzi, and GR Vossoughi. Optimal yaw moment control law for improved vehicle handling. *Mechatronics*, 13(7):659–675, 2003.

[33] Mohammad Pirani, John W Simpson-Porco, and Baris Fidan. System-theoretic performance metrics for low-inertia stability of power networks. pages 5106–5111, 2017.

[34] Mohammad Pirani, Ehsan Hashemi, Baris Fidan, and John W Simpson-Porco. H performance of mechanical and power networks. *IFAC-PapersOnLine*, 50(1):5196–5201, 2017.

[35] Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1964.

[36] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[37] Reza Hajiloo, Amir Khajepour, Alireza Kasaiezadeh, Shih-Ken Chen, and Bakhtiar Litkouhi. Integrated lateral and roll stability control of multi-actuated vehicles using prioritization model predictive control. *IEEE Transactions on Vehicular Technology*, 71(8):8318–8329, 2022.

[38] Sergiy Antonov, Achim Fehn, and Andreas Kugi. Unscented kalman filter for vehicle state estimation. *Vehicle System Dynamics*, 49(9):1497–1520, 2011.

[39] Zeyu Ma, Xuewu Ji, Yunqing Zhang, and James Yang. State estimation in roll dynamics for commercial vehicles. *Vehicle system dynamics*, 55(3):313–337, 2017.

[40] JL Bentley, FP Preparata, and MG Faust. Approximation algorithms for convex hulls. *Communications of the ACM*, 25(1):64–68, 1982.

[41] AN Gamby and J Katajainen. Convex-hull algorithms: Implementation, testing, and experimentation. *Algorithms*, 11(12):195, 2018.

[42] H Sartipizadeh and TL Vincent. Computing the approximate convex hull in high dimensions. *arXiv preprint arXiv:1603.04422*, 2016.

[43] R Seidel. Convex hull computations. In *Handbook of discrete and computational geometry*, pages 687–703. Chapman and Hall/CRC, 2017.

[44] S Shahid. *Finding Optimized Bounding Boxes of Polytopes in D-dimensional Space and Their Properties in K-dimensional Projections*. Thesis, Michigan State University. Computer Science, 2014.

[45] Henning Holzmann, Oliver Nelles, Christoph Halfmann, and Rolf Isermann. Vehicle dynamics simulation based on hybrid modeling. In *1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No. 99TH8399)*, pages 1014–1019. IEEE, 1999.

[46] Yongjun Pan, Xiaobo Nie, Zhixiong Li, and Shuitao Gu. Data-driven vehicle modeling of longitudinal dynamics based on a multibody model and deep neural networks. *Measurement*, 180:109541, 2021.

[47] Stefano Melzi and Edoardo Sabbioni. On the vehicle sideslip angle estimation through neural networks: Numerical and experimental results. *Mechanical Systems and Signal Processing*, 25(6):2005–2019, 2011.

[48] Wang Wei, Bei Shaoyi, Zhang Lanchun, Zhu Kai, Wang Yongzhi, and Hang Weixing. Vehicle sideslip angle estimation based on general regression neural network. *Mathematical Problems in Engineering*, 2016(1):3107910, 2016.

[49] Angelo Bonfitto, Stefano Feraco, Andrea Tonoli, and Nicola Amati. Combined regression and classification artificial neural networks for sideslip angle estimation and road condition identification. *Vehicle system dynamics*, 58(11):1766–1787, 2020.

[50] Sebastian S James, Sean R Anderson, and Mauro Da Lio. Longitudinal vehicle dynamics: A comparison of physical and data-driven models under large-scale real-world driving conditions. *IEEE Access*, 8:73714–73729, 2020.

[51] Guillaume Devineau, Philip Polack, Florent Altché, and Fabien Moutarde. Coupled longitudinal and lateral control of a vehicle using deep learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 642–649. IEEE, 2018.

[52] Mauro Da Lio, Daniele Bortoluzzi, and Gastone Pietro Rosati Papini. Modelling longitudinal vehicle dynamics with neural networks. *Vehicle System Dynamics*, 58(11):1675–1693, 2020.

[53] Zygimantas Ziaukas, Alexander Busch, and Mark Wielitzka. Estimation of vehicle side-slip angle at varying road friction coefficients using a recurrent artificial neural network. In *2021 IEEE Conference on Control Technology and Applications (CCTA)*, pages 986–991. IEEE, 2021.

[54] Hao Chen, Shanhe Lou, and Chen Lv. Hybrid physics-data-driven online modelling: Framework, methodology and application to electric vehicles. *Mechanical Systems and Signal Processing*, 185:109791, 2023.

[55] Qin Li, Boyuan Zhang, Hongwen He, Yong Wang, Deqiang He, and Shuai Mo. A hybrid physics-data driven approach for vehicle dynamics state estimation. *Mechanical Systems and Signal Processing*, 225:112249, 2025.

[56] Jianwei Chen, Chuanqiang Yu, Yafei Wang, Zhisong Zhou, and Zhihao Liu. Hybrid modeling for vehicle lateral dynamics via agru with a dual-attention mechanism under limited data. *Control Engineering Practice*, 151:106015, 2024.

[57] Zhisong Zhou, Yafei Wang, Qinghui Ji, Daniel Wellmann, Yifan Zeng, and Chengliang Yin. A hybrid lateral dynamics model combining data-driven and physical models for vehicle control applications. *IFAC-PapersOnLine*, 54(20):617–623, 2021.

[58] Haakon Robinson, Suraj Pawar, Adil Rasheed, and Omer San. Physics guided neural networks for modelling of non-linear dynamics. *Neural Networks*, 154:333–345, 2022.

[59] Zhisong Zhou, Yafei Wang, Guofeng Zhou, Xulei Liu, Mingyu Wu, and Kunpeng Dai. Vehicle lateral dynamics-inspired hybrid model using neural network for parameter identification and error characterization. *IEEE Transactions on Vehicular Technology*, 2024.

[60] Hao Zheng, Yinong Li, Ling Zheng, and Ehsan Hashemi. Koopman-based hybrid modeling and zonotopic tube robust mpc for motion control of automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[61] Binrui Liu, Xinguang He, and Na Liu. Hybrid residual deep learning models with physical knowledge for improving plant transpiration estimation. *Computers and Electronics in Agriculture*, 212:108135, 2023.

[62] Ehsan Askari, Davide Gorgoretti, and Guillaume Crevecoeur. Hybrid modeling of multibody vehicles with partially known physics: discovering complex behaviors of tires. *Multibody System Dynamics*, pages 1–32, 2024.

[63] Xiaoqiang Sun, Houzhong Zhang, Yingfeng Cai, Shaohua Wang, and Long Chen. Hybrid modeling and predictive control of intelligent vehicle longitudinal velocity considering nonlinear tire dynamics. *Nonlinear Dynamics*, 97:1051–1066, 2019.

[64] Zhuoping Yu, Xinchen Hou, Bo Leng, and Yuyao Huang. Mass estimation method for intelligent vehicles based on fusion of machine learning and vehicle dynamic model. *Autonomous Intelligent Systems*, 2(1):4, 2022.

[65] Beatriz L Boada, Maria Jesus L Boada, and Hui Zhang. Sensor fusion based on a dual kalman filter for estimation of road irregularities and vehicle mass under static and dynamic conditions. *IEEE/ASME Transactions on Mechatronics*, 24(3):1075–1086, 2019.

[66] Liang Chen, Zhaobo Qin, Yougang Bian, Manjiang Hu, and Xiaoyan Peng. Data-driven tire-road friction estimation for electric-wheel vehicle with data category selection and uncertainty evaluation. *IEEE Transactions on Industrial Electronics*, 2024.

[67] Sadra Rafatnia and Mehdi Mirzaei. Estimation of reliable vehicle dynamic model using imu/gnss data fusion for stability controller design. *Mechanical Systems and Signal Processing*, 168:108593, 2022.

[68] Xiang Chen, Shaohua Li, Liang Li, Wanzhong Zhao, and Shuo Cheng. Longitudinal-lateral-cooperative estimation algorithm for vehicle dynamics states based on adaptive-square-root-cubature-kalman-filter and similarity-principle. *Mechanical Systems and Signal Processing*, 176:109162, 2022.

[69] Liang Li, Gang Jia, Xu Ran, Jian Song, and Kaihui Wu. A variable structure extended kalman filter for vehicle sideslip angle estimation on a low friction road. *Vehicle System Dynamics*, 52(2):280–308, 2014.

[70] Yan Wang, Liwei Xu, Fengjiao Zhang, Haoxuan Dong, Ying Liu, and Guodong Yin. An adaptive fault-tolerant ekf for vehicle state estimation with partial missing measurements. *IEEE/ASME Transactions on Mechatronics*, 26(3):1318–1327, 2021.

[71] Aymen Alshawi, Stefano De Pinto, Pietro Stano, Sebastiaan van Aalst, Kylian Praet, Emilie Boulay, Davide Ivone, Patrick Gruber, and Aldo Sorniotti. An adaptive unscented kalman filter for the estimation of the vehicle velocity components, slip angles, and slip ratios in extreme driving manoeuvres. *Sensors*, 24(2):436, 2024.

[72] Dongchan Kim, Kyushik Min, Hayoung Kim, and Kunsoo Huh. Vehicle sideslip angle estimation using deep ensemble-based adaptive kalman filter. *Mechanical Systems and Signal Processing*, 144:106862, 2020.

[73] Alberto Bertipaglia, Mohsen Alirezaei, Riender Happee, and Barys Shyrokau. An unscented kalman filter-informed neural network for vehicle sideslip angle estimation. *IEEE Transactions on Vehicular Technology*, 2024.

[74] Beatriz L Boada, María Jesús López Boada, and V Diaz. Vehicle sideslip angle measurement based on sensor data fusion using an integrated anfis and an unscented kalman filter algorithm. *Mechanical Systems and Signal Processing*, 72:832–845, 2016.

[75] Cuong M Nguyen, Anh-Tu Nguyen, and Sébastien Delprat. Neural-network-based fuzzy observer with data-driven uncertainty identification for vehicle dynamics estimation under extreme driving conditions: Theory and experimental results. *IEEE Transactions on Vehicular Technology*, 72(7):8686–8696, 2023.

[76] Tommaso Novi, Renzo Capitani, and Claudio Annicchiarico. An integrated artificial neural network–unscented kalman filter vehicle sideslip angle estimation based on inertial measurement unit measurements. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 233(7):1864–1878, 2019.

[77] Francesco Tufano, Dario Giuseppe Lui, Simone Battistini, Renato Brancati, Basilio Lenzo, and Stefania Santini. Vehicle sideslip angle estimation under critical road conditions via nonlinear kalman filter-based state-dependent interacting multiple model approach. *Control Engineering Practice*, 146:105901, 2024.

[78] Wan Wenkang, Feng Jingan, Song Bao, and Li Xinxin. Vehicle state estimation using interacting multiple model based on square root cubature kalman filter. *Applied Sciences*, 11(22):10772, 2021.

[79] Dae Jung Kim, Seung-Hi Lee, and Chung Choo Chung. Interacting multiple model kalman filter based vehicle lateral motion estimation under various road surface conditions. In *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1234–1239. IEEE, 2018.

[80] Hitoshi Tsunashima, M Murakami, and J Miyataa. Vehicle and road state estimation using interacting multiple model approach. *Vehicle System Dynamics*, 44(sup1):750–758, 2006.

[81] Giseo Park. Vehicle sideslip angle estimation based on interacting multiple model kalman filter using low-cost sensor fusion. *IEEE Transactions on Vehicular Technology*, 71(6):6088–6099, 2022.

[82] Lipeng Cao, Yugong Luo, Yongsheng Wang, Jian Chen, and Yansong He. Vehicle sideslip trajectory prediction based on time-series analysis and multi-physical model fusion. *Journal of Intelligent and Connected Vehicles*, 6(3):161–172, 2023.

[83] Ehsan Hashemi, Mohammad Pirani, Amir Khajepour, Baris Fidan, Shih-ken Chen, and Bakhtiar Litkouhi. Fault tolerant consensus for vehicle state estimation: A cyber-physical approach. *IEEE Transactions on Industrial Informatics*, 15(9):5129–5138, 2019.

[84] Mehdi Zabihi, Reza Valiollahi Mehrizi, Alireza Kasaiezadeh, Mohammad Pirani, and Amir Khajepour. A hybrid model-data vehicle sensor and actuator fault detection and diagnosis system. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[85] Alberto Bertipaglia, D de Mol, M Alirezaei, Riender Happee, and B Shyrokau. Model-based vs data-driven estimation of vehicle sideslip angle and benefits of tyre force measurements. *arXiv preprint arXiv:2206.15119*, 2022.

[86] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[87] Xue-Bo Jin, Ruben Jonhson Robert Jeremiah, Ting-Li Su, Yu-Ting Bai, and Jian-Lei Kong. The new trend of state estimation: From model-driven to hybrid-driven methods. *Sensors*, 21(6):2085, 2021.

[88] Tao Zhang, Yuan Zou, Xudong Zhang, Ningyuan Guo, and Wenwei Wang. Data-driven based cruise control of connected and automated vehicles under cyber-physical system framework. *IEEE Transactions on Intelligent Transportation Systems*, 22(10):6307–6319, 2020.

[89] Attila Widner, Viktor Tihanyi, and Tamás Tettamanti. Framework for vehicle dynamics model validation. *IEEE Access*, 10:35422–35436, 2022.

[90] Zi Cong Guo, Vassili Korotkine, James R Forbes, and Timothy D Barfoot. Koopman linearization for data-driven batch state estimation of control-affine systems. *IEEE Robotics and Automation Letters*, 7(2):866–873, 2021.