

Fault Diagnosis and Reliability-Based Topology Selection of Vehicle State Estimation

by

Mohammadreza Ghorbani

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

© Mohammadreza Ghorbani 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: Chi-Guhn Lee
Professor, Dept. of Mechanical and Industrial Engineering,
University of Toronto

Supervisor(s): Amir Khajepour
Professor, Dept. of Mechanical and Mechatronics Engineering,
University of Waterloo

Internal Member: Baris Fidan
Professor, Dept. of Mechanical and Mechatronics Engineering,
University of Waterloo

Internal-External Member: Dan Davison
Associate Professor, Dept. of Electrical and Computer
Engineering, University of Waterloo

Internal-External Member: Sirisha Rambhatla
Assistant Professor, Dept. of Management Science and
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

Modern vehicle systems are increasingly reliant on accurate and robust state estimation to ensure safe and reliable operation of advanced control functionalities, such as driver-assistance and autonomous driving systems. However, the inherent complexity, along with various sensor faults, environmental disturbances, and model uncertainties, poses significant challenges to the resilience and reliability of vehicle state estimators—especially in safety-critical applications. This thesis addresses these challenges by developing a unified framework that enables real-time fault diagnosis and reliability-based reconfiguration of estimation architectures.

The core idea is to first model the interconnected architecture of vehicle state estimations as a directed graph—termed the estimation graph—where each node represents a local estimator and edges capture structural dependencies. Within this graph, multiple redundant estimation paths may exist for a given state, enabled by sensory and model-based redundancies. However, faults introduce varying levels of estimation uncertainty across these paths. This research contributes a significant methodological advancement by introducing computationally efficient techniques for real-time reliability assessment, suitable for embedded implementation. These methods enable online selection of the most reliable estimation path based on a quantified reliability index, which reflects the uncertainty due to fault propagation and supports dynamic reconfiguration to the most reliable estimation topology.

Complementing this, the thesis presents a unified and hybrid fault detection and isolation (FDI) methodology that integrates residual-based analysis with data-driven learning, supported by model-based quantified fault likelihoods to enhance diagnostic performance. Moreover, structural dependencies within the estimation architecture are encoded into graph-based representations—namely, the estimation graph and fault interaction graph—which enable structural analysis and scalable fault localization. Leveraging these structural insights, two distributed fault isolation strategies are proposed: a consensus-based approach that enables partial supervision through neighborhood-informed decision-making across fault sources, and a graph neural network (GCN)-based global classifier that incorporates structural priors to enhance diagnostic accuracy and reduce training cost—making it well-suited for large-scale dynamical systems. These structure-aware and computationally efficient designs improve diagnostic performance, reduce retraining overhead compared to centralized approaches, and ensure scalability in complex systems.

The proposed framework is validated through high-fidelity vehicle simulations and experimental on-road data, demonstrating its effectiveness in isolating faults, quantifying

uncertainty, and improving state estimation accuracy. Beyond vehicles, the methodologies developed here are applicable to a wide range of large-scale networked systems—including industrial automation and smart infrastructure—where fault tolerance, modularity, and real-time operation are paramount. This work lays a principled foundation for scalable and resilient state estimation in the face of uncertainty and faults, marking a significant step toward safer and more reliable autonomous systems.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Amir Khajepour, for his tremendous support, kindness, patience, and guidance throughout the course of my PhD. Beyond technical mentorship, I have learned many valuable lessons from him that have profoundly influenced both my academic and personal life.

I am also very grateful to Prof. Mohammad Pirani, whose constant encouragement, support, and approachability made this journey much smoother. His insightful guidance and positive attitude were always uplifting.

I would also like to express my appreciation to Dr. Ladan Khoshnevisan for her generous support and kindness during this journey. She was always available and provided valuable help whenever needed, for which I am truly thankful.

Special thanks to my dear friend Dr. Farshad Kamran for his insightful advice and generous support throughout my PhD.

I gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), General Motors, and the Ontario Research Fund. Without their support, this research would not have been possible.

Dedication

To my family—especially my beloved parents—for their unconditional love, endless sacrifices, and unwavering belief in me.

To my mother, whose love has been a constant source of strength. I will never forget those early days of elementary school when she stayed by my side to ease my anxiety, the countless mornings she rose with me to prepare breakfast, urging me to eat so I would feel energetic and confident at school. Her kindness, care, and daily calls from across the world are a reflection of a love that is truly immeasurable.

To my father, who always encouraged me to learn and grow. He taught me the value of time and knowledge, often reminding me, “Water will not always fill the cruse,” a gentle yet powerful lesson in making the most of every moment.

To my brothers and my lovely sister, whose support and encouragement have meant more than words can express. In moments when I felt tired or discouraged, their presence, belief, and uplifting words always gave me renewed strength.

Everything I have achieved, including this milestone, stands on the foundation they have built with their love, wisdom, and devotion.

Table of Contents

Examining Committee	ii
Author's Declaration	iii
Abstract	iv
Acknowledgements	vi
Dedication	vii
List of Figures	xiv
List of Tables	xxiii
List of Abbreviations	xxv
List of Symbols	xxvii
1 Introduction	1
1.1 Research Context: Structured Approach to Reliable Estimation	5
1.2 Objectives and Contributions	6
1.3 Thesis Organization	10

2	Literature Review	13
2.1	State Estimation Architecture	13
2.2	Uncertainty Quantification in Dynamical Systems	16
2.3	Fault Detection and Diagnosis	20
3	Graph Representation for Vehicle State Estimation	25
3.1	Estimation Graph	25
3.2	Vehicle Estimation Graph	26
3.2.1	Lateral Load Transfer Estimator (E_1)	27
3.2.2	Normal Load Estimator (E_2)	28
3.2.3	Longitudinal Tire Force Estimator by Tire Model (E_3)	29
3.2.4	Longitudinal Tire Force Estimator by Wheel Dynamics (E_4)	30
3.2.5	Lateral Tire Force Estimator (E_5)	30
3.2.6	Kinematics-Based Tire Force Estimator (E_6)	31
3.2.7	Longitudinal Velocity Estimation via Wheel Dynamics (E_7)	32
3.2.8	Lateral Velocity Estimator (E_8)	32
3.2.9	Road Grade Angle Estimator (E_9)	33
3.2.10	Road Bank Angle Estimator (E_{10})	38
3.3	Structural Analysis	40
3.3.1	Structural Fault Isolability Analysis	44
3.3.2	Correlation Analysis for Residual Selection	45
3.3.3	Residual Generation	47
3.4	Summary	48
4	Estimation Uncertainty Quantification	50
4.1	Problem Formulation	50
4.2	Analytical Uncertainty Quantification in Kalman-Based Estimators	52
4.2.1	Computational Complexity	57

4.3	Uncertainty Quantification Using Polynomial Chaos Expansion	58
4.3.1	Mathematical Foundations of Polynomial Chaos Expansion	59
4.3.2	Uncertainty Quantification via Intrusive gPC (Galerkin Method) . .	65
4.3.3	Uncertainty Quantification via Non-intrusive gPC (Stochastic Collocation Method)	69
4.3.4	Computational Complexity	71
4.4	Uncertainty Quantification for Learning-Based Estimators	72
4.4.1	Monte Carlo Dropout	72
4.4.2	Conformal Prediction	72
4.5	Summary	72
5	A Framework for Fault Detection, Isolation, and Identification	74
5.1	Fault Detection	75
5.1.1	Residual-Based Fault Detection	75
5.1.2	Learning-Based Fault Detection	76
5.1.3	Hybrid Voting-Based Fault Detection	77
5.2	Fault Isolation	78
5.2.1	Residual-Based Fault Isolation	78
5.2.2	Model-Based Likelihood Test for Fault Isolation	80
5.2.3	Learning-Based Fault Isolation	82
5.2.4	Hybrid Fault Isolation via Feature Integration	83
5.2.5	Hybrid Consensus-Based Fault Isolation	85
5.2.6	Hybrid Fault Isolation via Graph Convolutional Network	94
5.3	Fault Identification	96
5.3.1	Fault Identifiability	97
5.3.2	Derivation of Influence Matrices	98
5.4	Summary	99

6	Numerical and Experimental Validation	101
6.1	Validation of Vehicle State Estimation Graph	101
6.1.1	Experimental Test Setup	101
6.1.2	Velocity and Longitudinal Tire Force Estimation (Test 1)	103
6.1.3	Longitudinal Velocity Estimation (Test 2)	105
6.1.4	Velocity and Tire Force Estimation (Test 3)	105
6.1.5	Roll Angle Estimation Using Suspension Height Sensors	108
6.1.6	Road Slope Angle Estimation	109
6.2	Validation of Uncertainty Quantification	111
6.2.1	Analytical Uncertainty Quantification	111
6.2.2	Surrogate-Based Uncertainty Quantification (Non-intrusive)	115
6.2.3	Surrogate-Based Uncertainty Quantification (Intrusive)	119
6.2.4	Computational Performance Analysis	123
7	Fault Detection and Diagnosis Results	125
7.1	Fault Onset Detection Stage	126
7.1.1	Residual-based Fault Detection	126
7.1.2	Data-driven Fault Detection	127
7.1.3	Hybrid Fault Detection	128
7.2	Fault Isolation Stage	129
7.2.1	Model-based Likelihood Test for Fault Isolation	129
7.2.2	Data-driven Fault Isolation Results	131
7.2.3	Hybrid Fault Isolation Results	132
7.3	Fault Identification Stage	134
7.4	Distributed Fault Isolation in Large-Scale Systems	136
7.4.1	Vehicle Fault Interaction Graph	137
7.4.2	Numerical Evaluation of Local Fault Detectors	140
7.4.3	Consensus-Based Fault Isolation	143
7.4.4	Graph Convolutional Network for Fault Isolation	152
7.5	Summary	156

8	Optimal Topology in Estimation Graph	158
8.1	Fault Detection and Isolation (Stage 1)	159
8.2	Estimation Uncertainty Quantification (Stage 2)	162
8.3	Optimal Estimation Path Selection (Stage 3)	163
8.4	Experimental Validation and Results	166
8.5	Summary	168
9	Conclusion and Future Directions	170
	References	175
	APPENDICES	188
A	Preliminaries and System Modeling	189
A.1	Mathematical Preliminaries	189
A.2	Graph-Theoretic Preliminaries	190
A.2.1	Graph-Related Terminology	190
A.2.2	Topological Properties for Node Influence	193
A.3	Vehicle Motion Model	195
A.3.1	Longitudinal Dynamics	195
A.3.2	Lateral Dynamics	195
A.3.3	Lateral Dynamics (Linear Bicycle Model)	197
A.3.4	Normal Load Distribution	197
A.3.5	Pitch and Roll Dynamics	198
A.3.6	Suspension Kinematics	199
A.3.7	Tire Model	201
A.3.8	Wheel Dynamics	202
A.4	Fault Modeling	203
A.5	Fault Detectability and Isolability	206

A.5.1	Fault Detectability Condition	206
A.5.2	Fault Isolability Condition	207
A.5.3	Fault Identifiability Condition	208
B	Estimation Uncertainty Quantification in a Luenberger Observer	209
B.1	Uncertainty Propagation in Longitudinal Force Observer via push-forward approach	211
C	Polynomial Chaos Expansion Theory	213
D	Uncertainty Propagation in Stochastic ODEs using Galerkin Projection	215
D.1	Integration using Quadrature Rule	218
D.2	Uncertainty Propagation in Dynamical Systems with Parametric and Process Noise	220
E	One-Mode Projection	227
	Glossary	230

List of Figures

1.1	Software architecture for an automated vehicle, illustrating the interaction among different software components. The estimation and fault diagnosis modules form the core of the architecture. If a failure occurs in the estimation module and is not detected or compensated by the fault diagnosis module, it can propagate to other software components, potentially causing a cascading failure throughout the interconnected system.	2
1.2	A cascaded structure for vehicle state estimation, illustrating the hierarchical interaction among functions responsible for estimating tire-road forces, vehicle velocity, and vehicle and road angles. A malfunction in any function on the upstream can propagate to those on the downstream. It is worth noting that the depicted topology represents just one possible solution among many for the given estimation problem.	3
1.3	The graph of multiple estimation paths. Nodes represent local estimators; dashed lines indicate cascaded estimation paths from the sensor to sink nodes, ε_i denotes sensor layer measurement uncertainty, and δ_i represents node estimation uncertainty.	6
1.4	Real-time quantification of estimation uncertainty in cascaded estimators to obtain the reliable estimation path. E_i denotes local estimator at node i .	7
1.5	Overall proposed framework for reliable state estimation in a complex vehicle system.	8
1.6	Organization and flow of the thesis chapters, illustrating their structure and interdependencies.	12

2.1	Different state estimation architecture for large-scale systems, a) centralized estimation architecture, b) general decentralized or distributed estimation architecture, c) cascaded or hierarchical estimation architecture, and d) federated estimation architecture. In general, state estimation architectures are classified into two major categories: centralized and decentralized. For example, hierarchical and federated architectures are special cases of decentralized estimation architectures.	15
3.1	Illustration of a general estimation graph. The graph is a directed acyclic graph (DAG), representing the cascaded structure of local estimators. For clarity, self-loops are omitted. s_i represents sensor nodes and E_i denotes local estimators. and The red dashed arrows depict an example estimation path originating from a sensor set in layer 0 and terminating at a sink node in layer 2. Here, s_i represents sensor nodes and E_i denotes local estimators. For the formal definition of a path, see Section A.2.1.	26
3.2	Vehicle state estimation graph with local estimator nodes E_1 to E_{10} .	27
3.3	Unknown Input Observer (UIO) structure.	34
3.4	Candidate state estimation paths extracted from the vehicle estimation graph shown in Figure 3.2. The node labeled S represents the sensor set available for each estimation path. A directed walk from S to a node E_i implies that the required sensor information for that estimator is accessible. Each path corresponds to a feasible estimation sequence composed of local estimators. Depending on the target states to be estimated, multiple sub-paths may be derived from each full path.	41
4.1	An estimation graph. The in-neighbors of arbitrary nodes i and j are represented by $\mathcal{I} = \mathcal{N}^{in}(i)$ and $\mathcal{J} = \mathcal{N}^{in}(j)$, respectively. Self-loops are omitted for clarity.	51
4.2	Orthonormal Hermite, Legendre, and Laguerre polynomials of degrees 0 to 3.	63
5.1	Overview of proposed 3-stage fault detection and diagnosis (FDD) framework.	74

5.2	Network architecture for fault detection. The model combines a 1D convolutional layer to capture local spatial features, and an RNN layer to capture temporal dependencies. A dense layer with a sigmoid output performs binary classification to detect faults.	77
5.3	The Architecture of simple recurrent neural network (RNN) layer.	78
5.4	Preprocessing of raw time-series data for neural network training and testing.	79
5.5	Network architecture for fault isolation. The model combines a 1D convolutional layer to capture local spatial features, and an RNN layer to capture temporal dependencies. A dense layer with a softmax output performs multi-class classification to isolate the fault source.	83
5.6	Hybrid structure for fault isolation. Likelihoods for each fault are quantified mathematically by (5.4). A sliding window is applied to time-series features, with each window labeled by the last time step. The two network branches are then integrated.	84
5.7	Framework for distributed fault isolation in networked dynamical systems.	86
5.8	Illustration of (a) Bipartite Fault Signature Graph (BFGS), showing the relationships between faults and their associated residual signatures; and (b) Fault Interaction Graph (FIG), depicting direct relationships between faults based on shared residuals via one-mode projection of the BFGS.	87
5.9	Fault Interaction Graph (FIG) with labeled and unlabeled nodes. Labeled nodes have labeled faulty data, while unlabeled nodes contain only healthy data.	88
5.10	Architecture of the supervised binary classifier at node i .	90
5.11	Architecture of the unsupervised autoencoder for binary fault classification at node i .	91
5.12	Consensus-based node score aggregation for global fault isolation.	94

5.13	Architecture of the GCN-based model for global fault localization. Each graph node corresponds to a local detector, as illustrated in Figure 5.10.	95
6.1	Hardware layout of the experimental setup.	103
6.2	Estimated longitudinal and lateral velocities during Test 1.	104
6.3	Longitudinal tire force estimation by E_3 using the tire model in Test 1.	104
6.4	Longitudinal velocity estimation using the model-based estimator E_7 . The two front wheels are experiencing slip.	105
6.5	Velocity estimation results from the model-based observer on Test 3 with combined wheel slip.	106
6.6	Lateral tire force estimation by E_4 (wheel rotational dynamics) and E_6 (kinematics-based) in Test 3.	107
6.7	Lateral tire force estimation by E_5 (tire model) and E_6 (kinematics) in Test 3.	108
6.8	Estimated suspension heights.	109
6.9	Estimated roll angle using different combinations of suspension height sensors.	110
6.10	Road slope angle estimation using unknown input observer.	110
6.11	A front-axle-drive E-Class SUV model in CarSim, used as the test platform.	111
6.12	An example of a cascaded estimation path for vehicle lateral velocity and lateral tire forces. The path includes three estimator nodes: E_1 estimates roll angle and load transfer, E_2 estimates vertical loads, and E_5 estimates lateral velocity and tire forces. It is assumed that the longitudinal velocity remains constant and the longitudinal tire forces are negligible.	112
6.13	Cornering maneuver test at a constant speed of 70 km/h: Steering angle, roll angle, yaw rate, and lateral acceleration.	112

6.14	Uncertainty quantification framework for a cascaded estimation path, implemented in Simulink with CarSim integration.	114
6.15	Comparison of analytical and Monte Carlo-based estimation uncertainty in lateral tire forces under (a) bias fault and (b) excessive noise fault in a_y . The variance of both faults is set to $0.5 \text{ m}^2/\text{s}^4$.	116
6.16	Analytical vs. Monte Carlo-simulated quantification of estimation uncertainty of lateral velocity V_y under bias and excessive noise faults in a_y measurement. The variance of both faults is set to $0.5 \text{ m}^2/\text{s}^4$.	117
6.17	Time evolution of the generalized polynomial chaos (gPC) coefficients $c_k(t)$ associated with each orthogonal basis function.	119
6.18	Surrogate-based vs. Monte Carlo-simulated quantification of estimation uncertainty of lateral tire forces $F_{y,ij}$ under bias fault in a_y measurement. The variance of bias fault is set to $0.5 \text{ m}^2/\text{s}^4$.	120
6.19	Surrogate-based vs. Monte Carlo-simulated quantification of estimation uncertainty of lateral velocity V_y under a bias fault in a_y measurement. The variance of bias fault is set to $0.5 \text{ m}^2/\text{s}^4$.	120
6.20	Intrusive surrogate-based (Galerkin method) vs. Monte Carlo-simulated quantification of estimation uncertainty of lateral velocity V_y under a bias fault in a_y measurement. The variance of the bias fault is set to $0.5 \text{ m}^2/\text{s}^4$. The longitudinal velocity is constant at 70 km/h.	122
6.21	Time evolution of the generalized polynomial chaos (gPC) coefficients $c_k(t)$ associated with each orthogonal basis function.	123
7.1	Multi-stage hybrid FDD framework. $\hat{t}_f$ is the detected fault onset time, and the hypothesis $\hat{H}_i$ denotes the isolated fault class.	126
7.2	The behavior of residuals in non-faulty conditions under both smooth and harsh cornering maneuvers. For harsh maneuvers, residuals do not maintain zero mean.	128
7.3	Comparing fault detection performance using residual patterns.	130

7.4	Estimation residuals under a bias fault in $a_{y,m}$ are shown versus the $1\text{-}\sigma$ uncertainty band for different fault cases. The fault occurs at $t = 5$. As expected, the residuals closely match the uncertainty bound in column 1, corresponding to bias in $a_{y,m}$	131
7.5	Log-likelihood values of residuals for different fault types, with a bias of 0.7 m/s^2 in $a_{y,m}$ occurred at $t = 5$ s. As expected, the likelihood for the $a_{y,m}$ bias is highest.	132
7.6	Log-likelihood values of residuals for different fault types, with a bias of 0.01 rad/sec in $\dot{\psi}_m$ occurred at $t = 0$ s.	132
7.7	Comparison of confusion matrices for the model-based, pure data-driven, and hybrid fault isolation approaches.	133
7.8	Isolation accuracy of different methods vs. data size.	134
7.9	Model-based identification of a bias fault of 0.7 m/s^2 in $a_{y,m}$ occurred at $t = 1$ s.	135
7.10	A model-based identification of a 1 m/s^2 bias in $a_{z,m}$ occurring at $t = 1$ s. As expected, with the fault undetectable at $\delta = 0$, the value converges incorrectly after $t = 7$ s.	135
7.11	Sensor layout for fault diagnosis framework.	137
7.12	Correlation matrices between residuals for two sensor groups: (a) suspension height sensors, and (b) yaw rate sensors.	139
7.13	Sensitivity scores of the trained neural network model for detecting a fault in the a_x measurement with respect to different residual features.	140
7.14	Fault interaction graph (FIG) for the vehicle system, obtained via one-mode projection of the bipartite fault-signature graph (BFSG). Each node corresponds to one of the 18 fault sources, and edges indicate fault interactions arising from shared influences on residuals.	142
7.15	Reconstruction error histogram for fault detection of $a_{y,m}$ using an autoencoder trained on healthy data. The threshold is set to balance detection accuracy, as shown in the confusion matrix.	144
7.16	Confusion matrix for fault isolation with all nodes labeled.	145
7.17	Confusion matrix for fault isolation with all nodes unlabeled.	146

7.18	Illustration of two cases of consensus for global fault localization.	147
7.19	Different topologies of the Fault Interaction Graph (FIG), each exhibiting distinct fault isolation performance based on connectivity patterns.	150
7.20	Average number of labeled and unlabeled neighbors per unlabeled node under different labeling configurations. As k increases, unlabeled nodes are more exposed to labeled ones ($L \rightarrow U$) and less connected to each other ($U \rightarrow U$). .	151
7.21	Classification accuracy, precision, and coverage ratio of unlabeled nodes under different FIG topologies. Bar plots show the performance metrics (left y-axis), while the orange line shows the coverage ratio (right y-axis). Increased coverage corresponds to improved performance across configurations.	153
7.22	Comparison of the proposed distributed GCN-based architecture and the centralized architecture in terms of fault isolation accuracy and training efficiency (number of epochs to convergence) versus the number of model parameters.	155
7.23	Comparison of the proposed distributed GCN-based architecture and the centralized architecture in terms of fault isolation accuracy and training efficiency (number of epochs to convergence) versus the training data size.	155
8.1	Structured framework for fault isolation and reliability assessment of multiple estimation paths under diverse fault scenarios to enable optimal path selection.	159
8.2	Illustration of two distinct estimation paths for the lateral velocity V_y within the vehicle estimation graph.	160
8.3	Residual patterns under different fault sources and lateral disturbance. . .	161
8.4	Comparing lateral velocity estimation and its quantified uncertainty for paths 1 and 2 following a bias fault onset in a_y at 2 seconds (bias = 0.5 m/s^2).	164
8.5	Comparing lateral velocity estimation and its quantified uncertainty for paths 1 and 2 following a bias fault onset in $\dot{\psi}$ at 2 seconds (bias = 0.02 rad/s).	165
8.6	Comparison of uncertainty measure ($\bar{\sigma}[\hat{V}_y]$) in paths 1 and 2 for different fault sources and fault types. The threshold is set at $0.1 \max(V_y)$	166

8.7	Residuals of lateral acceleration a_y in estimation paths 1 and 2. A bias fault of 1 m/s^2 is injected into a_y at $t = 6 \text{ s}$	167
8.8	Lateral velocity estimation (top) and reliability index (bottom) for estimation paths 1 and 2. A bias fault of 1 m/s^2 is injected into a_y at $t = 6 \text{ s}$. Fault detection occurs at $t = 6.1 \text{ s}$, after which the reliability indices are evaluated, triggering a reconfiguration from path 1 to path 2 at $t = 6.2 \text{ s}$	169
A.1	An example digraph, its strongly connected components, and the corresponding condensation digraph. The condensation digraph forms an acyclic digraph.	191
A.2	(a) A potential path (stream) in an acyclic digraph. (b) Acyclic digraph illustrating a cascaded estimation scheme. The in-neighbors of node i are represented by $\mathcal{N}^{in}(i)$, and its upstream nodes are denoted by $\mathcal{N}^{up}(i)$. For clarity, self-loops are not shown.	192
A.3	A weighted digraph and its adjacency matrix.	193
A.4	A weighted bipartite graph, where w_{ij} denotes the weight of the edge between node $u_i \in \mathcal{U}$ and node $v_j \in \mathcal{V}$	193
A.5	Illustration of node configurations: (a) a minimum-size 1-tuple dominating set and (b) a maximal independent set within a graph.	194
A.6	Vehicle dynamics illustration and geometric definitions: (a) side view showing longitudinal acceleration and pitch dynamics, (b) rear view illustrating lateral acceleration and roll behavior, and (c) top view depicting velocity components, yaw rate, tire forces, and geometric parameters. Forces and accelerations are shown in the vehicle coordinate system, and the center of gravity (CG) is used as the reference point for motion states.	196
A.7	Illustration of vehicle body motion due to road geometry: (a) pitch dynamics on a road with grade angle, and (b) roll dynamics on a road with bank angle.	199

A.8	Suspension kinematics and sensor positions in the vehicle frame.	200
A.9	Interpretation of the coefficients of Magic Formula tire model [1].	202
D.1	Comparison between Monte Carlo (MC) simulation and Galerkin-based generalized polynomial chaos (gPC) expansion results. Second-order polynomials are used for both input variables X_1 and X_2 , resulting in $p = 6$ expansion terms.	219
D.2	Time evolution of the (gPC) expansion coefficients.	220
D.3	Comparison between Monte Carlo (MC) simulation and Galerkin-based generalized polynomial chaos (gPC) expansion results for the stochastic ODE with both parametric uncertainty and additive process noise. Second-order polynomials ($\rho = 2$) are used for the two independent input variables X_1 and X_2 . The random variables are distributed as follows: $X_1 \sim \mathcal{N}(0.05, 0.01^2)$, $X_2 \sim \mathcal{N}(8, 0.01^2)$, and the process noise $\omega(t)$ is zero-mean white noise with variance 0.01^2	226
E.1	One-mode projection of the bipartite graph in Figure A.4, illustrating direct relationships between nodes $u \in \mathcal{U}$ based on their shared neighbors $v \in \mathcal{V}$	228

List of Tables

3.1	Binary structural matrix $\mathbf{S}$: Rows represent estimation paths $\mathcal{P}_i$; columns represent faults f_i . Entry of 'x' indicates structural dependency.	42
3.2	Incidence matrix $\mathbf{\Gamma}$: Rows represent residuals r_i ; columns represent the estimation paths $\mathcal{P}_i$. Entry of 'x' indicates structural dependency.	43
3.3	Fault Signature Matrix: Rows represent residuals r_i ; columns represent faults f_j . Entry of 'x' indicates structural dependency.	46
3.4	Summary of residual definitions. Superscript (E_i) indicates that the variable is estimated by local estimator E_i	49
4.1	Recursive quantification of state estimation uncertainty	56
4.2	Hermite, Legendre, and Laguerre Orthonormal Polynomials	63
4.3	Hermite polynomials for two random variables X_1 and X_2	65
5.1	Example fault signature matrix indicating the sensitivity of each residual r_i to the fault scenarios f_j	80
6.1	Specifications of Chevrolet Equinox test vehicle.	102
6.2	Maneuvers and road conditions in the experimental test dataset.	102
6.3	Parameters of the Test Vehicle from CarSim	113
6.4	IMU Sensor Specifications	114
6.5	Comparing time-averaged RMSE from Monte Carlo and analytical SD ($\bar{\sigma}_a$) for estimated states under various faults.	118
6.6	Comparing time-averaged RMSE from Monte Carlo and surrogate-based SD ($\bar{\sigma}_{gPC}$) for estimated states under various faults.	121

6.7	Comparison of runtime for computing state covariance matrices across the estimation path in Figure 6.12.	124
7.1	Total dataset time samples and percentage for each fault case in 100 simulations for detection and isolation tasks.	127
7.2	Comparing the contribution of Conv and RNN layers to network performance in Figure 5.2 on the test dataset.	129
7.3	Summary of fault sources, associated variables, and injected fault ranges. Sensor and actuator faults are simulated as constant random biases within the specified ranges. Unknown disturbances and parameter errors are modeled as deviations from near-zero or nominal values.	138
7.4	Fault signature matrix for a vehicle system. Rows represent residuals as fault signatures, and columns represent the fault sources.	141
7.5	Accuracy of local fault detectors at each node. Supervised classifiers are trained on labeled faulty data, while unsupervised autoencoders are trained on healthy data only.	143
7.6	Consensus scores when all nodes are unlabeled.	148
7.7	Consensus scores when node 1 is unlabeled and neighbors are labeled. . . .	148
7.8	Comparison of fault isolation methods in terms of classification accuracy and model complexity.	156
8.1	Fault signature matrix indicating which residuals are impacted by each fault. A residual is marked as impacted if its magnitude exceeds a predefined threshold (ϵ).	162

List of Abbreviations

k -TDS k -Tuple dominating set 149, 194

BFSG bipartite fault signature graph 85

DAG directed acyclic graph 190

FDD fault detection and diagnosis 4

FDI fault detection and isolation 20

FIG fault interaction graph 85, 149

FWD Front-wheel drive 195, 202

GCN graph convolutional network 22, 136

GNN graph neural network 22

gPC generalized polynomial chaos 59, 61

IMU inertial measurement unit 5, 205

MC Monte Carlo 57

MIS maximum independent set 194

NDS networked dynamical system 20

PCE polynomial chaos expansion 8

RMSE root mean squared error 115

RNN recurrent neural network 76, 95

UIO unknown input observer 21, 33, 35

UQ uncertainty quantification 13, 51

List of Symbols

- $F_{x,f}$ Total longitudinal force at the front axle (N) 31
- $F_{x,ij}$ Longitudinal tire force at wheel ij (N), where $ij \in \{\text{fl}, \text{fr}, \text{rl}, \text{rr}\}$ 195, 201
- $F_{x,r}$ Total longitudinal force at the rear axle (N) 31
- $F_{y,f}$ Total lateral force at the front axle (N) 31
- $F_{y,ij}$ Lateral tire force at wheel ij (N), where $ij \in \{\text{fl}, \text{fr}, \text{rl}, \text{rr}\}$ 195, 201
- $F_{y,r}$ Total lateral force at the rear axle (N) 31
- $F_{z,ij}$ Vertical tire force at wheel ij (N), where $ij \in \{\text{fl}, \text{fr}, \text{rl}, \text{rr}\}$ 195
- I_w Moment of inertia of the wheel ($\text{kg}\cdot\text{m}^2$) 203
- I_{xx} Roll moment of inertia of the vehicle ($\text{kg}\cdot\text{m}^2$) 199
- I_{yy} Pitch moment of inertia of the vehicle ($\text{kg}\cdot\text{m}^2$) 199
- I_{zz} Yaw moment of inertia of the vehicle ($\text{kg}\cdot\text{m}^2$) 195
- R_e Effective wheel radius (m) 203
- $T_{b,ij}$ Brake torque applied at wheel ij (Nm), where $ij \in \{\text{fl}, \text{fr}, \text{rl}, \text{rr}\}$ 30, 32, 203
- $T_{d,ij}$ Drive torque applied at wheel ij (Nm), where $ij \in \{\text{fl}, \text{fr}, \text{rl}, \text{rr}\}$ 30, 32, 203
- V_x Longitudinal velocity of the vehicle (m/s) 195
- V_y Lateral velocity of the vehicle (m/s) 195
- $\Delta F_{z,l}$ Lateral load transfer to the left side of the vehicle (N) 27

Δz_{ij} Suspension vertical displacement at wheel ij , where $ij \in \{\text{fl}, \text{fr}, \text{rl}, \text{rr}\}$ (m) 199

α_{ij} Tire slip angle at wheel ij 202

δ_f Front axle steering angle, assuming $\delta_{fl} = \delta_{fr}$ 31, 195

$\dot{\phi}$ Roll rate of the vehicle (rad/s) 199

$\dot{\psi}$ Yaw rate of the vehicle (rad/s) 202

$\dot{\theta}$ Pitch rate of the vehicle (rad/s) 199

λ_{ij} Tire slip ratio at wheel ij 201

$\mathbb{C}[\cdot]$ Covariance operator 190

$\mathbb{E}[\cdot]$ Expectation operator 190

$\mathbf{f}_b$ Observation fault vector: representing any sensor fault and external disturbances acting on the output model of the system 51

$\mathbf{f}_d$ Process fault vector: representing any actuator fault and external disturbances acting on the process model of the system 51

μ Road friction coefficient 201

ω_{ij} Rotational speed of wheel ij (rad/s), where $ij \in \{\text{fl}, \text{fr}, \text{rl}, \text{rr}\}$ 203

ϕ Roll angle of the vehicle body (rad) 197, 199

ϕ_r Road bank angle (rad) 199

σ Cornering relaxation length used in the lateral tire force dynamics model 30

θ Pitch angle of the vehicle body (rad) 199

θ_r Road grade angle (rad) 199

a_x Longitudinal acceleration of the vehicle (m/s²) 198

a_y Lateral acceleration of the vehicle (m/s²) 198

a_z Vertical acceleration of the vehicle (m/s²) 198

b_w Track width of the vehicle (m) 195, 198

- c_ϕ Roll damping coefficient (N·m·s/rad) 199
- c_θ Pitch damping coefficient (N·m·s/rad) 199
- g Gravitational acceleration (9.81 m/s²) 197
- h_P Vertical distance from pitch center (PC) to center of gravity (CG) (m) 199
- h_R Vertical distance from roll center (PC) to center of gravity (CG) (m) 199
- h_{CG} Height of the center of gravity (m) 198
- h_{RC} Height of the roll center (m) 28
- k_ϕ Roll stiffness coefficient (N·m/rad) 199
- k_θ Pitch stiffness coefficient (N·m/rad) 199
- k_f, k_r Suspension stiffness coefficients of the front (k_f) and rear (k_r) axles 28
- l_f Distance from the center of gravity to the front axle (m) 195
- l_r Distance from the center of gravity to the rear axle (m) 195
- m Total mass of the vehicle (kg) 195
- m_s Sprung mass of the vehicle (kg) 199
- B_x, C_x, D_x, E_x Empirical coefficients in the Magic Formula for longitudinal tire model 201
- B_y, C_y, D_y, E_y Empirical coefficients in the Magic Formula for lateral tire model 202
- $(\cdot)_m$ Measured value of a variable; e.g., $a_{y,m}$ denotes the measured lateral acceleration 27, 197

Chapter 1

Introduction

In recent years, improving the comfort, stability, and handling performance of vehicles has been of great interest in both academia and the automotive industry. Many considerable advancements in chassis control systems, such as active safety control, advanced driver assistance systems (ADAS), and anti-lock braking systems (ABS), have been made. Consequently, the need for reliable vehicle state information to successfully implement those control systems is felt more than ever. However, directly measuring all vehicle states using onboard sensors is often costly or even infeasible. Moreover, sensor data may be imprecise or unreliable under complex operating conditions. Therefore, accurate and robust state estimation has become essential for effective vehicle control, and many studies in the literature—surveyed in [2, 3, 4]—underline its importance.

With the ever-increasing developments in modern vehicles and automated driving, safety is becoming essential to yielding high levels of autonomy. One of the most critical components of safety in road vehicles is functional safety—defined under ISO26262 (Road Vehicles: Functional Safety) as the absence of unreasonable risk due to hazards caused by malfunctioning behavior of electrical/electronic systems. Figure 1.1 illustrates a representative software architecture for an automated vehicle, highlighting the central role of the state estimation and fault diagnosis modules. As shown, the state estimation module lies at the core of the system; any malfunction in this module—if not detected and mitigated by the fault diagnosis and recovery mechanism—can propagate to other software components, posing serious safety risks. Therefore, robust and reliable state estimation, coupled with effective fault diagnosis and recovery, is essential not only for current mass-production vehicles but also for achieving full autonomy goals soon.

The complexity of interconnected vehicle systems necessitates the use of a multi-path

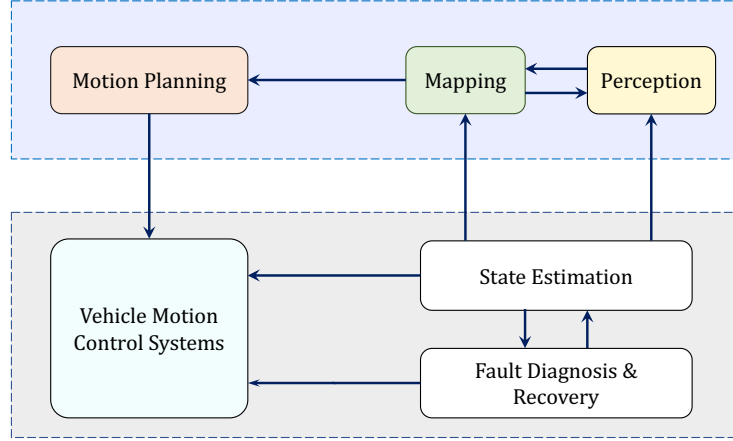


Figure 1.1: Software architecture for an automated vehicle, illustrating the interaction among different software components. The estimation and fault diagnosis modules form the core of the architecture. If a failure occurs in the estimation module and is not detected or compensated by the fault diagnosis module, it can propagate to other software components, potentially causing a cascading failure throughout the interconnected system.

cascaded structure for state estimation, where multiple sub-modules estimate individual vehicle states—such as tire forces, vehicle velocity, and wheel torques—in a hierarchical and interdependent manner [5, 6, 7]. This modular approach offers advantages in terms of flexibility, robustness, and the ability to incorporate diverse models tailored to specific dynamics or operational conditions. However, it also introduces a critical vulnerability: estimation errors or faults in one sub-module can propagate through the cascade, degrading the performance of downstream estimators. Hence, cascading failures may occur not only at the vehicle system level, as discussed earlier, but also within the state estimation module itself, posing a significant risk to the reliability and safety of the overall system. Ensuring the resilience of such cascaded architectures—while leveraging redundancy and modular fault isolation capabilities—is essential for robust vehicle operation.

Figure 1.2 schematically illustrates a representative topology for estimating vehicle states and tire-road forces. In this context, topology refers to a cascaded arrangement of interdependent functions, where each sub-module estimates a specific set of vehicle states or parameters. Such a modular cascaded structure is often necessary due to the large number of states to be estimated and the limitations of centralized approaches in terms of robustness, fault tolerance, and scalability. However, in the cascaded cascaded structure faults or inaccuracies in upstream sub-functions—such as those related to raw measurements—can

propagate to downstream estimators, potentially compromising the accuracy and reliability of the entire state estimation process. This raises an important question: can the estimation topology be reconfigured to improve resilience against cascading failures while maintaining estimation performance? Notably, the depicted topology represents just one of many feasible configurations, and alternative architectures may offer enhanced fault tolerance or robustness.

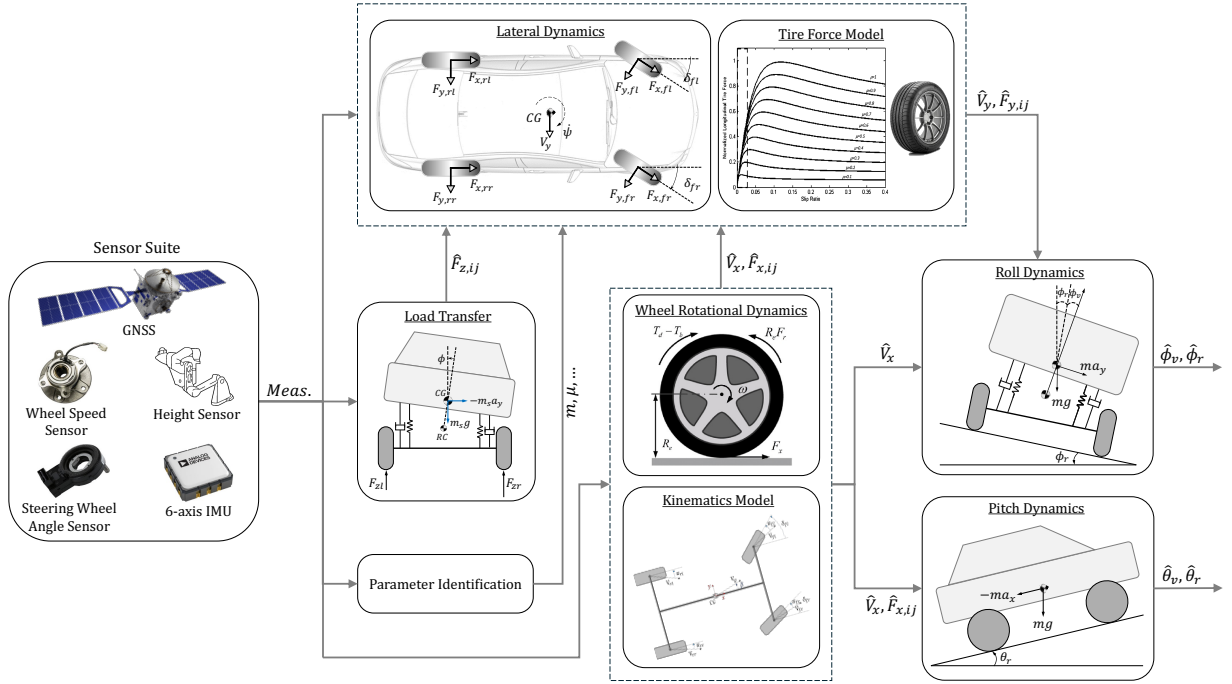


Figure 1.2: A cascaded structure for vehicle state estimation, illustrating the hierarchical interaction among functions responsible for estimating tire-road forces, vehicle velocity, and vehicle and road angles. A malfunction in any function on the upstream can propagate to those on the downstream. It is worth noting that the depicted topology represents just one possible solution among many for the given estimation problem.

The need for reconfigurable estimation structures arises not only in response to faults but also due to varying vehicle operating conditions. For instance, during low-speed maneuvers such as parking, accurate estimation of the yaw angle may be more critical than that of the side-slip angle. In contrast, at high speeds or during aggressive driving, other states may take precedence. This operational variability highlights the importance of multi-path estimation, where different estimation topologies can be selectively activated based on the

current task, operating condition, or detected fault.

Fortunately, such flexibility is enabled by the redundancy inherent in vehicle systems—including redundant sensor measurements (physical redundancy) and a wide range of overlapping physical and data-driven models with different levels of fidelity (functional redundancy), such as kinematic, dynamic, and tire models. These redundancies allow for multiple alternative estimation pathways, enabling reconfiguration of the estimation structure to maintain accuracy, robustness, and computational efficiency despite changing conditions or component failures.

To enable such reconfiguration in a fault-resilient manner, effective fault detection and diagnosis (FDD) is essential. Modern vehicles rely on numerous interacting subsystems—such as sensors, actuators, and estimation modules—that are susceptible to a wide range of faults. As vehicles become more automated and interconnected, maintaining high reliability and meeting strict safety standards increasingly depends on timely and accurate FDD. Accurate fault diagnosis enhances system resilience, mitigates the risk of secondary hazards, and ensures safe and stable operation [8]. An effective FDD framework must be able to detect the occurrence of a fault (*detection*), identify its type and location (*isolation*), and, when necessary, estimate its magnitude (*identification*); fault *diagnosis* includes both isolation and identification.

However, implementing FDD in modern vehicle systems is challenging. Fault propagation through a large network of estimators complicates fault isolation and amplifies error effects across interconnected modules. In addition, many estimators rely on models that are only valid within specific operating ranges; deviations from these conditions introduce modeling errors that can reduce the effectiveness of traditional FDD techniques. The combined effect of structural complexity, model uncertainty, and the diversity of potential faults makes traditional fault diagnosis approaches inadequate for modern vehicle architectures.

Nonetheless, diagnostic performance can be significantly improved by leveraging structural connectivity, system redundancies, and prior model knowledge—such as known fault distributions—within the estimation graph. While centralized models provide a rigorous foundation for diagnosability and system decomposition, they often obscure localized fault behavior and fail to exploit distributed redundancy, which limits their fault isolability. Therefore, there is a pressing need for scalable FDD methods that not only integrate structural information and prior system knowledge but also remain adaptable to modular, large-scale architectures. Such methods are essential for achieving reliable, fault-tolerant estimation in safety-critical vehicle applications.

In summary, the increasing complexity, modularity, and safety demands of modern vehicle systems call for jointly reconfigurable estimation and fault diagnosis frameworks that

can dynamically adapt to faults and changing operational conditions. Although this work is motivated by vehicle systems, the proposed methods and insights are broadly applicable to large-scale networked dynamical systems, including power grids, industrial automation, and other safety-critical cyber-physical systems, where reliability and resilience are of paramount importance.

1.1 Research Context: Structured Approach to Reliable Estimation

The solution to the state estimation problem in complex interconnected systems is not unique, as physical and functional redundancies give rise to multiple estimation paths; see Fig. 1.3. These redundant estimation paths enable the development of a reconfigurable, fault-tolerant estimation scheme that enhances reliability and accuracy across various operating conditions.

Furthermore, complex systems with various sensors and components are subject to diverse fault sources. Each estimation path offers a different level of reliability depending on the fault scenario. Thus, a structured framework is developed to select the most reliable path under different fault scenarios. By localizing the fault source and evaluating the impact of the fault on different estimation paths, the system can reconfigure to the pathway with the lowest associated uncertainty, ensuring safe and reliable operation even in the presence of faults.

In general, the sources of estimation uncertainty can be categorized as: (i) sensor faults (e.g., noise, bias, drift), (ii) actuator faults, (iii) external disturbances (e.g., wind, road grade), (iv) model imperfections (e.g., parameter uncertainty, unmodeled dynamics), and (v) algorithmic errors (e.g., linearization error). For notational simplicity, the term fault is used throughout this thesis to collectively refer to all such sources of deviation from nominal behavior, including sensor and actuator faults, unknown external disturbances, and parametric uncertainties. Investigating the impact of each of these fault sources on the overall estimation performance is essential to achieve reliable estimation. For instance, given a bias in inertial measurement unit (IMU) measurements, the question becomes: how does this bias affect various estimation topologies, and which path yields the lowest uncertainty in the final estimate?

To address this, the goal is to identify the optimal estimation path in terms of reliability under fault conditions. This involves two key steps: first, quantifying the uncertainty introduced by each source along all feasible estimation paths; second, selecting the path

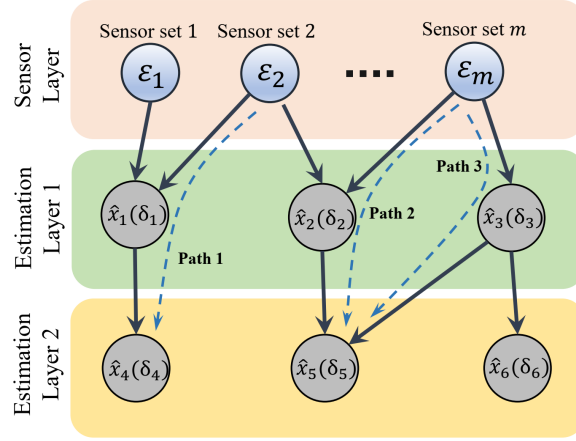


Figure 1.3: The graph of multiple estimation paths. Nodes represent local estimators; dashed lines indicate cascaded estimation paths from the sensor to sink nodes, ε_i denotes sensor layer measurement uncertainty, and δ_i represents node estimation uncertainty.

that minimizes this uncertainty. However, this process depends critically on the ability to detect when a fault has occurred and to isolate its source. From both computational and operational standpoints, it is inefficient and unnecessary to evaluate all paths continuously; instead, uncertainty quantification should be activated in response to a detected fault, guided by the diagnosis of the fault source.

Figure 1.4 illustrates the core idea of this work. On the left, various sensor fault sources—such as steering angle and IMU bias—are shown to influence the estimation accuracy of vehicle states like lateral velocity and tire forces. These fault sources induce uncertainty distributions in downstream estimators. On the right, a directed estimation graph represents a modular cascaded architecture where nodes E_i are local estimators, and edges represent estimation dependencies. The uncertainty associated with each node reflects the compounded impact of sensor faults and structural propagation. By continuously assessing these uncertainties in real time, the framework identifies the most reliable estimation path based on the current system conditions and fault scenario.

1.2 Objectives and Contributions

The scope of this thesis is focused on reliability-based topology selection for vehicle state estimators, particularly in scenarios involving sensor and actuator faults, unknown external disturbances, and parameter uncertainties. The estimation architecture is assumed to be

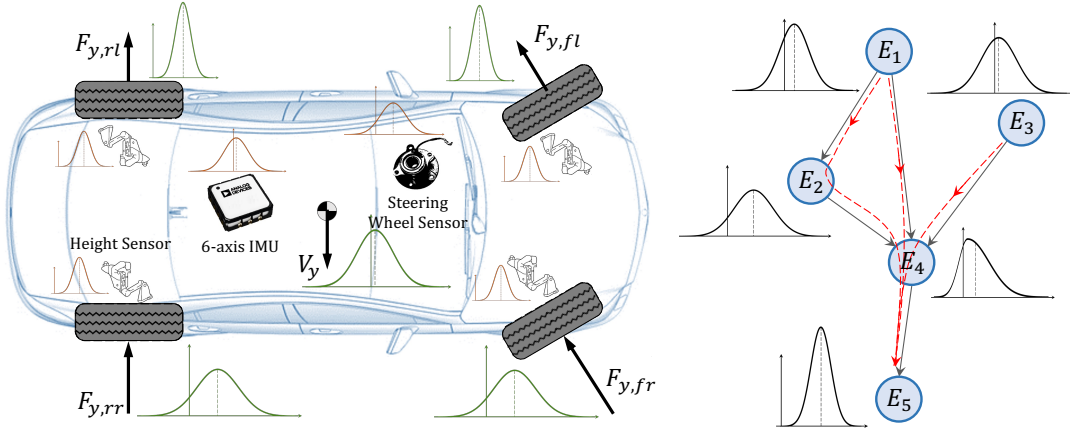


Figure 1.4: Real-time quantification of estimation uncertainty in cascaded estimators to obtain the reliable estimation path. E_i denotes local estimator at node i .

cascaded with predefined interconnections among local estimators. Emphasis is placed on ensuring the computational efficiency of the proposed methods for real-time implementation, especially in the context of uncertainty quantification for nonlinear or data-driven observers, where sampling-based approaches may be impractical. Theoretical developments are validated through both high-fidelity simulations using vehicle models and real-world experimental data collected from on-road vehicle tests.

Figure 1.5 illustrates the overall framework proposed in this thesis for achieving reliable state estimation in complex vehicle systems subject to unintended fault and uncertainty sources. The framework consists of three main stages: fault detection and diagnosis, estimation uncertainty quantification, and reliability-based reconfiguration. In the first stage, the framework performs real-time fault detection followed by fault isolation to localize the source of the fault. In scenarios where the fault is identifiable, the fault magnitude $\hat{f}$ is also estimated. Once the fault has been isolated, the second stage quantifies its impact on each feasible estimation path in the network. Specifically, an uncertainty measure Σ_f^l is computed for each path l with respect to the isolated fault source f , representing the expected estimation uncertainty under that fault scenario. In the final stage, these quantified uncertainties are used to assess the reliability of each estimation path. The system then reconfigures to the path that minimizes uncertainty, thereby selecting the most reliable estimation topology in the presence of the detected fault. In cases where the fault magnitude has been identified, the framework may also apply direct fault compensation to mitigate its effect on estimation performance.

Building on the structure of the proposed framework, this thesis is driven by the following

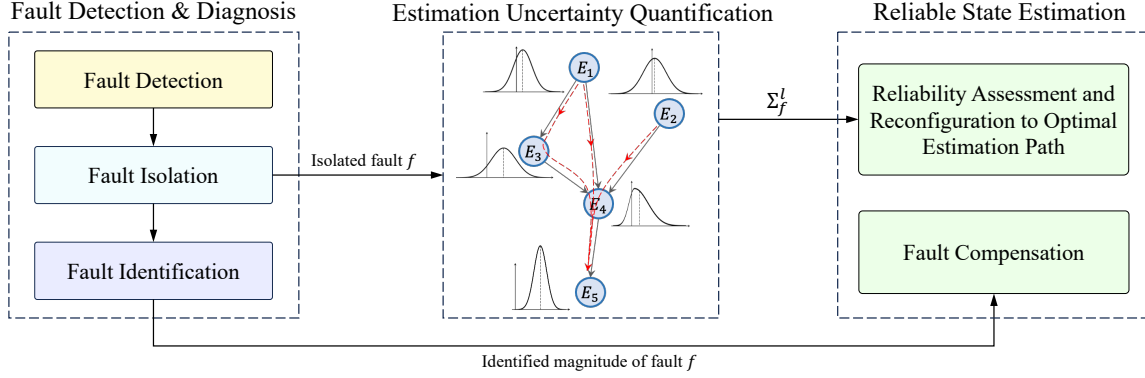


Figure 1.5: Overall proposed framework for reliable state estimation in a complex vehicle system.

main research objectives:

1. **Enable reliability-based reconfiguration in a vehicle estimation graph.** Given an estimation graph with fixed interconnections, the objective is to develop a structured framework that identifies the most reliable estimation path in the presence of faults. This involves real-time fault detection, fault source localization, and the selection of an alternative path based on its sensitivity to the isolated fault. This reconfiguration enables continued reliable estimation even in cases where direct fault compensation is infeasible.
2. **Develop computationally-efficient tools for quantifying estimation uncertainty to support path selection.** The objective is to quantify the impact of isolated faults on each feasible estimation path within the given estimation graph. This includes computing fault-dependent uncertainty measures to evaluate and compare the reliability of alternative paths efficiently at runtime. Additionally, the framework is extended to handle nonlinear or black-box estimators through surrogate modeling approaches such as polynomial chaos expansion (PCE), enabling time-efficient uncertainty quantification in real time without relying on computationally intensive sampling methods like Monte Carlo simulation.
3. **Design a hybrid fault detection and isolation framework integrating model-based and data-driven approaches.** This objective aims to improve fault detection and isolation accuracy by integrating model-based and data-driven approaches. Specifically, the framework leverages residual distributions derived from system models to

compute fault-specific likelihoods, which capture how different fault sources affect residuals across the estimation graph. These quantified likelihoods are then combined with raw residual data in a hybrid neural network architecture to enhance diagnostic performance, especially under uncertainty, unmodeled dynamics, or data imbalance. The approach exploits both temporal and structural patterns in the residuals and further improves fault isolability by incorporating system-level prior knowledge—such as known fault distributions and structural dependencies—into the diagnostic process, without incurring the computational burden of multiple hypothesis testing or maintaining a bank of observers.

4. **Fault magnitude identification and compensation.** This objective aims to estimate the severity of a fault by quantifying its impact on residuals and its propagation across estimation paths. When the fault magnitude is identifiable, the framework enables reliable state estimation through direct fault compensation.
5. **Develop a structure-aware, distributed fault isolation framework suitable for large-scale interconnected systems.** This objective focuses on designing a distributed learning-based framework that exploits the structural connectivity among fault sources, encoded in a fault interaction graph. By assigning lightweight local classifiers to individual fault sources and enabling topology-aware information propagation, the approach ensures computational scalability and robustness. Structural relationships are leveraged to improve fault isolability and enable simultaneous localization of multiple non-entangled faults.
6. **Enable partial supervision and modular scalability in evolving systems.** The proposed framework is designed to operate under partial supervision, using both labeled and unlabeled data to make reliable fault localization decisions. Through a consensus-based mechanism, information is propagated across the estimation graph, improving accuracy even with limited labeled data. Its modular structure supports subsystem heterogeneity and allows incremental scalability, enabling the integration of new fault sources or subsystems without requiring full model retraining. Additionally, the framework enables graph-theoretic analysis of fault isolability using concepts, providing structural insight into the influence of labeled and unlabeled faults.

While this work is focused on vehicle systems, the methods developed here are broadly applicable to large-scale networked dynamical systems—including power grids, industrial automation, and complex cyber-physical infrastructures—where reliability, adaptability, and scalability are critical to safe and efficient operation.

1.3 Thesis Organization

Figure 1.6 illustrates the organization and logical flow of the thesis chapters. The structure is divided into three main parts: Background, Methodology, and Results & Discussion.

Chapter 2 presents a comprehensive review of the literature on reliable state estimation in large-scale interconnected dynamical systems, focusing on three key perspectives: estimation architecture, uncertainty quantification, and fault detection and diagnosis (FDD). The necessary mathematical foundations for the subsequent developments in the thesis is provided in Appendix A. It includes essential graph-theoretic concepts, detailed modeling of vehicle motion dynamics, and formal conditions for fault detectability, isolability, and identifiability. The vehicle motion model serves as the basis for the development of vehicle state estimators in Chapter 3, while the graph-theoretic framework is employed for the analysis of fault isolation performance.

Chapters 3 to 5 comprise the main methodological contributions. Chapter 3 introduces the concept of the estimation graph and presents a graph-based representation of vehicle motion state estimation. Various state estimators are developed and defined as nodes within this graph structure. The chapter also includes structural analyses to evaluate fault isolability and guide the selection of informative residuals for fault diagnosis. The resulting estimation graph serves as the foundation for the uncertainty quantification and fault isolation frameworks developed in subsequent chapters.

Chapter 4 develops two computationally efficient methods for quantifying the impact of soft faults—specifically sensor bias and excessive noise—on the estimation uncertainty in networked dynamical systems. The first approach provides an analytical formulation of uncertainty propagation in Kalman-based estimators. The second approach constructs a surrogate model based on generalized polynomial chaos (gPC) expansion, suitable for nonlinear and black-box estimators. These methods enable real-time reliability assessment of estimation graphs and facilitate the selection of the most reliable estimation path under faulty conditions. Moreover, the uncertainty quantification framework is extended to residual signals, enabling fault isolation through likelihood-based testing and hybrid classification methods. This extension establishes a direct connection from Chapter 4 to the likelihood-based fault isolation technique in Chapter 5.

Chapter 5 presents a comprehensive framework for fault detection, isolation, and identification. Multiple approaches are developed, including residual-based, learning-based, and hybrid methods. Fault isolation is achieved by integrating model-based likelihood testing with data-driven classifiers, leveraging the residual uncertainty quantified in Chapter 4. Several hybrid isolation schemes are proposed, such as feature-level integration, consensus-

based label propagation, and graph convolutional networks. The chapter concludes with the derivation of influence matrices to support fault identification.

Chapters 6 to 8 present the results and discussion. In Chapter 6, the vehicle state estimation graph introduced in Chapter 3 is validated through experimental testing and numerical simulations under both normal and harsh driving conditions. In addition, the uncertainty quantification methods developed in Chapter 4 are evaluated using high-fidelity co-simulations of CarSim and Simulink through Monte Carlo experiments. A computational performance analysis is also conducted to assess the real-time feasibility of the proposed uncertainty quantification framework.

Chapter 7 presents the results of the FDI methods developed in Chapter 5. The performance of residual-based, data-driven, and hybrid detection methods is compared. Fault isolation results are reported using model-based likelihood testing, learning-based classifiers, and their integration. The chapter further includes fault identification results based on the derived influence matrices. In addition, a distributed fault isolation approach is validated for large-scale systems using the proposed vehicle fault interaction graph and consensus-based decision-making across local detectors.

Finally, Chapter 8 integrates the uncertainty quantification methods developed in Chapter 4 with the fault detection and isolation strategies from Chapter 5 to enable real-time reconfiguration of the estimation graph. A reliability index is defined based on the quantified uncertainty of the target states, and by comparing this index across available estimation paths, the most reliable path is selected dynamically in the presence of faults.

Chapter 9 concludes the thesis by summarizing the key contributions and presenting potential directions for future research.

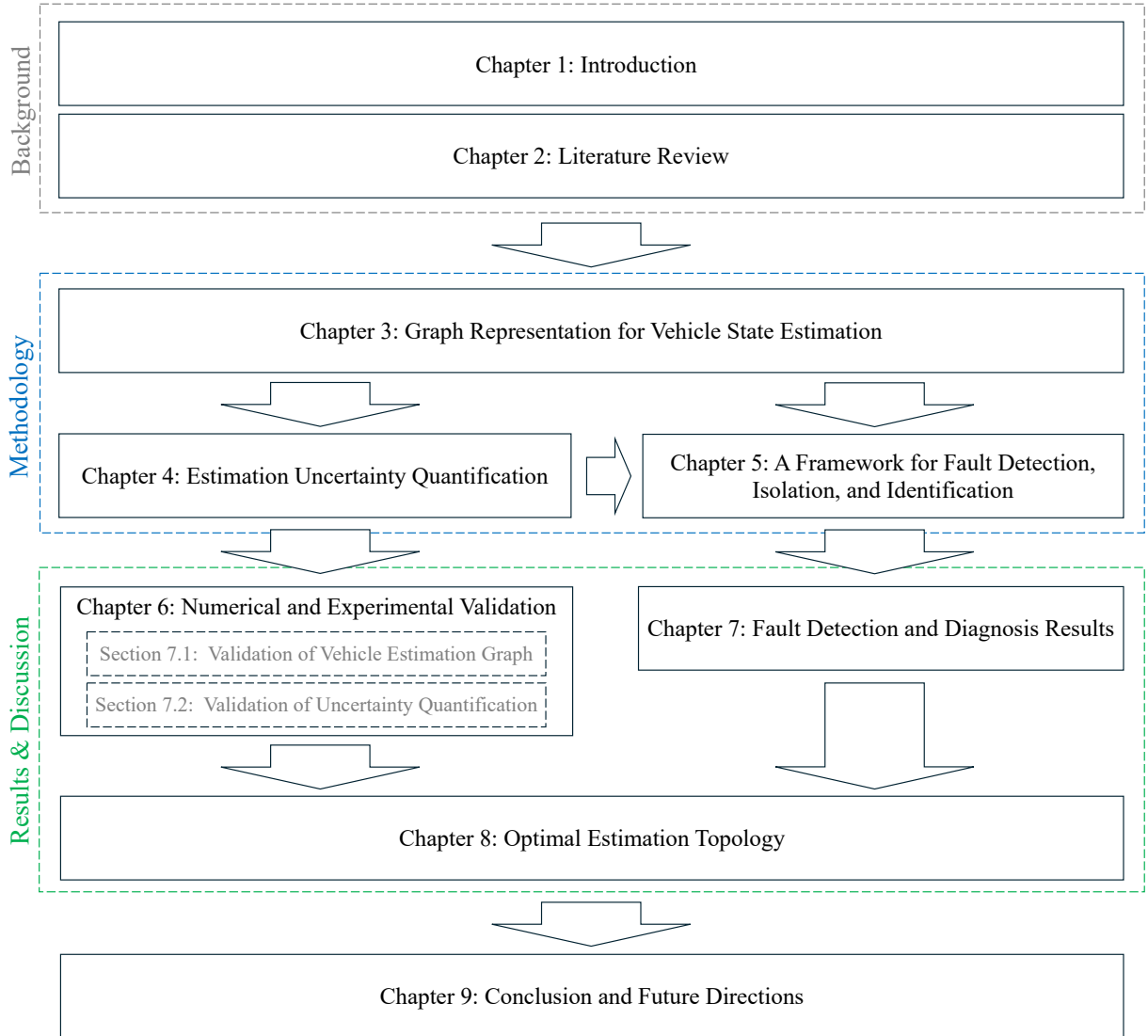


Figure 1.6: Organization and flow of the thesis chapters, illustrating their structure and interdependencies.

Chapter 2

Literature Review

This chapter reviews the state-of-the-art research related to reliable state estimation in large-scale interconnected systems, with a particular focus on vehicle applications. Section 2.1 presents different state estimation architectures for large-scale systems, highlighting the limitations of conventional decentralized schemes when applied to systems with non-linear or hybrid estimation models. Section 2.2 surveys uncertainty quantification (UQ) techniques for dynamical systems and state estimators, including probabilistic and non-probabilistic methods, with particular attention to scalable formulations applicable to structured estimation networks. Finally, Section 2.3 reviews the recent advances in FDI methods, including observer-based, machine learning-based, and hybrid frameworks, with an emphasis on scalability, structural awareness, and real-time applicability in complex systems. These three perspectives—estimation structure, uncertainty quantification, and diagnosis—collectively form the foundation for the proposed reliability-based topology selection framework presented in subsequent chapters.

2.1 State Estimation Architecture

State estimation for large-scale dynamical systems has been a major focus over the past few decades. Due to computational, practical, and reliability constraints, *decentralized* estimation architectures have been widely adopted for multi-sensor systems [9, 10].

Figure 2.1 illustrates different estimation architectures. In the centralized approach (a), all sensor data are processed by a single central estimator (CE). In contrast, architectures (b)–(d) adopt decentralized designs, where the system is decomposed into subsystems and

local estimators (LEs) process local data. Architecture (b) represents a general decentralized framework, where LEs operate on local measurements and may exchange information with neighbors. Hierarchical (c) and federated (d) architectures are structured variants of (b), distinguished by specific communication patterns: (c) organizes estimators in layers with multi-stage fusion, while (d) has independent LEs sending their outputs to a central fusion unit. Both follow the decentralized paradigm but with additional structural constraints. While these architectures enhance scalability and modularity, applying them to complex systems with heterogeneous physics-based and data-driven models remains challenging due to issues of consistency, interconnection, and uncertainty propagation.

To address the computational requirements and practical limitations, large-scale dynamical systems are decomposed into several input/output decentralized interconnected subsystems of a lower order, and the objective is to achieve the centralized estimation performance via decentralized architecture [11]. The problem of multi-sensor state estimation via decentralized architecture encompasses various applications, including target tracking [12], network systems [13], and integrated navigation [14]. The problem in such scenarios is how to incorporate the distributed observations efficiently, referred to as “data fusion” in the literature; see [15]. In multi-sensor data fusion, either no process model is considered [16], or all the sensors share the same full process model through communication with each other.

The decentralization approaches can be utilized to create various estimation pathways and obtain the best estimation structure. However, the need for state-space representation of the system and some particular restrictions to meet stability and observability requirements hinder one from taking full advantage of the proposed methods for nonlinear complicated dynamical systems. For instance, vehicle dynamic systems with non-linear dynamics and potentially data-driven models are difficult to be represented in a decomposed state-space format. The limitations associated with the decentralized state estimation approach in the literature can be summarized as follows:

- They all deal with the state space representation
- In fact, they have mostly decentralized Kalman filtering, while there might exist other model-based or data-driven observers in the estimation process.
- Some special assumptions are made for observability and stability criteria
- There is no consideration of sensor bias error or modeling errors

The cascaded architecture in Figure 2.1c) offers a modular and scalable structure particularly suited to complex systems such as vehicles. By organizing estimators in sequence,

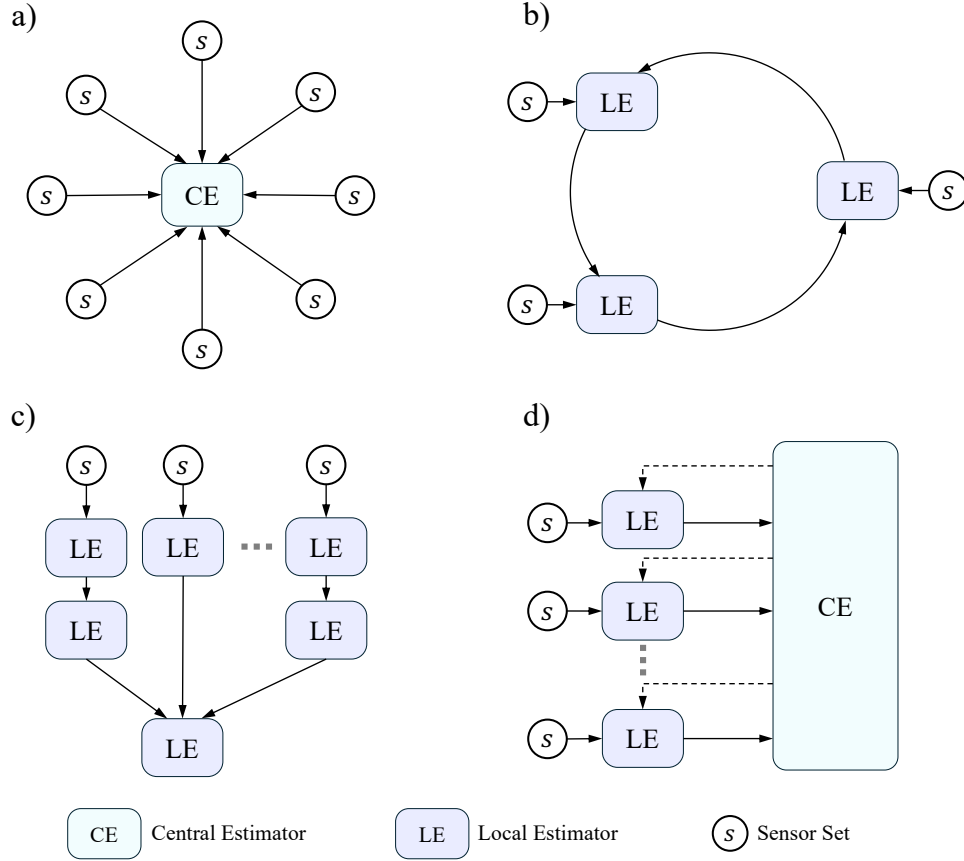


Figure 2.1: Different state estimation architecture for large-scale systems, a) centralized estimation architecture, b) general decentralized or distributed estimation architecture, c) cascaded or hierarchical estimation architecture, and d) federated estimation architecture. In general, state estimation architectures are classified into two major categories: centralized and decentralized. For example, hierarchical and federated architectures are special cases of decentralized estimation architectures.

it enables the use of diverse models—ranging from simple kinematics to complex dynamics—adapted to different operational conditions. This flexibility supports redundancy and multi-path estimation. However, such modularity introduces the risk of fault propagation, where errors in upstream estimators affect downstream accuracy. Unlike centralized schemes, cascaded architectures expose structural redundancies but require careful analysis to ensure fault resilience. This highlights the need for a reliability-aware framework to evaluate and

select the most robust estimation pathways in real time.

The reliable estimation of vehicle states via decentralized architectures has been also investigated in the literature. Li and Zhang [17] provides a reliable integrated navigation approach for automated vehicles based on an adaptive fuzzy federated Kalman filter. Li et al. [18] has also proposed a hybrid decentralized filtering to reliable estimation of both vehicle side-slip and yaw angles. The well-known federated Kalman filtering proposed by Carlson [19] has been adopted to combine the estimates from the Kalman-based vehicle state estimator using two local filters and one master filter. The adaptive mechanism proposed in the global federated Kalman filter ensures the estimation reliability under complex driving situations, such as GPS failure and various driving maneuvers. However, the federated filtering algorithm needs an Upper Bound technique to eliminate the correlation between local estimates and a reset procedure to make the global estimate optimal. To achieve reliable and accurate positioning performance, Li and Xu [20] proposes a fusion method consisting of a sliding mode observer to estimate vehicle state information and a federated Kalman filter to isolate and accommodate GPS failures due to its decentralized filtering architecture.

The problem in this study is totally different from the aforementioned decentralized filtering methods addressing the multi-sensor data fusion problem. The optimality of the cascaded state estimation topologies has not been investigated in the literature. The approach of this study is more general, considering different model-based or data-driven estimation methods, as well as various uncertainty sources. The propagation of uncertainty through any possible estimation pathway is quantified to obtain the most reliable estimation topology in each running condition.

2.2 Uncertainty Quantification in Dynamical Systems

The UQ in computational applications is the quantitative characterization of inadequacy and inaccuracy in the underlying mathematical model. The primary goal of uncertainty quantification in the context of dynamical systems and state estimation is to determine the likelihood of the states in each time step, given the uncertainty in measurements and mathematical models.

There are different methods to propagate uncertainty for dynamical systems [21, 22]. The methods for propagating uncertainty through differential and algebraic equations governing the system can be classified into four categories:

Exact analytical methods: The *forward propagation* or *push-forward* method [23] is simply the evolution of the probability density function (pdf) through a function $\mathcal{F}$. For example, the propagation of estimation error covariance in linear Kalman filtering is one example of the push-forward method. This task can be typically complicated if the inputs have a complicated distribution or process $\mathcal{F}$ is non-linear. The Fokker-Planck equation [24, 25], also known as the forward Kolmogorov equation, allows one to quantify the temporal variations in dynamical systems and in stochastic filtering problems [26]. However, solving this partial differential equation can be extremely difficult, especially for real-time applications. The analytical solution for the Fokker-Planck equation exists only for stationary pdfs and is limited to a small class of dynamic systems. The Bayesian framework can also be used to propagate uncertainty in dynamical systems and even in filtering problems [27]. However, the resulting functional equations cannot be generally solved directly. As a special case, the linear Gaussian models can be solved analytically.

Local expansion-based methods: The problem of uncertainty propagation for non-linear dynamical systems can be solved approximately using the linearization techniques [27]. The Taylor series expansion [28] and perturbation method [29] fit into this category. Non-linear propagation of Gaussian statistics can be approximated using Taylor series expansion of the dynamical equation governing the system [30]. These approximate methods are weak against the large variability of inputs and nonlinearity of process [31].

Sampling-based methods: These methods rely on a repeatedly random sampling of system inputs and statistical inference regarding the output distribution. The Monte Carlo simulation (MCS), due to the ease of implementation, is the most popular sampling-based method. However, by increasing the dimensionality, these methods become computationally expensive for dynamic systems [32]. Moreover, obtaining convergent statistics via Monte Carlo for computationally expensive realistic models is not feasible since one may be able to run only a handful of simulations [33]. In the context of state estimation, the Monte Carlo method is used to approximate the pdf evolution of states in ensemble Kalman filtering [34], adaptive Kalman filtering [35], particle filter [36], and Monte Carlo filtering [37].

Surrogate-based methods: The straightforward approach for the propagation of uncertainty is the Monte Carlo simulation. However, MCS is a computationally expensive approach for a dynamical system. In surrogate modeling approaches [38], instead of running a massive number of simulations, some functions are learned to adequately approximate the evaluation of quantities from simulation with less computational effort. The polynomial

chaos expansion (PCE) method is widely-used surrogate modeling for dynamical systems [39, 40, 41, 42].

There are several studies that have analyzed the effect of uncertain parameters on the accuracy of the vehicle dynamics model; see [43]. Sandu et al. [44] has studied the stochastic response of the nonlinear quarter car model under parametric and external uncertainties using the polynomial chaos method. It is shown that the PCE method is more efficient than MC and more accurate than the statistical linearization method. Kewlani et al. [45] has applied the generalized polynomial chaos (gPC) method in the analysis of vehicle dynamic response under the uncertainties associated with terrain and vehicle parameters. They showed that the PC method outperforms the Monte Carlo technique in terms of computational cost.

The above-mentioned probabilistic methods provide statistics of system responses if the pdf of system inputs is given. However, uncertain-but-bounded inputs may not be described easily using precise probability density functions. In this case, the interval method [46, 47] as a non-probabilistic method can more appropriately quantify the uncertainties of bounded inputs solely by a set of lower and upper bounds without any need to precise pdfs. For example, it will be more suitable to treat parameters like vehicle mass and tire-road friction coefficient as interval variables [43].

Two basic questions in the context of state estimation are: 1) "How does the probability distribution function propagate over time in a dynamical system?" and 2) "How does the measurement update affect the pdf?" The answer to this question is comprehensively reviewed in [48]. For nonlinear and non-Gaussian filtering, the answer to the above questions becomes even more challenging. Generally, for nonlinear filtering, no exact solution can be derived; hence different numerical approximation methods like the sequential Monte Carlo approach [37] have been developed to address the problem. The sequential Monte Carlo approaches enable online estimation by combining the Monte Carlo sampling methods and Bayesian framework with reasonable computational cost. These sampling methods enable one to propagate uncertainty through any estimation path with nonlinear governing equations.

A key insight is that faults can be viewed as uncertainty sources within the estimation process. According to [49, 50], anomalous sensor behavior can be represented by the six main types: 1) spike: abrupt, sharp change in the recorded data; 2) constant: a period of constant measurement indifferent to the actual conditions, 3) drift: slow, continuous change in data over time, 4) bias: constant offset in sensor readings, 5) noise: considerable increase in the variance of data, and 6) miss: a period of unavailable data in the record. Generally, the bandwidth of estimators can be adjusted to mitigate intermittent high-amplitude faults,

like spikes. Data miss or constant value faults are easily detectable and can be mitigated using redundant virtual or physical sensors. However, bias faults and excessive noise are common soft sensor faults that remain usable if identifiable and thus require further investigation. External disturbances and actuator faults can also be considered bias and noise in the process model. Uncertainty in nonlinearly introduced parameters or states can be quantified through linearization via the Taylor series expansion or perturbation method, assuming minor uncertainties compared to the signal value.

Traditionally, UQ in dynamical systems relies on Monte Carlo (MC) sampling—such as in the ensemble Kalman filter [34]—to propagate uncertainty through estimators, but these methods are computationally intensive. Surrogate-based approaches like gPC [45] or stochastic collocation [51, 52] offer computational efficiency but require explicit system formulations and tailored implementations, such as the Kalman-PCE estimator [53]. These methods also suffer from the curse of dimensionality, limiting their scalability in large-scale systems. Covariance-based analytical methods can handle specific uncertainty types. Zanetti and Bishop [54] treats uncompensated bias as an uncertainty factor and quantifies its impact within a single Kalman filter—but extending this to cascaded or modular networks of estimators becomes non-trivial due to interconnections and cross-correlations among modules. Hence, there is a pressing need for scalable, computationally efficient UQ methods capable of capturing fault-driven uncertainties across structured, multi-path estimation setups.

Another approach to handling uncertainties is interval observers, which compute bounding sets for system states under known disturbances or faults [55]. For example, Huang et al. [56] applied interval observers for sensor fault detection in uncertain systems, yielding bounded residuals rather than exact estimates, while Raka and Combastel [57] used them to create adaptive thresholds for robust fault detection. Although effective under bounded disturbances, these methods generally focus on isolated estimators and often lack the precision or scalability required for large-scale systems.

In contrast, this work introduces a scalable analytical framework enabling real-time reliability assessment and improved fault isolation across multiple estimation pathways. By representing the estimation architecture as a graph and incorporating prior knowledge of fault distributions, the proposed approach enables selection of the most reliable path under a fault. This structured reliability analysis addresses a critical gap in the fault-tolerant estimation literature, where FDI in cascaded estimators often overlooks the downstream impact of faults on residuals. In cases where faults are isolated but not directly observable, the framework allows for dynamic reconfiguration toward paths with minimal sensitivity to those faults, thereby ensuring reliable state estimation despite fault presence.

2.3 Fault Detection and Diagnosis

Large-scale networked dynamical systems—such as power grids, industrial processes, and complex cyber-physical systems like modern vehicles—are widely deployed in critical applications where safety and reliability are of paramount importance. These systems are inherently complex, comprising numerous interconnected components that operate under stringent performance and safety requirements. Nevertheless, faults such as sensor failures, actuator malfunctions, and noise-induced disturbances are often unavoidable due to system complexity and external uncertainties. If not promptly detected and addressed, these faults can severely degrade system performance, result in significant economic losses, and in worst cases, escalate into cascading failures across the interconnected subsystems [58]. Fault detection and isolation (FDI) and continuous monitoring play a central role in enhancing the resilience of large-scale networked systems, as they provide the necessary situational awareness and enable timely detection, localization, and mitigation of faults and anomalies, thus ensuring that the system can maintain acceptable performance and recover from unexpected disruptions despite model uncertainties and external attacks [59].

FDI in large-scale networked dynamical systems (NDSs) faces significant challenges arising from their inherent complexity, large scale, system heterogeneity, communication constraints, and the scarcity of historical fault data. As the system scale and interconnection density increase, centralized FDI schemes become computationally intractable and less adaptable to dynamic structural changes, uncertainties, and plug-and-play operations [58]. Moreover, centralized approaches suffer from a lack of robustness, flexibility, and the sparsity introduced by communication protocols. To overcome these limitations, recent research has shifted toward distributed and decentralized FDI frameworks, where fault detection tasks are allocated to local detectors or estimators operating with local measurements. These distributed schemes not only reduce computational burden by decomposing the overall problem into smaller units but also enhance the system’s flexibility and modularity, facilitating local fault detection and isolation without compromising the scalability of the approach [60]. Additionally, it has been recognized that distributed architectures are more effective in handling network-induced phenomena such as time delays, communication constraints, and incomplete measurements, which frequently arise in large-scale cyber-physical systems [61]. Nevertheless, ensuring scalable and real-time fault localization across large-scale systems remains an open research problem that necessitates further investigation.

FDI methods are generally categorized into model-based and data-driven approaches, with recent studies increasingly advocating hybrid strategies that combine the robustness and interpretability of model-based techniques with the flexibility of data-driven methods [62].

Model-based FDI methods rely on analytical redundancy to detect inconsistencies between measured and estimated system behavior. Common strategies in this category include parity space approaches, observer-based methods, Kalman filtering, and parameter estimation techniques [63]. Particularly, observer-based FDI methods have been widely adopted to generate residual signals that reflect discrepancies indicative of faults [64]. Unknown input observers (UIOs) [65] have emerged as a prominent approach for achieving disturbance decoupling, enhancing residual robustness by isolating the effects of faults from unknown inputs. UIO-based schemes have proven especially effective in scenarios where faults are modeled as unknown inputs or uncertainties, enabling reliable detection and isolation even under partial observability and external disturbances [66, 67]. Model-based distributed fault detection and isolation (FDI) methods have been extensively studied as scalable and flexible alternatives to centralized approaches for large-scale networked systems. Shames et al. [68] pioneered a fully distributed observer-based FDI framework using local banks of UIOs, enabling each node to detect and isolate faults through local residual generation and neighbor communication. Khan et al. [60] proposed a computationally efficient FDI scheme using a single directional UIO per node, reducing complexity while maintaining isolation capabilities in second-order networked control systems. In [69, 70], they further advanced distributed FDI by proposing networked estimators with single time-scale protocols and graph-theoretic residual-based detection, capable of handling both full-rank and rank-deficient systems under Gaussian noise without restrictive assumptions on communication or observability. Pirani et al. [71] focused on graph-theoretic approaches to leverage system topology for distributed FDI and resilience analysis.

Data-driven methods are increasingly gaining interest in FDI due to the availability of historical data and their ability to learn complex patterns without relying on accurate system models. These methods are typically classified as supervised or unsupervised. Supervised approaches use labeled data and feature extraction for fault classification, but are limited by the need for comprehensive fault labels. In contrast, unsupervised methods detect unknown or unforeseen faults without requiring labeled data. From a data-driven perspective, FDI task can be formulated as a time-series classification problem. Numerous deep learning methods have been developed for this task. Ismail Fawaz et al. [72] and Mohammadi Foumani et al. [73] provide comprehensive reviews and benchmarks of deep neural network architectures for time-series classification. In the context of fault diagnosis, deep learning techniques such as support vector machine (SVM) [74], convolutional neural network (CNN) [75], recurrent neural network (RNN) [76], and Transformer-based models [77] have been widely used to extract discriminative temporal features from sensory data [78]. Autoencoders have also been applied as unsupervised feature extractors, leveraging reconstruction errors for fault detection without requiring labeled data [79]. Despite their

strong performance, these methods often overlook the spatial and topological relationships inherent to physical systems.

Graph neural networks (GNNs) have emerged as structure-aware learning architectures capable of embedding physical topology directly into the learning process [80]. Chen et al. [81] introduced a GNN model using correlation-based adjacency matrices and residual attention to improve robustness. Spatial-temporal graph convolutional network (GCN) [82], interaction-based edge learning [83], and topology-guided GNN [84] further enhance fault diagnosis accuracy and interpretability. Nguyen et al. [85] combined GCN with RNN in a recurrent-GNN (R-GNN) architecture to jointly model the topological and temporal dynamics of power grids. Darvishi et al. [86] integrated spatiotemporal feature fusion and attribution-based interpretability for cyber-physical fault diagnosis. Benchmark studies [87] emphasize that proper graph construction and pooling significantly affect diagnostic performance. GNNs thus provide a unified mechanism for combining spatial structure with temporal dynamics, improving interpretability, generalization, and scalability in large-scale systems.

Several recent studies extend this line of research by integrating structural models and fault connectivity into GNN-based architectures. Nguyen and Chioua [88] embedded predefined fault propagation paths into a GNN using subgraphs derived from process topology, while Zhang and Hu [89] used hierarchical graphs and link prediction for fault chain modeling in steam turbines. Chen et al. [90] integrated structural analysis (SA) and GCN to create a semi-supervised hybrid model based on residual-informed association graphs. Similarly, Miao et al. [91] constructed spatial-temporal graphs from robot dynamics and used backward difference features to track fault evolution. Guo et al. [92] proposed a multi-view sensor GNN framework with attention over heterogeneous sensor configurations, and Wu et al. [93] incorporated physical sensor topology into a GNN-transformer hybrid for robust fault isolation. These approaches demonstrate the value of combining physics-based modeling, residual structures, and system inter-connectivity with graph learning to achieve scalable, interpretable fault localization in networked systems.

Residual signals in model-based FDI can be systematically arranged into a fault signature matrix, which indicates the sensitivity of each residual to specific faults and serves as the foundation for isolability analysis [94, 95]. The structural relationships between residuals and faults can also be represented as a fault-residual bipartite graph, enabling graphical reasoning about diagnosability and guiding residual selection [96]. In addition, fault-to-fault relationships can be captured using the fault isolability matrix [97], which encodes whether pairs of faults can be distinguished from one another based on their residual signatures—effectively describing potential fault interactions. Leveraging such structural representations, several hybrid approaches have been developed that combine analytical

redundancy relations with machine learning. These include classifier-based models [96], residual evaluation frameworks [98, 94], and probabilistic reasoning using Bayesian networks. Zhao et al. [99] proposed a three-layer diagnostic Bayesian network combining qualitative prior knowledge, fault hypotheses, and sensor-derived symptoms, with fault signatures generated via residual analysis. Atoui et al. [100] presented a conditional Gaussian network (CGN) to fuse model-based residuals and data-driven measurements. Both methods enable probabilistic reasoning under uncertainty, missing data, and model inaccuracies. More recent works such as Bond Graph-CNN [101], Res-HGCN [102], and topology-informed GNNs [103] further demonstrate the value of embedding physical system knowledge into learning-based models. Collectively, these works underscore the importance of hybrid structural and learning-based frameworks for scalable and interpretable fault localization.

FDI has become increasingly important in modern vehicle systems due to their growing complexity, sensor density, and stringent safety requirements [104]. A wide range of FDI methods has been explored, including model-based, data-driven, and hybrid approaches. Observer-based FDI, which identifies faults through estimation residuals, has been widely applied in automotive systems [105, 106, 107]. Hashemi et al. [108] employed UIO to estimate road bank and grade angles under varying friction, while Meléndez-Useros et al. [109] designed dedicated UIOs for each fault type to address decoupling limitations. Benmoussa et al. [110] used bond graph modeling for structured residual generation and isolability analysis in vehicle braking systems. To enhance robustness under external disturbances and uncertainty, Kurniawan and Marton [111] developed an outlier-resilient method combining transfer entropy and Mahalanobis distance. Distributed architectures have also been explored: Safavi et al. [112] proposed a multi-layer framework integrating distributed Kalman filtering, CNNs, and temporal fusion transformers for fault isolation and health forecasting in autonomous vehicles. Deep learning methods have shown strong potential as well. Man et al. [113] introduced a spatiotemporal GCN with attention mechanisms to model dynamic sensor dependencies in electric vehicles, while Dowdeswell et al. [114] used semi-supervised learning with latent representations and signal reconstruction errors to detect unknown sensor faults with minimal labeled data. Although effective, such observer-based designs face scalability issues in systems with numerous fault sources, where assigning separate observers to each scenario becomes impractical. This challenge motivates the need for approaches that exploit structural redundancy and prior knowledge of fault distributions to retain diagnostic performance in large-scale architectures.

Despite the significant progress in centralized, distributed, and hybrid FDI frameworks, several critical challenges remain unaddressed—particularly in the context of large-scale, evolving networked systems. Centralized learning-based approaches often require complete retraining when new fault types or uncertainty sources are introduced, making them imprac-

tical for systems that evolve over time or encounter previously unseen faults. Moreover, the availability of labeled data is often imbalanced or incomplete, especially for rare or newly emerging faults, limiting the effectiveness of supervised global models. Scalability also poses a key challenge: as the number of subsystems and fault sources increases, computational complexity and memory requirements grow rapidly, undermining real-time deployment. Heterogeneity in subsystem dynamics, sensor modalities, and data characteristics further complicates fault isolation, as a one-size-fits-all diagnostic model may fail to generalize across different parts of the system. Additionally, model-based and observer-based techniques, while interpretable and structured, often face limitations in handling unmodeled dynamics, partial observability, and scalability to high-dimensional or interconnected systems. These gaps highlight the need for a scalable, modular, and learning-based FDI framework that can incorporate structural knowledge of heterogeneous subsystem characteristics, handle partially labeled data, and adapt to dynamic fault scenarios without retraining from scratch.

Chapter 3

Graph Representation for Vehicle State Estimation

3.1 Estimation Graph

Estimation Graph. In this study, the estimation graph is defined as a directed graph whose nodes represent local estimators. Structurally, the estimation graph corresponds to a condensation digraph, meaning it is acyclic and contains no strongly connected components (see Figure 3.1). This formulation assumes that any estimation architecture containing strongly connected components—such as coupled or bidirectionally connected estimators—has been transformed into its equivalent condensation digraph.

Each local estimator receives inputs from upstream estimators in higher layers as well as from associated sensor sets. An edge (u, v) in the estimation graph indicates that all relevant information from node u is transmitted to node v . It is further assumed that every estimator depends on its own state estimates, implying the existence of a self-loop at each node. For clarity, self-loops are omitted from the graphical representation in Figure 3.1.

Estimation Path. An estimation path is defined as a subgraph of the estimation graph that contains a single sink node and includes one or more directed walks from source nodes in layer 0 to this sink. This concept follows the general definition of a path (or stream) introduced in Section A.2.1. An illustrative example of an estimation path is shown in Figure 3.1, where the red dashed arrows highlight the flow of information from sensor nodes s_1 and s_2 to a sink estimator E_5 .

Estimation Layer. An estimation layer k is defined as the set of estimator nodes that are located exactly k hops away from the sensor layer (layer 0) in the estimation graph. All sensor nodes are assigned to layer 0, and any estimator that receives direct input from sensor nodes is placed in layer 1. Recursively, any estimator that receives input exclusively from nodes in layer $k - 1$ (or lower) and has not already been assigned to a lower layer is placed in layer k .

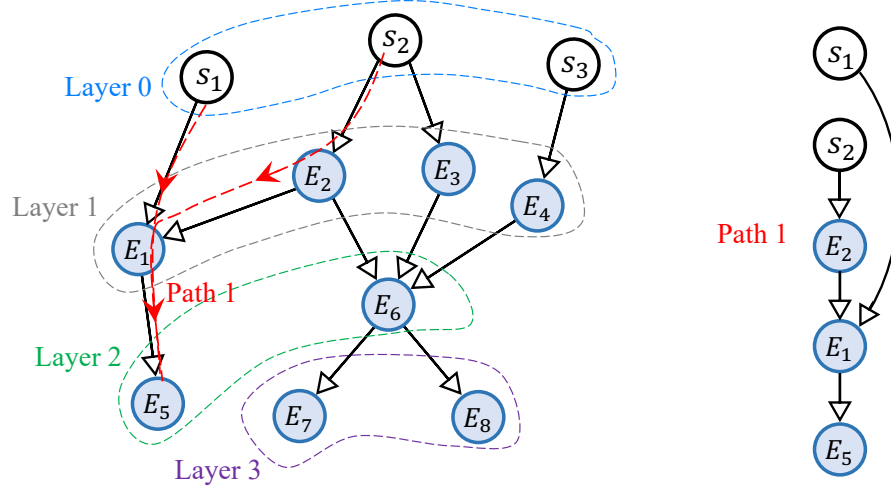


Figure 3.1: Illustration of a general estimation graph. The graph is a directed acyclic graph (DAG), representing the cascaded structure of local estimators. For clarity, self-loops are omitted. s_i represents sensor nodes and E_i denotes local estimators. and The red dashed arrows depict an example estimation path originating from a sensor set in layer 0 and terminating at a sink node in layer 2. Here, s_i represents sensor nodes and E_i denotes local estimators. For the formal definition of a path, see Section A.2.1.

3.2 Vehicle Estimation Graph

Building upon the vehicle motion model presented in Section A.3, this section describes the development of vehicle motion state estimators. An overview of the vehicle estimation graph is shown in Figure 3.2. This graph represents one possible configuration for vehicle state estimation; alternative estimation graphs with different estimator types and connectivity structures can also be constructed. In the illustrated configuration, the estimators are

interconnected in a cascaded structure, where the estimated states from one estimator may serve as inputs to downstream estimators. The following subsections describe each local estimator in the graph.

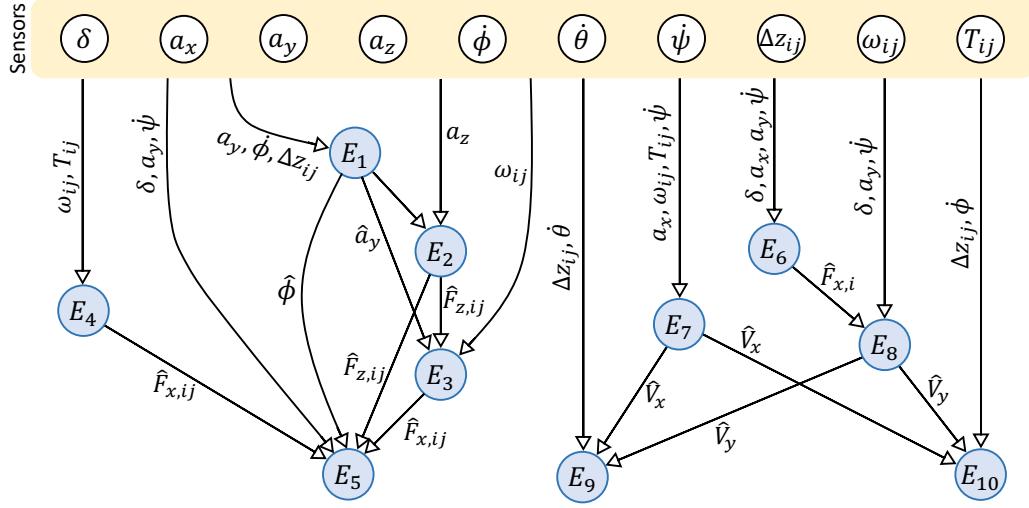


Figure 3.2: Vehicle state estimation graph with local estimator nodes E_1 to E_{10} .

3.2.1 Lateral Load Transfer Estimator (E_1)

This estimator, originally developed in [115], estimates the lateral load transfer to the left, denoted by $\Delta F_{z,l}$, during cornering. This estimator employs a linear Kalman filter with the state vector

$$\mathbf{x} = [\Delta F_{z,l} \quad a_y \quad \dot{a}_y \quad \phi \quad \dot{\phi}]^\top,$$

and the observation vector

$$\mathbf{y} = [a_{y,m} \quad \phi_m \quad \dot{\phi}_m \quad \Delta F_{z,l}]^\top.$$

Note: The subscript $(\cdot)_m$ indicates that the corresponding variable is measured.

The measured lateral acceleration $a_{y,m}$ is expressed as $a_{y,m} = a_y - g \sin \phi$. Given the geometry of roll motion, the roll angle ϕ_m is measured using suspension deflections:

$$\phi_m = -\frac{\Delta z_{fl} - \Delta z_{fr} + \Delta z_{rl} - \Delta z_{rr}}{2b_w} \quad (3.1)$$

where b_w is the track width and Δz_{ij} denotes the suspension deflection at corner ij . The lateral load transfer to the left ΔF_{z_l} is calculated by:

$$\begin{aligned}\Delta F_{z,l} &= F_{z,fl} + F_{z,rl} - F_{z,fr} - F_{z,rr} \\ &= -2\frac{k_\phi}{b_w}\phi_m + 2ma_{y,m}\frac{h_{RC}}{b_w}\end{aligned}\quad (3.2)$$

where k_ϕ is the roll stiffness and h_{RC} is the height of the roll center.

The linear state-space representation of estimator E_1 is given by:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & \frac{2mh_{RC}}{b_w} & 0 & -2\frac{k_f+k_r}{b_w} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & -\frac{mh_R}{I_{xx}+mh_R^2} & 0 & \frac{mgh_R-k_\phi}{I_{xx}+mh_R^2} & \frac{-c_\phi}{I_{xx}+mh_R^2} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & -g & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where I_{xx} is the roll moment of inertia, h_R is the vertical distance from the CG to the roll center, and k_f , k_r and c_ϕ are the front/rear suspension stiffnesses and roll damping coefficient, respectively.

3.2.2 Normal Load Estimator (E_2)

This estimator estimates the vertical tire forces acting at each wheel. The vertical load distribution, obtained by combining the longitudinal and lateral load transfer relations in (A.6) and (A.7), is given by:

$$\begin{cases} F_{z,fl}, F_{z,fr} = \frac{1}{2}m\left(\frac{l_r}{l}g - \frac{h_{CG}}{l}a_x\right) \pm m\left(\frac{l_r}{l}g - \frac{h_{CG}}{l}a_x\right)\frac{h_{CG}}{b_w g}a_y, \\ F_{z,rl}, F_{z,rr} = \frac{1}{2}m\left(\frac{l_f}{l}g + \frac{h_{CG}}{l}a_x\right) \pm m\left(\frac{l_f}{l}g + \frac{h_{CG}}{l}a_x\right)\frac{h_{CG}}{b_w g}a_y, \end{cases}\quad (3.3)$$

Taking the time derivative of (3.3), a process model is formulated with the state vector

$$\mathbf{x} = [F_{z,fl} \quad F_{z,fr} \quad F_{z,rl} \quad F_{z,rr}]^\top,$$

which is modeled as a random walk process with transition matrix $\mathbf{A} = \mathbf{0}$, assuming slowly varying normal forces during the estimation horizon.

The observation vector is defined as:

$$\mathbf{y} = \begin{bmatrix} \widehat{\Delta F}_{z,l} & F_{z,fl} + F_{z,fr} & F_{z,fl} - F_{z,fr} & ma_{z,m} \end{bmatrix}^\top,$$

where $\widehat{\Delta F}_{z,l}$ is obtained from the previous estimator E_1 (Section 3.2.1), the next two outputs are algebraically computed using (3.3), and $a_{z,m}$ is the measured vertical acceleration. This structure satisfies the observability condition for the four-state model and enables accurate estimation of wheel vertical loads.

3.2.3 Longitudinal Tire Force Estimator by Tire Model (E_3)

A discrete-time EKF is developed to jointly estimate the vehicle's longitudinal velocity and longitudinal tire-road forces using a nonlinear tire model and wheel speed measurements. Unlike conventional approaches, it does not require wheel torque inputs, and instead utilizes the Pacejka tire model in conjunction with estimated normal loads from the vertical load estimator (E_2).

The state vector of the estimator is defined as:

$$\mathbf{x} = [V_x \quad F_{x,fl} \quad F_{x,fr} \quad F_{x,rl} \quad F_{x,rr}]^\top.$$

The process model combines the longitudinal dynamics of the vehicle, given in (A.1), with first-order dynamics describing the transient behavior of the tire forces:

$$\begin{cases} \dot{V}_x = \frac{1}{m} \sum_{i \in \{f,r\}} \sum_{j \in \{l,r\}} F_{x,ij}, \\ \dot{F}_{x,ij} = \frac{1}{\tau} (\bar{F}_{x,ij} - F_{x,ij}), \end{cases} \quad (3.4)$$

where $\tau = 0.1$ s is the filter time constant, chosen to suppress high-frequency fluctuations in the estimated force, especially in low-slip conditions where the tire force is highly sensitive to small variations in slip ratio, and $\bar{F}_{x,ij}$ denotes the steady-state tire force computed via the Pacejka tire model (A.13).

The measurement model enforces consistency between the predicted steady-state longitudinal tire forces and the measured wheel speeds. For all four wheels, it is represented as:

$$\mathbf{y} = \begin{bmatrix} f(F_{z,fl}, V_x, \omega_{fl}) - F_{x,fl} \\ f(F_{z,fr}, V_x, \omega_{fr}) - F_{x,fr} \\ f(F_{z,rl}, V_x, \omega_{rl}) - F_{x,rl} \\ f(F_{z,rr}, V_x, \omega_{rr}) - F_{x,rr} \end{bmatrix}, \quad (3.5)$$

where $f(\cdot)$ denotes the nonlinear tire model defined in (A.13). The output vector $\mathbf{y}$ is expected to be zero in steady-state, i.e., $\mathbf{y} = \mathbf{0}$, indicating consistency between the estimated and predicted longitudinal tire forces.

3.2.4 Longitudinal Tire Force Estimator by Wheel Dynamics (E_4)

This estimator leverages wheel rotational dynamics, as described in (A.18), to estimate the longitudinal tire force $F_{x,ij}$ at each wheel. The approach requires the availability of wheel torque measurements $T_{b,ij}$ and $T_{d,ij}$. The state vector for each wheel $ij \in \{fl, fr, rl, rr\}$ is defined as

$$\mathbf{x}_{ij} = [\omega_{ij} \quad \dot{\omega}_{ij} \quad F_{x,ij}]^\top,$$

where ω_{ij} and $\dot{\omega}_{ij}$ denote the angular velocity and angular acceleration of the wheel, respectively. The measurement vector is given by

$$\mathbf{y}_{ij} = [\omega_{ij,m} \quad \dot{\omega}_{ij,m}]^\top,$$

A discrete-time linear Kalman filter is implemented independently for each wheel to estimate the corresponding tire-road interaction force.

3.2.5 Lateral Tire Force Estimator (E_5)

This estimator aims to estimate the lateral tire-road forces acting at each wheel by leveraging the full vehicle lateral dynamics model previously introduced in (A.2). The nonlinear lateral tire-road interaction force is modeled using the Magic Formula (Pacejka) tire model, given by (A.16), where the lateral force $F_{y,ij}$ at each tire is a function of the slip angle α_{ij} , normal load $F_{z,ij}$, and road friction coefficient μ_r .

To account for the transient response of lateral forces, the following first-order relaxation model is used:

$$\dot{F}_{y,ij} = \frac{V_x}{\sigma} (-F_{y,ij} + \bar{F}_{y,ij}), \quad (3.6)$$

where σ is the relaxation length and $\bar{F}_{y,ij}$ is the steady-state lateral tire-road force computed using the Pacejka model.

An Extended Kalman Filter (EKF) developed in [116] is used to estimate the lateral tire forces of a four-wheel vehicle. The state vector for this estimator is defined as

$$\mathbf{x} = [V_y \quad \dot{\psi} \quad F_{y,fl} \quad F_{y,fr} \quad F_{y,rl} \quad F_{y,rr}]^\top.$$

The measurement vector is defined as:

$$\mathbf{y} = [\dot{\psi}_m \quad a_{y,m}]^\top,$$

where $\dot{\psi}_m$ is the measured yaw rate, and measured lateral acceleration $a_{y,m}$ is given by (A.3).

The longitudinal velocity V_x is approximated by averaging the rear wheel speeds, assuming negligible longitudinal slip at the rear axle. Additionally, the front axle longitudinal force $F_{x,f}$ is assumed to be estimated via other estimators and provided as an input to the estimator.

3.2.6 Kinematics-Based Tire Force Estimator (E_6)

This estimator computes the total tire-road forces at the front and rear axles using a kinematics-based approach. The process model assumes a random walk with transition matrix $\mathbf{A} = \mathbf{0}$ to reflect the slowly varying nature of the forces. The state vector is defined as

$$\mathbf{x} = [F_{x,f} \quad F_{x,r} \quad F_{y,f} \quad F_{y,r}]^\top,$$

where $F_{x,f}$, $F_{x,r}$ denote the total longitudinal forces, and $F_{y,f}$, $F_{y,r}$ denote the total lateral forces at the front and rear axles, respectively.

The measurement vector is given as:

$$\mathbf{y} = [ma_{x,m} \quad ma_{y,m} \quad I_{zz}\dot{\psi}_m \quad 0]^\top,$$

The corresponding output matrix $\mathbf{C}$ that relates the state vector to the measurements is defined as:

$$\mathbf{C} = \begin{bmatrix} \cos \delta_f & 1 & -\sin \delta_f & 0 \\ \sin \delta_f & 0 & \cos \delta_f & 1 \\ l_f \sin \delta_f & 0 & l_f \cos \delta_f & -l_r \\ l_f & -l_r & 0 & 0 \end{bmatrix},$$

where δ_f is the front axle steer angle.

A discrete-time linear Kalman filter is employed to estimate the axle-level tire forces. The individual tire forces are subsequently computed by distributing the total axle forces in proportion to the estimated normal load on each wheel, using the formulas (A.6)–(A.7).

3.2.7 Longitudinal Velocity Estimation via Wheel Dynamics (E_7)

An Extended Kalman Filter (EKF) is employed to estimate the vehicle's longitudinal velocity V_x , accounting for wheel slip using wheel torque and speed measurements. The state vector is defined as

$$\mathbf{x} = [V_x \quad \omega_{fl} \quad \omega_{fr} \quad \omega_{rl} \quad \omega_{rr}]^\top,$$

while the measurement vector consists of the measured wheel speeds

$$\mathbf{y} = [\omega_{fl,m} \quad \omega_{fr,m} \quad \omega_{rl,m} \quad \omega_{rr,m}]^\top.$$

The process model incorporates both the longitudinal kinematics of the vehicle and the wheel dynamics, formulated as:

$$\begin{cases} \dot{V}_x = a_{x,m} + \dot{\psi} V_y, \\ \dot{\omega}_{ij} = \frac{1}{I_w} (T_{d,ij} - T_{b,ij} - R_e F_{x,ij}), \end{cases} \quad (3.7)$$

where $F_{x,ij}$ denotes the longitudinal tire force computed using the nonlinear Pacejka tire model introduced in (A.13), and the input torques $T_{d,ij}$ and $T_{b,ij}$ are assumed to be available. In this formulation, the impact of V_y to the longitudinal motion is considered negligible and thus set to zero. The measured longitudinal acceleration $a_{x,m}$ is supplied as an input to the process model.

3.2.8 Lateral Velocity Estimator (E_8)

The lateral velocity V_y is estimated using a single-track (bicycle) vehicle model, where the left and right wheels at each axle are combined, and lateral load transfer effects are neglected. The roll angle ϕ is assumed to be zero in this formulation. The state vector and measurement vector are defined as

$$\mathbf{x} = [V_y \quad \dot{\psi}]^\top, \quad \mathbf{y} = [a_{y,m} \quad \dot{\psi}_m]^\top,$$

where the measured lateral acceleration $a_{y,m}$ is modeled as:

$$a_{y,m} = \frac{1}{m} \left\{ F_{y,f} \cos \delta_f + F_{y,r} + F_{x,f} \sin \delta_f \right\}. \quad (3.8)$$

The lateral dynamics of the vehicle are governed by the single-track model equations:

$$\begin{cases} \dot{V}_y = -V_x\dot{\psi} + \frac{1}{m}\left\{F_{y,f}\cos\delta_f + F_{y,r} + F_{x,f}\sin\delta_f\right\}, \\ \ddot{\psi} = \frac{1}{I_{zz}}\left\{l_f(F_{y,f}\cos\delta_f + F_{x,f}\sin\delta_f) - l_rF_{y,r}\right\}. \end{cases} \quad (3.9)$$

where the lateral tire forces $F_{y,f}$ and $F_{y,r}$ are computed using the nonlinear Pacejka tire model introduced in (A.16), while the front axle longitudinal force $F_{x,f}$ is obtained from the estimator described in Section 3.2.6.

To ensure numerical stability and improve the conditioning of the EKF, the lateral force states $F_{y,ij}$ are expressed in kilonewtons (kN), such that all state variables maintain a comparable order of magnitude.

3.2.9 Road Grade Angle Estimator (E_g)

This section presents the design of an UIO [117] for estimating the road grade angle as an unknown external disturbance.

Unknown Input Observer

The system is modeled with an additive unknown input that represents the disturbances as system uncertainties. The state-space model is described as:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{R}\mathbf{f} \\ \mathbf{y} = \mathbf{C}\mathbf{x} \end{cases} \quad (3.10)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^r$ is the known control input, and $\mathbf{y} \in \mathbb{R}^m$ is the measured output vector. The term $\mathbf{f} \in \mathbb{R}^q$ represents an unknown input (or disturbance), and $\mathbf{R}$ is a known matrix that characterizes how the unknown input affects the system. The state transition matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, input matrix $\mathbf{B} \in \mathbb{R}^{n \times r}$, and output matrix $\mathbf{C} \in \mathbb{R}^{m \times n}$ are assumed to be known.

Definition 1 (Unknown Input Observer). *An observer is said to be an unknown input observer (UIO) for the system described by (3.10) if its estimation error vector $\mathbf{e}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t)$ asymptotically converges to zero, independent of the influence of unknown disturbances $\mathbf{f}(t)$ present in the system.*

The structure of the unknown input observer for system (3.10) is defined as:

$$\begin{cases} \dot{\mathbf{z}} = \mathbf{F}\mathbf{z} + \mathbf{T}\mathbf{B}\mathbf{u} + \mathbf{K}\mathbf{y} \\ \hat{\mathbf{x}} = \mathbf{z} + \mathbf{H}\mathbf{y} \end{cases} \quad (3.11)$$

where $\hat{\mathbf{x}} \in \mathbb{R}^n$ denotes the estimated state vector, and $\mathbf{z} \in \mathbb{R}^n$ is the internal state of the observer. The matrices $\mathbf{F}, \mathbf{T}, \mathbf{K}, \mathbf{H}$ are design parameters selected to achieve decoupling from unknown inputs and to meet desired observer performance. The observer receives the system input $\mathbf{u}$ and output $\mathbf{y}$, and generates an estimate $\hat{\mathbf{x}}$ of the system state. The schematic diagram illustrating this observer structure is shown in Figure 3.3.

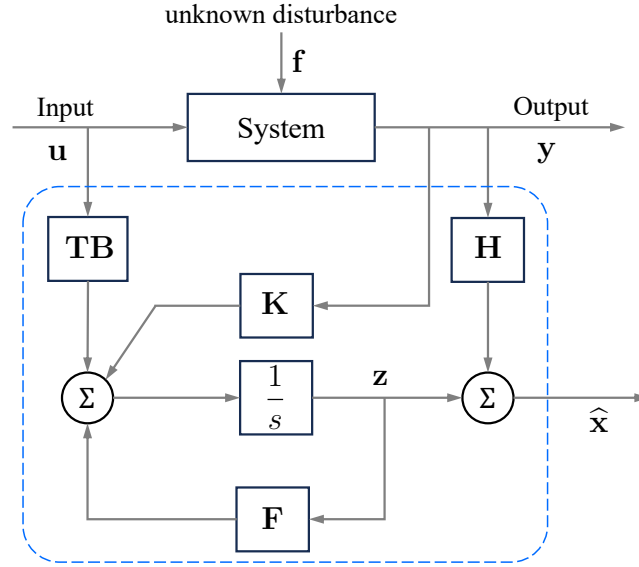


Figure 3.3: Unknown Input Observer (UIO) structure.

The estimation error of the unknown input observer, defined as $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$, evolves according to the following dynamics:

$$\begin{aligned} \dot{\mathbf{e}} = & (\mathbf{A} - \mathbf{H}\mathbf{C}\mathbf{A} - \mathbf{K}_1\mathbf{C})\mathbf{e} + [\mathbf{F} - (\mathbf{A} - \mathbf{H}\mathbf{C}\mathbf{A} - \mathbf{K}_1\mathbf{C})]\mathbf{z} \\ & + [\mathbf{K}_2 - (\mathbf{A} - \mathbf{H}\mathbf{C}\mathbf{A} - \mathbf{K}_1\mathbf{C})\mathbf{H}]\mathbf{y} \\ & + [\mathbf{T} - (\mathbf{I} - \mathbf{H}\mathbf{C})]\mathbf{B}\mathbf{u} + (\mathbf{H}\mathbf{C} - \mathbf{I})\mathbf{R}\mathbf{f} \end{aligned} \quad (3.12)$$

where $\mathbf{I}$ is an identity matrix, and the gain matrix $\mathbf{K}$ is defined as $\mathbf{K} = \mathbf{K}_1 + \mathbf{K}_2$.

To achieve perfect decoupling from the unknown input $\mathbf{f}(t)$, the observer parameters must satisfy the following constraints:

$$(\mathbf{H}\mathbf{C} - \mathbf{I})\mathbf{R} = \mathbf{0}, \quad (3.13)$$

$$\mathbf{T} = \mathbf{I} - \mathbf{H}\mathbf{C}, \quad (3.14)$$

$$\mathbf{F} = \mathbf{A} - \mathbf{H}\mathbf{C}\mathbf{A} - \mathbf{K}_1\mathbf{C}, \quad (3.15)$$

$$\mathbf{K}_2 = \mathbf{F}\mathbf{H}. \quad (3.16)$$

Under these design constraints, the error dynamics simplify to:

$$\dot{\mathbf{e}}(t) = \mathbf{F}\mathbf{e}(t). \quad (3.17)$$

If all eigenvalues of $\mathbf{F}$ lie in the left-half complex plane, the estimation error $\mathbf{e}(t)$ asymptotically converges to zero. This guarantees that the observer described by (3.11) acts as a valid unknown input observer for the system (3.10), as it satisfies the definition provided in Definition 1.

Theorem 1. *A UIO exists for the system defined in (3.10) if and only if the following two conditions are satisfied (see [117]):*

1. $\text{rank}(\mathbf{C}\mathbf{R}) = \text{rank}(\mathbf{R})$
2. The pair $(\mathbf{C}, \mathbf{A}_1)$ is detectable¹, where

$$\mathbf{A}_1 = \mathbf{A} - \mathbf{R} [(\mathbf{C}\mathbf{R})^\top \mathbf{C}\mathbf{R}]^{-1} (\mathbf{C}\mathbf{R})^\top \mathbf{C}\mathbf{A}$$

A special solution for $\mathbf{H}$ that satisfies the condition (3.13) is given by:

$$\mathbf{H}^* = \mathbf{R} [(\mathbf{C}\mathbf{R})^\top \mathbf{C}\mathbf{R}]^{-1} (\mathbf{C}\mathbf{R})^\top. \quad (3.18)$$

In this case, the transition matrix $\mathbf{F}$ for the estimation error will be:

$$\mathbf{F} = \mathbf{A} - \mathbf{H}\mathbf{C}\mathbf{A} - \mathbf{K}_1\mathbf{C} = \mathbf{A}_1 - \mathbf{K}_1\mathbf{C}, \quad (3.19)$$

To guarantee that the estimation error $\mathbf{e}(t)$ converges to zero asymptotically, the matrix $\mathbf{F}$ must be stable. When the pair $(\mathbf{C}, \mathbf{A}_1)$ is observable, the gain matrix $\mathbf{K}_1$ can be designed using a standard pole placement technique to place the eigenvalues of $\mathbf{F}$ in the left half-plane, thereby ensuring the stability of the estimation error dynamics.

The following steps summarize the procedure for designing a UIO for the system described in for the system described in (3.10):

¹A pair $(\mathbf{C}, \mathbf{A})$ is detectable when all unobservable modes for this pair are stable. Detectability is a weaker condition than observability.

1. Ensure that the rank condition holds, i.e., $\text{rank}(\mathbf{CR}) = \text{rank}(\mathbf{R})$. If not satisfied, a UIO does not exist and the procedure should be terminated.
2. : Compute observer matrices $\mathbf{H}$, $\mathbf{T}$, and $\mathbf{A}_1$:

$$\mathbf{H} = \mathbf{R} [(\mathbf{CR})^\top \mathbf{CR}]^{-1} (\mathbf{CR})^\top, \quad \mathbf{T} = \mathbf{I} - \mathbf{HC}, \quad \mathbf{A}_1 = \mathbf{TA}.$$

3. If the pair $(\mathbf{C}, \mathbf{A}_1)$ is observable, a UIO exists and the observer gain $\mathbf{K}_1$ can be designed via pole placement applied to the system matrix $\mathbf{F} = \mathbf{A}_1 - \mathbf{K}_1 \mathbf{C}$.

For cases where the pair $(\mathbf{C}, \mathbf{A}_1)$ is not observable but merely detectable, more advanced procedures involving system decomposition are required. Readers are referred to [117] for a detailed treatment of such scenarios.

4. Compute $\mathbf{F}$ and $\mathbf{K}$:

$$\mathbf{F} = \mathbf{A}_1 - \mathbf{K}_1 \mathbf{C}, \quad \mathbf{K}_2 = \mathbf{FH}, \quad \mathbf{K} = \mathbf{K}_1 + \mathbf{K}_2.$$

Finally, the unknown disturbance $\mathbf{f}$ can be estimated by differentiating the output equation [118]. The time derivative output model yields:

$$\dot{\mathbf{y}} = \mathbf{C}\dot{\mathbf{x}} = \mathbf{CAx} + \mathbf{CBu} + \mathbf{CRf}.$$

Substituting the estimated state $\hat{\mathbf{x}}$ into the expression and rearranging terms, the disturbance estimate is computed as:

$$\hat{\mathbf{f}} = (\mathbf{CR})^\dagger [\dot{\mathbf{y}} - \mathbf{CA}\hat{\mathbf{x}} - \mathbf{CBu}], \quad (3.20)$$

where $(\mathbf{CR})^\dagger$ denotes the left inverse² of $\mathbf{CR}$.

Unknown Input Observer Design for Pitch Dynamics

The vehicle pitch motion model given in (A.9) is considered for estimating the pitch angle θ and road grade angle θ_r as the unknown disturbance. The total input term $-a_x + g \sin(\theta + \theta_r)$ is treated as an unknown disturbance $\mathbf{f}_d$. The measurement vector is defined as $\mathbf{y} = [\theta_m \quad \dot{\theta}_m]^\top$, with the output matrix $\mathbf{C} = \mathbf{I}_{2 \times 2}$. The UIO is designed by following the standard procedure, summarized as follows.

The system matrices are defined as:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{k_\theta}{I_{yy} + m_s h_P^2} & -\frac{c_\theta}{I_{yy} + m_s h_P^2} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0 \\ \frac{m_s h_P}{I_{yy} + m_s h_P^2} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

²The left inverse of a full column-rank matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ with $n \geq m$ is given by $\mathbf{A}^\dagger = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$.

Step 1: The rank condition

$$\text{rank}(\mathbf{CR}) = \text{rank}(\mathbf{R})$$

is verified as $\mathbf{C}$ is unity matrix and both ranks equal to 1. Therefore, the existence condition for a UIO is met.

Step 2: The matrices $\mathbf{H}$, $\mathbf{T}$, and $\mathbf{A}_1$ are computed as:

$$\mathbf{H} = \mathbf{R} [(\mathbf{CR})^\top \mathbf{CR}]^{-1} (\mathbf{CR})^\top = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{T} = \mathbf{I} - \mathbf{HC} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{A}_1 = \mathbf{TA} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Step 3: The pair $(\mathbf{C}, \mathbf{A}_1)$ is observable, which ensures the possibility of placing the observer poles arbitrarily. Hence, the observer gain $\mathbf{K}_1$ is designed via pole placement. The desired eigenvalues are set to -5 and -2 , leading to the gain matrix:

$$\mathbf{K}_1 = \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix}.$$

Step 4: The matrices $\mathbf{F}$, $\mathbf{K}_2$, and the total observer gain $\mathbf{K}$ are computed as follows:

$$\begin{aligned} \mathbf{F} &= \mathbf{A}_1 - \mathbf{K}_1 \mathbf{C} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} -5 & 1 \\ 0 & -2 \end{bmatrix}, \\ \mathbf{K}_2 &= \mathbf{FH} = \begin{bmatrix} -5 & 1 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix}, \\ \mathbf{K} &= \mathbf{K}_1 + \mathbf{K}_2 = \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix} = \begin{bmatrix} 5 & 1 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

Once the observer is designed, the unknown disturbance $\mathbf{f}_d$ can be estimated by:

$$\begin{aligned} \hat{\mathbf{f}}_d &= (\mathbf{CR})^\dagger [\dot{\mathbf{y}} - \mathbf{CA}\hat{\mathbf{x}} - \mathbf{CB}\mathbf{u}] \\ &= \begin{bmatrix} 0 & \frac{I_{yy} + m_s h_P^2}{m_s h_P} \end{bmatrix} \left(\begin{bmatrix} \dot{\theta}_m \\ \ddot{\theta}_m \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -\frac{k_\theta}{I_{yy} + m_s h_P^2} & -\frac{c_\theta}{I_{yy} + m_s h_P^2} \end{bmatrix} \begin{bmatrix} \hat{\theta} \\ \hat{\dot{\theta}} \end{bmatrix} \right) \\ &= \frac{I_{yy} + m_s h_P^2}{m_s h_P} \left(\ddot{\theta}_m + \frac{k_\theta}{I_{yy} + m_s h_P^2} \hat{\theta} + \frac{c_\theta}{I_{yy} + m_s h_P^2} \hat{\dot{\theta}} \right). \end{aligned}$$

Here, the total unknown disturbance is defined as $\mathbf{f}_d = -a_x + g \sin(\theta + \theta_r)$. The longitudinal acceleration a_x can be approximated using $\hat{a}_x = \hat{V}_x + \hat{V}_y \dot{\psi}_m$. By substituting the estimated disturbance $\hat{\mathbf{f}}_d$ and the estimated acceleration $\hat{a}_x$, an estimate of the road grade angle $\hat{\theta}_r$ can be computed. Under steady-state conditions, where the pitch dynamics are negligible (i.e., $\dot{\theta} \approx 0$ and $\ddot{\theta} \approx 0$), the road grade angle is estimated by:

$$\hat{\theta}_r = \arcsin \left(\frac{k_\theta}{m_s h_P g} \hat{\theta} + \frac{1}{g} (\hat{V}_x + \hat{V}_y \dot{\psi}_m) \right) - \hat{\theta}. \quad (3.21)$$

This completes the design and implementation of the unknown input observer for the pitch dynamics model. The proposed observer enables the estimation of both the vehicle pitch angle and the unknown road grade angle.

3.2.10 Road Bank Angle Estimator (E_{10})

The same procedure described in Section 3.2.9 using a UIO is adopted here for the estimation of the road bank angle. The vehicle roll motion model given in (A.10) is considered for estimating the roll angle ϕ and road grade angle ϕ_r as the unknown disturbance. The total input term $a_y + g \sin(\phi + \phi_r)$ is treated as an unknown disturbance $\mathbf{f}_d$. The measurement vector is defined as $\mathbf{y} = [\phi_m \quad \dot{\phi}_m]^\top$, with the output matrix $\mathbf{C} = \mathbf{I}_{2 \times 2}$. The UIO is designed by following the standard procedure, summarized as follows.

The system matrices are defined as:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{k_\phi}{I_{xx} + m_s h_P^2} & -\frac{c_\phi}{I_{xx} + m_s h_R^2} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0 \\ \frac{m_s h_R}{I_{xx} + m_s h_P^2} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Step 1: The rank condition

$$\text{rank}(\mathbf{CR}) = \text{rank}(\mathbf{R})$$

is verified as $\mathbf{C}$ is unity matrix and both ranks equal to 1. Therefore, the existence condition for a UIO is met.

Step 2: The matrices $\mathbf{H}$, $\mathbf{T}$, and $\mathbf{A}_1$ are computed as:

$$\mathbf{H} = \mathbf{R} [(\mathbf{CR})^\top \mathbf{CR}]^{-1} (\mathbf{CR})^\top = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{T} = \mathbf{I} - \mathbf{HC} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{A}_1 = \mathbf{TA} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Step 3: The pair $(\mathbf{C}, \mathbf{A}_1)$ is observable, which ensures the possibility of placing the observer poles arbitrarily. Hence, the observer gain $\mathbf{K}_1$ is designed via pole placement. The desired eigenvalues are set to -5 and -2 , leading to the gain matrix:

$$\mathbf{K}_1 = \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix}.$$

Step 4: The matrices $\mathbf{F}$, $\mathbf{K}_2$, and the total observer gain $\mathbf{K}$ are computed as follows:

$$\begin{aligned} \mathbf{F} &= \mathbf{A}_1 - \mathbf{K}_1 \mathbf{C} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} -5 & 1 \\ 0 & -2 \end{bmatrix}, \\ \mathbf{K}_2 &= \mathbf{F} \mathbf{H} = \begin{bmatrix} -5 & 1 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix}, \\ \mathbf{K} &= \mathbf{K}_1 + \mathbf{K}_2 = \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix} = \begin{bmatrix} 5 & 1 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

Once the observer is designed, the unknown disturbance $\mathbf{f}_d$ can be estimated by:

$$\begin{aligned} \hat{\mathbf{f}}_d &= (\mathbf{C}\mathbf{R})^\dagger [\dot{\mathbf{y}} - \mathbf{C}\mathbf{A}\hat{\mathbf{x}} - \mathbf{C}\mathbf{B}\mathbf{u}] \\ &= \begin{bmatrix} 0 & \frac{I_{xx} + m_s h_R^2}{m_s h_R} \end{bmatrix} \left(\begin{bmatrix} \dot{\phi}_m \\ \dot{\phi}_m \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -\frac{k_\phi}{I_{xx} + m_s h_R^2} & -\frac{c_\phi}{I_{xx} + m_s h_R^2} \end{bmatrix} \begin{bmatrix} \hat{\phi} \\ \hat{\dot{\phi}} \end{bmatrix} \right) \\ &= \frac{I_{xx} + m_s h_R^2}{m_s h_R} \left(\ddot{\phi}_m + \frac{k_\phi}{I_{xx} + m_s h_R^2} \hat{\phi} + \frac{c_\phi}{I_{xx} + m_s h_R^2} \hat{\dot{\phi}} \right). \end{aligned}$$

Here, the total unknown disturbance is defined as $\mathbf{f}_d = a_y + g \sin(\phi + \phi_r)$. The later acceleration a_y can be approximated using $\hat{a}_y = \hat{V}_y - \hat{V}_x \dot{\psi}_m$. By substituting the estimated disturbance $\hat{\mathbf{f}}_d$ and the estimated acceleration $\hat{a}_y$, an estimate of the road bank angle $\hat{\phi}_r$ can be computed. Under steady-state conditions, where the roll dynamics are negligible (i.e., $\dot{\phi} \approx 0$ and $\ddot{\phi} \approx 0$), the road bank angle is estimated by:

$$\hat{\phi}_r = \arcsin \left(\frac{k_\phi}{m_s h_R g} \hat{\phi} - \frac{1}{g} (\hat{V}_y - \hat{V}_x \dot{\psi}_m) \right) - \hat{\phi}. \quad (3.22)$$

This completes the design and implementation of the unknown input observer for the roll dynamics model. The proposed observer enables the estimation of both the vehicle roll angle and the unknown road bank angle.

3.3 Structural Analysis

The structural analysis of the estimation graph, illustrated in Figure 3.2, provides a systematic understanding of how different sensor faults and external disturbances propagate through estimation pathways to generate residuals designed for fault isolation. This analysis plays a critical role in fault diagnosis by identifying structurally sensitive residuals that are theoretically capable of responding to specific fault sources. The goal is to leverage both physical and functional redundancies in the estimation architecture to construct a structurally grounded fault signature matrix.

To facilitate this, Figure 3.4 enumerates the candidate estimation paths extracted from the estimation graph. Each path represents a feasible sequence of local estimators that can estimate intended states given the available sensor set. These paths originate from the sensors set node S and traverse through a series of estimators E_i . The selected paths capture the redundancy and modularity inherent in the estimation structure, enabling reconfigurable estimation strategies under various fault conditions. Table 3.1 encodes the structural dependencies between these estimation paths and various fault sources. An entry marked 'x' denotes that the corresponding fault structurally influences the estimation path, either directly or indirectly, based on the vehicle's physical and functional interconnections.

In the considered vehicle system, faults are categorized into three primary classes:

- **Sensor and actuator faults:** including bias, drift, and excessive noise in sensor measurements δ , a_x , a_y , a_z , $\dot{\phi}$, $\dot{\psi}$, Δz_{ij} , ω_{ij} , T_{ij} .
- **Unknown external disturbances:** such as wind gusts and variations in road grade and bank angles, represented by $a_{y,\text{wind}}$, θ_{road} , ϕ_{road} .
- **Parametric model errors:** arising from inaccurate or time-varying model parameters, such as the road friction coefficient μ_{road} .

Residual signals are then generated by exploiting estimation redundancies, as discussed in Subsection 3.3.3. The relationship between the generated residuals and the estimation paths is summarized in Table 3.2, where an 'x' denotes that a given residual is generated by, or structurally related to, the corresponding path.

Finally, the fault signature matrix is derived by combining the structural dependencies of faults on estimation paths with the mapping of residuals to those paths. This integration is formalized through a Boolean mapping algorithm, as described below. The resulting matrix, shown in Table 3.3, links each residual to the set of faults it is structurally sensitive

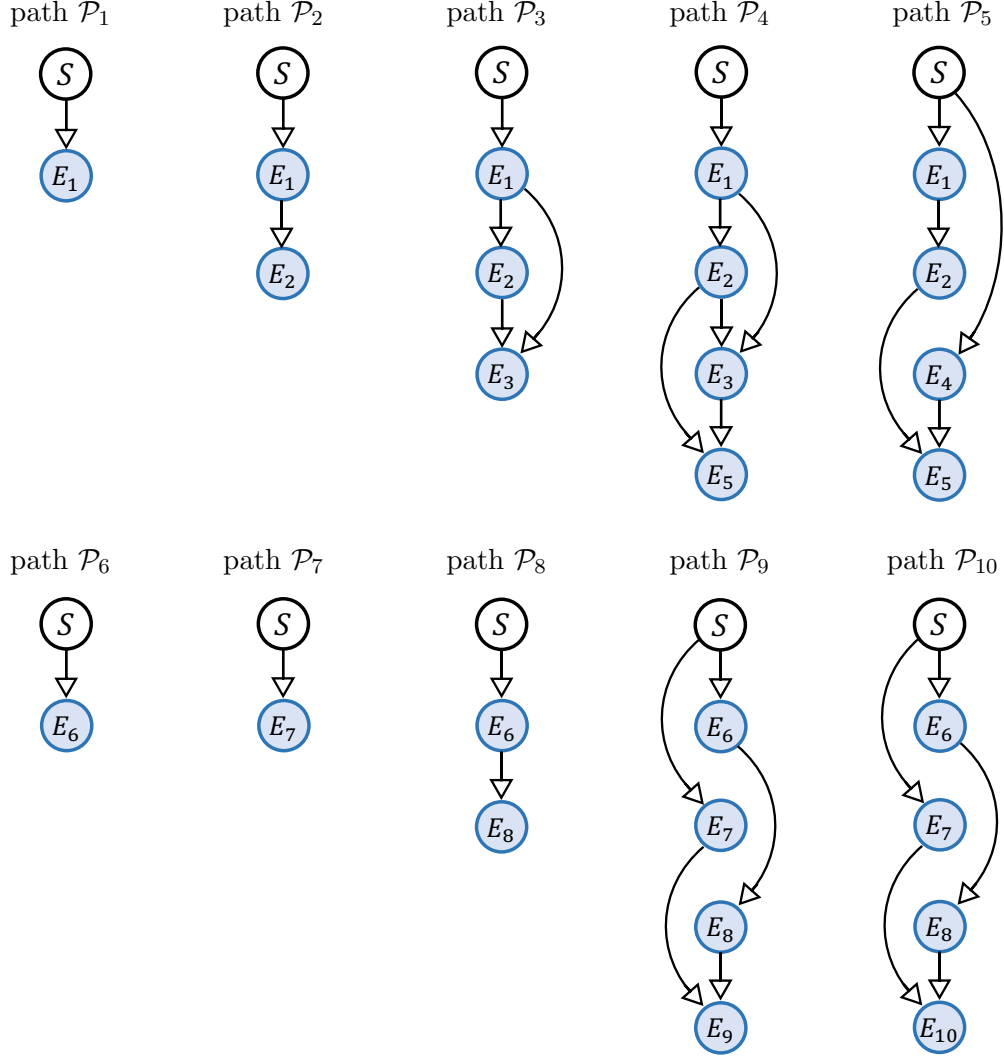


Figure 3.4: Candidate state estimation paths extracted from the vehicle estimation graph shown in Figure 3.2. The node labeled S represents the sensor set available for each estimation path. A directed walk from S to a node E_i implies that the required sensor information for that estimator is accessible. Each path corresponds to a feasible estimation sequence composed of local estimators. Depending on the target states to be estimated, multiple sub-paths may be derived from each full path.

Table 3.1: Binary structural matrix $\mathbf{S}$: Rows represent estimation paths $\mathcal{P}_i$; columns represent faults f_i . Entry of 'x' indicates structural dependency.

Paths	Fault sources												
	δ	a_x	a_y	a_z	$\dot{\phi}$	$\dot{\psi}$	Δz_{ij}	ω_{ij}	T_{ij}	θ_{road}	ϕ_{road}	$a_{y,\text{wind}}$	μ_{road}
$\mathcal{P}_1$			x		x		x						
$\mathcal{P}_2$			x	x	x		x						
$\mathcal{P}_3$			x	x	x		x	x					x
$\mathcal{P}_4$	x		x	x	x	x	x	x					x
$\mathcal{P}_5$	x		x	x	x	x	x	x	x			x	x
$\mathcal{P}_6$	x	x	x			x						x	
$\mathcal{P}_7$		x				x		x	x				x
$\mathcal{P}_8$	x	x	x			x						x	x
$\mathcal{P}_9$	x	x	x			x	x	x	x	x			x
$\mathcal{P}_{10}$	x	x	x		x	x	x	x	x		x	x	x

to. Importantly, the symbol 'x' denotes only the structural reachability of the fault to the residual; the actual sensitivity may be limited due to weak observability, measurement noise, or redundancy among residuals. Consequently, additional refinement techniques—such as feature selection or quantitative sensitivity analysis—may be employed to eliminate insensitive or redundant residuals, thereby constructing a more effective fault signature matrix for diagnostic purposes.

The algorithmic procedure to derive fault signature matrix is outlined as follows:

1. Let $\mathcal{F} = \{f_1, f_2, \dots, f_m\}$ denote the set of faults, and $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n\}$ denote the set of estimation paths.
2. Define a binary structural dependency matrix $\mathbf{S} \in \{0, 1\}^{n \times m}$, where the entry $S_{\ell j} = 1$ indicates that fault f_j structurally affects estimation path $\mathcal{P}_\ell$.
3. Let $\mathcal{R} = \{r_1, r_2, \dots, r_k\}$ be the set of residuals, and define a residual-path incidence matrix $\mathbf{\Gamma} \in \{0, 1\}^{k \times n}$, where $\Gamma_{i\ell} = 1$ indicates that residual r_i is generated or affected by estimation path $\mathcal{P}_\ell$.

Table 3.2: Incidence matrix $\mathbf{\Gamma}$: Rows represent residuals r_i ; columns represent the estimation paths $\mathcal{P}_i$. Entry of 'x' indicates structural dependency.

Residual	Variable	Estimation paths									
		$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	$\mathcal{P}_7$	$\mathcal{P}_8$	$\mathcal{P}_9$	$\mathcal{P}_{10}$
r_1	$\tilde{a}_{x,1}$					x					
r_2	$\tilde{a}_{x,2}$							x	x		
r_3	$\tilde{a}_{y,1}$				x	x					
r_4	$\tilde{a}_{y,2}$								x		
r_5	$\tilde{a}_z$		x							x	x
r_6	$\tilde{\dot{\phi}}_1$		x								
r_7	$\tilde{\dot{\phi}}_2$										x
r_8	$\tilde{\dot{\psi}}_1$								x		
r_9	$\tilde{\dot{\psi}}_2$					x					
r_{10}	$\tilde{\dot{\psi}}_3$								x		
r_{11-14}	$\widetilde{\Delta z_{ij}}$										
r_{15-18}	$\tilde{\omega}_{ij}$							x			
r_{19}	$\tilde{V}_x$			x				x		x	
r_{20}	$\tilde{V}_y$					x			x		
r_{21}	$\tilde{F}_{x,f}$				x		x				
r_{22}	$\tilde{F}_{y,f}$					x	x				
r_{23}	$\tilde{\theta}_r$									x	
r_{24}	$\tilde{\phi}_r$										x

4. The fault signature matrix $\Xi \in \{0, 1\}^{k \times m}$ is then computed by the Boolean matrix product:

$$\Xi = \Gamma \cdot \mathbf{S}, \quad (3.23)$$

where the Boolean matrix multiplication is defined using logical AND ($\wedge$) and logical OR ($\vee$). Specifically, each entry Ξ_{ij} is computed as:

$$\Xi_{ij} = \bigvee_{\ell=1}^n (\Gamma_{i\ell} \wedge S_{\ell j}), \quad (3.24)$$

meaning that residual r_i is structurally dependent on fault f_j if there exists at least one estimation path $\mathcal{P}_\ell$ such that:

- $\Gamma_{i\ell} = 1$: residual r_i is generated by, or related to, path $\mathcal{P}_\ell$, and
 - $S_{\ell j} = 1$: fault f_j affects path $\mathcal{P}_\ell$.
5. Finally, the binary matrix Ξ is converted to a symbolic matrix by replacing all non-zero entries with the marker ' $\mathbf{x}$ ', indicating the structural influence of each fault on the corresponding residual.

This formulation ensures that the derived fault signature matrix reflects the causal propagation of fault effects through the estimation graph to the residuals.

3.3.1 Structural Fault Isolability Analysis

The structural fault signature matrix Ξ in Table 3.3 encodes the binary structural relationships between residuals r_i and fault sources f_j . Each column ξ_j represents the structural signature of a fault f_j , indicating which residuals are theoretically sensitive to that fault.

For the matrix $\Xi \in \mathbb{R}^{n_r \times n_f}$ to enable *full fault isolability*, all columns of Ξ must be pairwise distinct. This implies that for any pair of faults f_j and f_k , their corresponding structural signatures must differ:

$$\xi_j \neq \xi_k, \quad \forall j \neq k.$$

Under this condition, each fault produces a unique pattern across the residuals, enabling unambiguous fault isolation based purely on structural observability. From an algebraic perspective, if Ξ is binary-valued, then full isolability further implies that the rank of Ξ should be n_f .

In the matrix Ξ shown in Table 3.3, there are 13 fault sources and 24 residuals. A detailed inspection of the columns reveals that all fault signatures are pairwise distinct, except for the two columns associated with ψ and μ_{road} , which are structurally identical. As a result, all faults are structurally isolable *except* for this pair. This means that faults in ψ and μ_{road} will produce identical residual patterns, making it impossible to distinguish between them based on structural information alone.

While structural isolability provides a theoretical foundation for fault isolation, practical isolability further depends on the magnitude of fault sensitivity, sensor noise, unmodeled dynamics, and detection thresholds. Thus, structural isolability is a *necessary but not sufficient* condition for successful fault isolation in practice. Further analysis using data-driven validation or sensitivity quantification may be required to confirm isolability under realistic conditions.

3.3.2 Correlation Analysis for Residual Selection

The structural fault signature matrix presented in Table 3.3 represents the binary structural dependencies between residuals and potential fault sources, derived from the estimation graph topology and path-residual mappings. In this matrix, a non-zero entry indicates that a fault can propagate, in principle, to a residual through a structural path. However, such structural connectivity does not necessarily imply a significant or informative fault signature.

In practice, the influence of a structurally connected fault on a residual may be negligible due to factors such as low sensitivity, model redundancy, weak observability, or measurement noise. Moreover, residuals generated by different estimation paths may exhibit high correlation or redundancy, resulting in overlapping or uninformative diagnostic content. This redundancy becomes particularly problematic in data-driven fault detection and diagnosis (FDD) methods, where the input feature set must be both compact and informative to ensure computational efficiency and high classification accuracy.

Therefore, the structurally-derived matrix in Table 3.3 should be treated as an initial estimate of the fault-residual relationships. Further refinement is essential to improve its diagnostic effectiveness. One promising approach involves applying correlation analysis to the residuals to assess statistical dependencies among them. In particular, the Pearson correlation coefficient is computed between each pair of residual signals:

$$\rho_{i,j} = \frac{\text{Cov}(r_i, r_j)}{\sigma_{r_i} \sigma_{r_j}}, \quad (3.25)$$

Table 3.3: Fault Signature Matrix: Rows represent residuals r_i ; columns represent faults f_j . Entry of 'x' indicates structural dependency.

Residual	Variable	Fault sources												
		δ	a_x	a_y	a_z	$\dot{\phi}$	$\dot{\psi}$	Δz_{ij}	ω_{ij}	T_{ij}	θ_{road}	ϕ_{road}	$a_{y,\text{wind}}$	μ_{road}
r_1	$\tilde{a}_{x,1}$	x	x	x	x	x	x	x	x	x			x	x
r_2	$\tilde{a}_{x,2}$	x	x	x			x		x	x			x	x
r_3	$\tilde{a}_{y,1}$	x		x	x	x	x	x	x	x			x	x
r_4	$\tilde{a}_{y,2}$	x	x	x			x						x	x
r_5	$\tilde{a}_z$	x	x	x	x	x	x	x	x	x	x	x	x	x
r_6	$\tilde{\dot{\phi}}_1$			x	x	x		x						
r_7	$\tilde{\dot{\phi}}_2$	x	x	x		x	x	x	x	x		x	x	x
r_8	$\tilde{\dot{\psi}}_1$	x	x	x			x						x	x
r_9	$\tilde{\dot{\psi}}_2$	x		x	x	x	x	x	x	x			x	x
r_{10}	$\tilde{\dot{\psi}}_3$	x	x	x			x						x	x
r_{11-14}	$\tilde{\Delta z_{ij}}$							x						
r_{15-18}	$\tilde{\omega}_{ij}$		x				x		x	x				x
r_{19}	$\tilde{V}_x$	x	x	x	x	x	x	x	x	x	x			x
r_{20}	$\tilde{V}_y$	x	x	x	x	x	x	x	x	x			x	x
r_{21}	$\tilde{F}_{x,f}$	x	x	x	x	x	x	x	x				x	x
r_{22}	$\tilde{F}_{y,f}$	x	x	x	x	x	x	x	x	x			x	x
r_{23}	$\tilde{\theta}_r$	x	x	x			x	x	x	x	x			x
r_{24}	$\tilde{\phi}_r$	x	x	x		x	x	x	x	x		x	x	x

where $\text{Cov}(r_i, r_j)$ denotes the covariance between residuals r_i and r_j , and σ_{r_i} , σ_{r_j} are their standard deviations. Residual pairs with high absolute correlation (e.g., $|\rho_{i,j}| > 0.9$) are considered redundant. For each fault, this analysis is applied to select a subset of residuals with low mutual correlation, thereby reducing redundancy while preserving fault-relevant information. This yields a more compact and informative residual set, improving the efficiency and robustness of data-driven FDD frameworks.

3.3.3 Residual Generation

In fault diagnosis, a residual is defined as the inconsistency between the actual behavior of the system and its predicted behavior derived from a mathematical model. Residuals serve as key diagnostic signals: they are expected to remain close to zero under fault-free conditions and exhibit systematic deviations in the presence of faults. This fundamental principle underpins the FDD process.

Within the vehicle estimation graph introduced in Figure 3.2, each local estimator utilizes its own internal model and available sensor measurements to compute residual signals. These residuals, also referred to as *fault signatures*, result from the physical and functional redundancies embedded in the system design.

Two primary types of residuals are employed:

1. **Measurement residuals**, defined as

$$\tilde{y} = y - \hat{y},$$

where y denotes the actual sensor measurement, and $\hat{y}$ is the predicted output computed using the estimator's internal model. These residuals directly capture the discrepancy between measured and estimated outputs and are particularly effective for detecting sensor faults.

2. **State residuals**, defined as

$$\tilde{x} = \hat{x}^{(1)} - \hat{x}^{(2)},$$

where $\hat{x}^{(1)}$ and $\hat{x}^{(2)}$ are independent estimates of the same state variable obtained from different estimation paths (i.e., different subsets of sensors or different estimator nodes). State residuals are useful for cross-validating estimators and detecting inconsistencies due to faulty sensors, actuators, or model discrepancies.

Under nominal (fault-free) conditions, both types of residuals are expected to be zero-mean and bounded, primarily influenced by sensor noise, unmodeled dynamics, and parameter uncertainties. In the presence of faults, however, residuals often exhibit persistent deviations, abrupt changes, or biased trends, thereby serving as indicators of abnormal system behavior. These deviations can be leveraged to trigger fault detection and contribute to fault isolation and identification processes.

The complete and minimal set of residuals generated by the vehicle estimation graph is summarized in Table 3.4. These residuals are selected from a larger set generated by the estimators, based on their sensitivity to the considered faults. Each row in the table presents a residual definition, its corresponding estimation equation, and the local estimator E_i responsible for generating the estimate. Superscripts indicate which estimator produced the estimate, allowing the spatial structure of residual generation to be tracked across the estimation graph. This detailed residual catalog serves as a foundation for constructing fault signatures and designing reliable fault detection and isolation (FDI) mechanisms.

3.4 Summary

This chapter introduced the estimation graph as a novel representation of cascaded estimators in complex systems, where directed edges capture the flow of estimated states. By defining estimation paths, previously hidden redundancies were systematically extracted and quantified. These paths enabled the construction of structured residuals, providing multiple, functionally independent views of the same states. A fault–path dependency matrix and a residual–path incidence matrix were constructed, from which a fault–residual signature matrix was derived using Boolean logic. The structural isolability of fault sources was then assessed. The proposed graph-based formulation offers a principled and scalable framework for root cause analysis, laying the structural groundwork for distributed fault diagnosis in Chapter 5.

Table 3.4: Summary of residual definitions. Superscript (E_i) indicates that the variable is estimated by local estimator E_i .

Residuals	Residual Equations
longitudinal acceleration	$\begin{aligned} \tilde{a}_{x,1} &= a_{x,m} - \hat{a}_x^{(1)}, \\ \hat{a}_x^{(1)} &= \frac{1}{m} \left[\left(\hat{F}_{x,fl}^{(E_4)} + \hat{F}_{x,fr}^{(E_4)} \right) \cos \delta + \left(\hat{F}_{x,rl}^{(E_4)} + \hat{F}_{x,rr}^{(E_4)} \right) - \hat{F}_{y,f}^{(E_5)} \sin \delta \right] \\ \tilde{a}_{x,2} &= a_{x,m} - \hat{a}_x^{(2)}, \quad \hat{a}_x^{(2)} = \dot{\hat{V}}_x^{(E_7)} - \dot{\psi}_m \hat{V}_y^{(E_8)} \end{aligned}$
Lateral acceleration	$\tilde{a}_{y,1} = a_{y,m} - \hat{a}_y^{(E_5)}, \quad \tilde{a}_{y,2} = a_y^{(m)} - \hat{a}_y^{(E_8)}$
Vertical acceleration	$\tilde{a}_z = a_{z,m} - \hat{a}_z, \quad \hat{a}_z = (a_{x,m} + g_x) \tan \hat{\theta}_r^{(E_9)} - (a_{y,m} + g_y) \tan \hat{\phi}_r^{(E_{10})} - g_z$
Roll rate	$\tilde{\phi}_1 = \dot{\phi}_m - \hat{\phi}^{(E_2)}, \quad \tilde{\phi}_2 = \dot{\phi}_m - \hat{\phi}^{(E_{10})}$
Yaw rate	$\tilde{\psi}_1 = \dot{\psi}_m - R_e(\omega_{rl,m} - \omega_{rr,m})/b_w, \quad \tilde{\psi}_2 = \dot{\psi}_m - \hat{\psi}^{(E_5)}, \quad \tilde{\psi}_3 = \dot{\psi}_m - \hat{\psi}^{(E_8)}$
Suspension Height	$\widetilde{\Delta z}_{ij} = \Delta z_{ij,m} - \widehat{\Delta z}_{ij}, \quad \widehat{\Delta z}_{ij} \text{ is computed analytically in Section A.3.6.}$
Wheel Speed	$\tilde{\omega}_{ij} = \omega_{ij,m} - \hat{\omega}_{ij}^{(E_7)}$
Vehicle Velocity	$\tilde{V}_x = \hat{V}_x^{(E_7)} - \hat{V}_x^{(E_3)}, \quad \tilde{V}_y = \hat{V}_y^{(E_8)} - \hat{V}_y^{(E_5)}$
Front Axle Forces	$\tilde{F}_{x,f} = \hat{F}_{x,f}^{(E_4)} - \hat{F}_{x,f}^{(E_6)}, \quad \tilde{F}_{y,f} = \hat{F}_{y,f}^{(E_5)} - \hat{F}_{y,f}^{(E_6)}$
Road Angles	$\tilde{\theta}_r = \hat{\theta}_r^{(E_9)}, \quad \tilde{\phi}_r = \hat{\phi}_r^{(E_{10})}$

Chapter 4

Estimation Uncertainty Quantification

In this chapter, different methods are developed for estimation uncertainty quantification in a complex interconnected dynamical systems.

4.1 Problem Formulation

Consider the networked dynamical system composed of N interconnected subsystems. Each subsystem $i \in \{1, \dots, N\}$ evolves according to general dynamics influenced by local and neighboring states, local control inputs, and potential fault signals. The continuous-time dynamics and measurement model for subsystem i are given by:

$$\begin{cases} \dot{\mathbf{x}}^i(t) = \mathbf{g}_i(\mathbf{x}^i(t), \mathbf{x}^j(t), \mathbf{u}^i(t), \mathbf{f}^i(t); \boldsymbol{\theta}^i) \\ \mathbf{y}^i(t) = \mathbf{h}_i(\mathbf{x}^i(t), \mathbf{x}^j(t), \mathbf{u}^i(t), \mathbf{f}^i(t); \boldsymbol{\theta}^i) \end{cases} \quad (4.1)$$

where $\mathbf{x}^i \in \mathbb{R}^{n_i}$ is the local state vector of subsystem i ; $\mathbf{x}^j$ represents the states of other interacting subsystems $j \in \mathcal{N}_i$, with $\mathcal{N}_i$ denoting the set of in-neighbors of subsystem i —that is, the subsystems whose states $\mathbf{x}_j$ directly influence the dynamics or measurements of i ; $\mathbf{u}^i \in \mathbb{R}^{p_i}$ is the local control input; $\mathbf{f}^i \in \mathbb{R}^{q_i}$ denotes local faults, unknown external disturbances, and parametric uncertainties; $\boldsymbol{\theta}^i$ denotes the local parameters, and $\mathbf{g}_i(\cdot)$, $\mathbf{h}_i(\cdot)$ are functions defining the dynamics and output equations, respectively. This formulation captures both intra- and inter-subsystem dependencies, along with the influence of faults and disturbances in a general setting.

The state estimation graph corresponding to the system described in (4.1) is illustrated in Figure 4.1, where each node represents a local estimator associated with a subsystem.

$\mathcal{N}^{in}(i)$ denotes the set of in-neighbors of node i . For notational simplicity, the in-neighbors of any node are henceforth denoted by corresponding uppercase letters; for instance, $\mathcal{I}$ and $\mathcal{J}$ represent the sets of in-neighbors for nodes i and j , respectively. Note that a self-loop at a node i makes i an in-neighbor of itself. The main problem is to understand how a fault source $\mathbf{f}^i$ at node i propagates through the estimation graph and affects the estimation accuracy at other nodes.

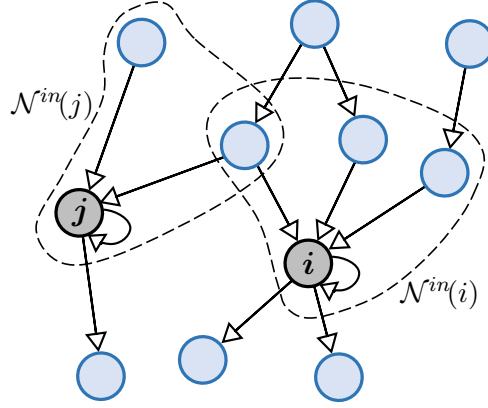


Figure 4.1: An estimation graph. The in-neighbors of arbitrary nodes i and j are represented by $\mathcal{I} = \mathcal{N}^{in}(i)$ and $\mathcal{J} = \mathcal{N}^{in}(j)$, respectively. Self-loops are omitted for clarity.

Suppose the following general local estimator is developed at an arbitrary node i

$$\hat{\mathbf{x}}^i(t) = f(\hat{\mathbf{x}}^{\mathcal{I}}(t), \mathbf{u}^i(t), \mathbf{y}^i(t); \boldsymbol{\theta}^i) \quad (4.2)$$

where $\mathbf{x}^i$ is the estimated states at node i , and $\mathbf{x}^{\mathcal{I}}$ denotes the estimated states of the in-neighbor nodes that share information with node i . A key question is how any faults (process fault $\mathbf{f}_d$, observation fault $\mathbf{f}_b$) and model errors with given distributions impact the estimation error in an arbitrary node i and its downstream nodes. Address this requires quantifying the propagation of these uncertainty sources along the estimation paths and evaluating the sensitivity of each path to different types of uncertainty. In the following, various UQ methods are developed for different estimators.

4.2 Analytical Uncertainty Quantification in Kalman-Based Estimators

Suppose a linear discrete-time subsystem in node i as

$$\begin{aligned}\mathbf{x}_k^i &= \sum_{j \in \mathcal{I}} \mathbf{F}_{k-1}^{ij} \mathbf{x}_{k-1}^j + \mathbf{G}_{k-1}^i \mathbf{u}_{k-1}^i + \Upsilon_{k-1}^i \mathbf{f}_d^i + \boldsymbol{\zeta}_{k-1}^i \\ \mathbf{y}_k^i &= \sum_{j \in \mathcal{I}} \mathbf{H}_k^{ij} \mathbf{x}_k^j + \Lambda_k^i \mathbf{f}_b^i + \boldsymbol{\eta}_k^i\end{aligned}\tag{4.3}$$

where $\mathbf{x}^i$ is a sub-vector of system's full state vector $\mathbf{x}$, $\mathbf{u}^i$ denotes the input sub-vector, and $\mathbf{y}^i$ represents the output sub-vector; the transition matrix $\mathbf{F}^{ij}$ describes how $\mathbf{x}^i$ is dependent on its in-neighbor states $\mathbf{x}^j$, and similarly, $\mathbf{H}^{ij}$ describes how the measurement sub-vector $\mathbf{y}^i$ is related to $\mathbf{x}^j$; the noise sequences $\boldsymbol{\zeta}_k^i$ and $\boldsymbol{\eta}_k^i$ are zero-mean, white Gaussian, mutually uncorrelated with covariance matrices $\mathbf{Q}_k^i$ and $\mathbf{R}_k^i$, respectively; $\mathbf{f}_d^i$ denotes process faults (e.g., additive actuator faults, external disturbance), and $\mathbf{f}_b^i$ denotes the observation fault (e.g., sensor bias, excessive sensor noise) at node i .

Remark: Given the faults and noise processes are considered mutually uncorrelated, the combined estimation uncertainty equals the sum of both. Thus, the noise sequences $\boldsymbol{\zeta}_k$ and $\boldsymbol{\eta}_k$ are ignored in the fault propagation analysis.

The derivation of *a priori* and *a posteriori* estimation uncertainty covariance given fault $\mathbf{f}^i$ occurrence is provided within the Kalman filter framework. Suppose two arbitrary nodes i and j in the estimation graph shown in Figure 4.1. The linear discrete-time state-space model in (4.3) holds for nodes i and j .

The *a priori* estimation error, $\tilde{\mathbf{x}}_k^- := \mathbf{x}_k - \hat{\mathbf{x}}_k^-$, of nodes i and j at time step k is given by

$$\begin{cases} \tilde{\mathbf{x}}_k^{i-} = \sum_{u \in \mathcal{I}} \mathbf{F}_{k-1}^{iu} \tilde{\mathbf{x}}_{k-1}^{u+} + \Upsilon_{k-1}^i \mathbf{f}_d^i \\ \tilde{\mathbf{x}}_k^{j-} = \sum_{v \in \mathcal{J}} \mathbf{F}_{k-1}^{jv} \tilde{\mathbf{x}}_{k-1}^{v+} + \Upsilon_{k-1}^j \mathbf{f}_d^j \end{cases}\tag{4.4}$$

The state update at time step k is $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$ where $\mathbf{K}_k$ is the Kalman gain. Thus, the *a posteriori* estimation error, $\tilde{\mathbf{x}}_k^+ := \mathbf{x}_k - \hat{\mathbf{x}}_k^+$, of nodes i and j after measurement update at time k is

$$\begin{cases} \tilde{\mathbf{x}}_k^{i+} = \tilde{\mathbf{x}}_k^{i-} - \sum_{u \in \mathcal{I}} \mathbf{K}_k^i \mathbf{H}_k^{iu} \tilde{\mathbf{x}}_k^{u-} - \mathbf{K}_k^i \Lambda_k^i \mathbf{f}_b^i \\ \tilde{\mathbf{x}}_k^{j+} = \tilde{\mathbf{x}}_k^{j-} - \sum_{v \in \mathcal{J}} \mathbf{K}_k^j \mathbf{H}_k^{jv} \tilde{\mathbf{x}}_k^{v-} - \mathbf{K}_k^j \Lambda_k^j \mathbf{f}_b^j \end{cases}\tag{4.5}$$

The estimation error cross-covariance matrices between node i and j , $\mathbf{P}_k^{ij-}$ and $\mathbf{P}_k^{ij+}$, are defined as:

$$\mathbf{P}_k^{ij-} \triangleq \mathbb{E} \left[\tilde{\mathbf{x}}_k^{i-} \tilde{\mathbf{x}}_k^{j- \top} \right], \quad \mathbf{P}_k^{ij+} \triangleq \mathbb{E} \left[\tilde{\mathbf{x}}_k^{i+} \tilde{\mathbf{x}}_k^{j+ \top} \right] \quad (4.6)$$

If $i = j$, then $\mathbf{P}_k^{ii}$ is called a auto-covariance matrix.

The *a priori* error cross-covariance $\mathbf{P}_k^{ij-}$ is obtained by substituting $\tilde{\mathbf{x}}_k^{i-}$ and $\tilde{\mathbf{x}}_k^{j-}$ from (4.4) into (4.6) as

$$\begin{aligned} \mathbf{P}_k^{ij-} &= \mathbb{E} \left[\tilde{\mathbf{x}}_k^{i-} \tilde{\mathbf{x}}_k^{j- \top} \right] \\ &= \mathbb{E} \left[\left(\sum_{u \in \mathcal{I}} \mathbf{F}_{k-1}^{iu} \tilde{\mathbf{x}}_{k-1}^{u+} + \Upsilon_{k-1}^i \mathbf{f}_d^i \right) \cdot \left(\sum_{v \in \mathcal{J}} \mathbf{F}_{k-1}^{jv} \tilde{\mathbf{x}}_{k-1}^{v+} + \Upsilon_{k-1}^j \mathbf{f}_d^j \right)^\top \right] \\ &= \sum_{u \in \mathcal{I}, v \in \mathcal{J}} \mathbf{F}_{k-1}^{iu} \mathbb{E} \left[\tilde{\mathbf{x}}_{k-1}^{u+} \tilde{\mathbf{x}}_{k-1}^{v+ \top} \right] \mathbf{F}_{k-1}^{jv \top} + \sum_{u \in \mathcal{I}} \mathbf{F}_{k-1}^{iu} \mathbb{E} \left[\tilde{\mathbf{x}}_{k-1}^{u+} \mathbf{f}_d^{j \top} \right] \Upsilon_{k-1}^{j \top} + \dots \\ &\quad \sum_{v \in \mathcal{J}} \Upsilon_{k-1}^i \mathbb{E} \left[\mathbf{f}_d^i \tilde{\mathbf{x}}_{k-1}^{v+ \top} \right] \mathbf{F}_{k-1}^{jv \top} + \Upsilon_{k-1}^i \mathbb{E} \left[\mathbf{f}_d^i \mathbf{f}_d^{j \top} \right] \Upsilon_{k-1}^{j \top} \end{aligned} \quad (4.7)$$

If $\mathbf{f}_d^j$ corresponds to an excessive white noise fault, the cross-covariance term vanishes, i.e., $\mathbb{E}[\tilde{\mathbf{x}}_k^{i+} \mathbf{f}_d^{j \top}] = \mathbf{0}$. In the case of a bias fault, the factorization $\mathbb{E}[\tilde{\mathbf{x}}_k^{i+} \mathbf{f}_d^{j \top}] = \mathbf{L}_k^{ij} \mathbf{D}^j$ can be applied. By substituting this expression into (4.7) and using the definitions in equations (A.24) and (4.6), the cross-covariance propagation is obtained as:

$$\begin{aligned} \mathbf{P}_k^{ij-} &= \sum_{u \in \mathcal{I}, v \in \mathcal{J}} \mathbf{F}_{k-1}^{iu} \mathbf{P}_{k-1}^{uv+} \mathbf{F}_{k-1}^{jv \top} + \Upsilon_{k-1}^i \mathbf{D}^{ij} \Upsilon_{k-1}^{j \top} + \dots \\ &\quad \sum_{u \in \mathcal{I}} \mathbf{F}_{k-1}^{iu} \mathbf{L}_{k-1}^{uj} \mathbf{D}^j \Upsilon_{k-1}^{j \top} + \Upsilon_{k-1}^i \mathbf{D}^i \sum_{v \in \mathcal{J}} \mathbf{L}_{k-1}^{vi \top} \mathbf{F}_{k-1}^{jv \top} \end{aligned} \quad (4.8)$$

where the recursive equation for influence matrix $\mathbf{L}_k^{ij}$ is derived as follows:

$$\begin{aligned} \mathbb{E} \left[\tilde{\mathbf{x}}_k^{i+} \mathbf{f}_d^{j \top} \right] &= \mathbb{E} \left[\left(\tilde{\mathbf{x}}_k^{i-} - \sum_{u \in \mathcal{I}} \mathbf{K}_k^i \mathbf{H}_k^{iu} \tilde{\mathbf{x}}_k^{u-} - \mathbf{K}_k^i \Lambda_k^i \mathbf{f}_b^i \right) \cdot \mathbf{f}_d^{j \top} \right] \\ &= \mathbb{E} \left[\tilde{\mathbf{x}}_k^{i-} \mathbf{f}_d^{j \top} \right] - \sum_{u \in \mathcal{I}} \mathbf{K}_k^i \mathbf{H}_k^{iu} \mathbb{E} \left[\tilde{\mathbf{x}}_k^{u-} \mathbf{f}_d^{j \top} \right] - \mathbf{K}_k^i \Lambda_k^i \mathbb{E} \left[\mathbf{f}_b^i \mathbf{f}_d^{j \top} \right] \end{aligned} \quad (4.9)$$

The last term vanishes since the observation fault at node i is assumed to be uncorrelated with the process fault at node j , i.e., $\mathbb{E} \left[\mathbf{f}_b^i \mathbf{f}_d^{j \top} \right] = \mathbf{0}$. Substituting $\tilde{\mathbf{x}}_k^{i-}$ and $\tilde{\mathbf{x}}_k^{u-}$ from (4.4)

yields:

$$\begin{aligned}
\mathbb{E}[\tilde{\mathbf{x}}_k^{i+} \mathbf{f}_d^{j\top}] &= \mathbb{E} \left[\left(\sum_{u \in \mathcal{I}} \mathbf{F}_{k-1}^{iu} \tilde{\mathbf{x}}_{k-1}^{u+} + \Upsilon_{k-1}^i \mathbf{f}_d^i \right) \mathbf{f}_d^{j\top} \right] - \dots \\
&\quad \sum_{u \in \mathcal{I}} \mathbf{K}_k^i \mathbf{H}_k^{iu} \mathbb{E} \left[\left(\sum_{v \in \mathcal{U}} \mathbf{F}_{k-1}^{uv} \tilde{\mathbf{x}}_{k-1}^{v+} + \Upsilon_{k-1}^u \mathbf{f}_d^u \right) \mathbf{f}_d^{j\top} \right] \\
&= \sum_{u \in \mathcal{I}} \mathbf{F}_{k-1}^{iu} \mathbb{E} \left[\tilde{\mathbf{x}}_{k-1}^{u+} \mathbf{f}_d^{j\top} \right] + \Upsilon_{k-1}^i \mathbb{E} \left[\mathbf{f}_d^i \mathbf{f}_d^{j\top} \right] - \dots \\
&\quad \sum_{u \in \mathcal{I}} \mathbf{K}_k^i \mathbf{H}_k^{iu} \sum_{v \in \mathcal{U}} \mathbf{F}_{k-1}^{uv} \mathbb{E} \left[\tilde{\mathbf{x}}_{k-1}^{v+} \mathbf{f}_d^{j\top} \right] - \sum_{u \in \mathcal{I}} \mathbf{K}_k^i \mathbf{H}_k^{iu} \Upsilon_{k-1}^u \mathbb{E} \left[\mathbf{f}_d^u \mathbf{f}_d^{j\top} \right]
\end{aligned} \tag{4.10}$$

where, by definition, $\mathbb{E}[\mathbf{f}_d^i \mathbf{f}_d^{j\top}] = \mathbf{D}^{ij}$. A zero-one shape matrix Ψ^{ij} is introduced such that $\mathbf{D}^{ij} = \Psi^{ij} \mathbf{D}^j$, indicating the presence of common process faults between nodes i and j ¹. Decomposing $\mathbb{E}[\tilde{\mathbf{x}}_k^{i+} \mathbf{f}_d^{j\top}]$ into $\mathbf{L}_k^{ij} \mathbf{D}^j$ and noting $\mathbf{D}^j > 0$ leads to:

$$\mathbf{L}_{k+1}^{ij} = \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbf{L}_k^{uj} - \mathbf{K}_k^i \mathbf{H}_k^{iu} \left[\sum_{v \in \mathcal{U}} \mathbf{F}_k^{uv} \mathbf{L}_k^{vj} - \Upsilon_k^u \Psi^{uj} \right] + \Upsilon_k^i \Psi^{ij} \tag{4.11}$$

Similarly, the *a posteriori* error cross-covariance $\mathbf{P}_k^{ij+}$ is obtained by substituting $\tilde{\mathbf{x}}_k^{i+}$ and $\tilde{\mathbf{x}}_k^{j+}$ from (4.5) into the definition of error covariance in (4.6), yielding:

$$\begin{aligned}
\mathbf{P}_k^{ij+} &= \mathbb{E} \left[\tilde{\mathbf{x}}_k^{i+} \tilde{\mathbf{x}}_k^{j+ \top} \right] \\
&= \mathbb{E} \left[\left(\tilde{\mathbf{x}}_k^{i-} - \sum_{u \in \mathcal{I}} \mathbf{K}_k^i \mathbf{H}_k^{iu} \tilde{\mathbf{x}}_k^{u-} - \mathbf{K}_k^i \Lambda_k^i \mathbf{f}_b^i \right) \cdot \left(\tilde{\mathbf{x}}_k^{j-} - \sum_{v \in \mathcal{J}} \mathbf{K}_k^j \mathbf{H}_k^{jv} \tilde{\mathbf{x}}_k^{v-} - \mathbf{K}_k^j \Lambda_k^j \mathbf{f}_b^j \right)^\top \right] \\
&= \mathbb{E} \left[\tilde{\mathbf{x}}_k^{i-} \tilde{\mathbf{x}}_k^{j- \top} \right] - \sum_{v \in \mathcal{J}} \mathbb{E} \left[\tilde{\mathbf{x}}_k^{i-} \tilde{\mathbf{x}}_k^{v- \top} \right] \mathbf{H}_k^{jv \top} \mathbf{K}_k^{j \top} - \mathbb{E} \left[\tilde{\mathbf{x}}_k^{i-} \mathbf{f}_b^{j \top} \right] \Lambda_k^{j \top} \mathbf{K}_k^{j \top} - \dots \\
&\quad \sum_{u \in \mathcal{I}} \mathbf{K}_k^i \mathbf{H}_k^{iu} \mathbb{E} \left[\tilde{\mathbf{x}}_k^{u-} \tilde{\mathbf{x}}_k^{j- \top} \right] + \sum_{u \in \mathcal{I}, v \in \mathcal{J}} \mathbf{K}_k^i \mathbf{H}_k^{iu} \mathbb{E} \left[\tilde{\mathbf{x}}_k^{u-} \tilde{\mathbf{x}}_k^{v- \top} \right] \mathbf{H}_k^{jv \top} \mathbf{K}_k^{j \top} + \dots \\
&\quad \sum_{u \in \mathcal{I}} \mathbf{K}_k^i \mathbf{H}_k^{iu} \mathbb{E} \left[\tilde{\mathbf{x}}_k^{u-} \mathbf{f}_b^{j \top} \right] \Lambda_k^{j \top} \mathbf{K}_k^{j \top} + \mathbf{K}_k^i \Lambda_k^i \sum_{v \in \mathcal{J}} \mathbb{E} \left[\mathbf{f}_b^i \tilde{\mathbf{x}}_k^{v- \top} \right] \mathbf{H}_k^{jv \top} \mathbf{K}_k^{j \top} - \dots \\
&\quad \mathbf{K}_k^i \Lambda_k^i \mathbb{E} \left[\mathbf{f}_b^i \tilde{\mathbf{x}}_k^{j- \top} \right] + \mathbf{K}_k^i \Lambda_k^i \left[\mathbf{f}_b^i \mathbf{f}_b^{j \top} \right] \Lambda_k^{j \top} \mathbf{K}_k^{j \top}
\end{aligned} \tag{4.12}$$

¹For example, if $i = j$, then $\Psi^{ij} = \mathbf{I}$, and if i and j does not share the common process fault, then $\Psi^{ij} = \mathbf{0}$.

If $\mathbf{f}_b^j$ corresponds to excessive white noise, then $\mathbb{E}[\tilde{\mathbf{x}}_k^{i-} \mathbf{f}_b^{j\top}] = \mathbf{0}$. In the case of a bias fault, the cross-covariance can be factorized as $\mathbb{E}[\tilde{\mathbf{x}}_k^{i-} \mathbf{f}_b^{j\top}] = \mathbf{M}_k^{ij} \mathbf{B}^j$. By definition, the covariance between observation faults at nodes i and j is given by $\mathbb{E}[\mathbf{f}_b^i \mathbf{f}_b^{j\top}] = \mathbf{B}^{ij}$, and the auto-covariance at node i is $\mathbb{E}[\mathbf{f}_b^i \mathbf{f}_b^{i\top}] = \mathbf{B}^i$. Substituting these expressions into the *a posteriori* covariance formulation yields:

$$\begin{aligned} \mathbf{P}_k^{ij+} &= \mathbf{P}_k^{ij-} - \mathbf{K}_k^i \sum_{u \in \mathcal{I}} \mathbf{H}_k^{iu} \mathbf{P}_k^{uj-} - \sum_{v \in \mathcal{J}} \mathbf{P}_k^{iv-} \mathbf{H}_k^{jv\top} \mathbf{K}_k^{j\top} + \dots \\ &\quad \mathbf{K}_k^i \sum_{u \in \mathcal{I}, v \in \mathcal{J}} \mathbf{H}_k^{iu} \mathbf{P}_k^{uv-} \mathbf{H}_k^{jv\top} \mathbf{K}_k^{j\top} + \mathbf{K}_k^i \mathbf{\Lambda}_k^i \mathbf{B}^{ij} \mathbf{\Lambda}_k^{j\top} \mathbf{K}_k^{j\top} - \dots \\ &\quad \left(\mathbf{M}_k^{ij} - \mathbf{K}_k^i \sum_{u \in \mathcal{I}} \mathbf{H}_k^{iu} \mathbf{M}_k^{uj} \right) \mathbf{B}^j \mathbf{\Lambda}_k^{j\top} \mathbf{K}_k^{j\top} - \mathbf{K}_k^i \mathbf{\Lambda}_k^i \mathbf{B}^i \left(\mathbf{M}_k^{ji} - \mathbf{K}_k^j \sum_{v \in \mathcal{J}} \mathbf{H}_k^{jv} \mathbf{M}_k^{vi} \right)^\top \end{aligned} \quad (4.13)$$

where the recursive equation for influence matrix $\mathbf{M}_k^{ij}$ can be obtained as follows:

$$\begin{aligned} \mathbb{E}[\tilde{\mathbf{x}}_{k+1}^{i-} \mathbf{f}_b^{j\top}] &= \mathbb{E} \left[\left(\sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \tilde{\mathbf{x}}_k^{u+} + \mathbf{r}_k^i \mathbf{f}_d^i \right) \mathbf{f}_b^{j\top} \right] \\ &= \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbb{E}[\tilde{\mathbf{x}}_k^{u+} \mathbf{f}_b^{j\top}] + \mathbf{r}_k^i \mathbb{E}[\mathbf{f}_d^i \mathbf{f}_b^{j\top}] \end{aligned} \quad (4.14)$$

The last term vanishes since the process fault at node i is assumed to be uncorrelated with the observation fault at node j , i.e., $\mathbb{E}[\mathbf{f}_d^i \mathbf{f}_b^{j\top}] = \mathbf{0}$. Substituting $\tilde{\mathbf{x}}_k^{u+}$ from (4.5) yields:

$$\begin{aligned} \mathbb{E}[\tilde{\mathbf{x}}_{k+1}^{i-} \mathbf{f}_b^{j\top}] &= \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbb{E}[\tilde{\mathbf{x}}_k^{u-} \mathbf{f}_b^{j\top}] - \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbf{K}_k^u \sum_{v \in \mathcal{U}} \mathbf{H}_k^{uv} \mathbb{E}[\tilde{\mathbf{x}}_k^{v-} \mathbf{f}_b^{j\top}] - \dots \\ &\quad \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbf{K}_k^u \mathbf{\Lambda}_k^u \mathbb{E}[\mathbf{f}_b^u \mathbf{f}_b^{j\top}] \end{aligned} \quad (4.15)$$

where, by definition, $\mathbb{E}[\mathbf{f}_b^i \mathbf{f}_b^{j\top}] = \mathbf{B}^{ij}$. A zero-one shape matrix Φ^{ij} is introduced such that $\mathbf{B}^{ij} = \Phi^{ij} \mathbf{B}^j$, indicating common output faults at nodes i and j ². By decomposing $\mathbb{E}[\tilde{\mathbf{x}}_k^{i-} \mathbf{f}_b^{j\top}]$ as $\mathbf{M}_k^{ij} \mathbf{B}^j$, and noting that $\mathbf{B}^j > 0$, it follows:

$$\mathbf{M}_{k+1}^{ij} = \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \left[\mathbf{M}_k^{uj} - \mathbf{K}_k^u \sum_{v \in \mathcal{U}} \mathbf{H}_k^{uv} \mathbf{M}_k^{vj} - \mathbf{K}_k^u \mathbf{\Lambda}_k^u \Phi^{uj} \right] \quad (4.16)$$

²For instance, $\Phi^{ij} = \mathbf{I}$ if $i = j$, and $\Phi^{ij} = \mathbf{0}$ if i and j share no common output fault.

Table 4.1: Recursive quantification of state estimation uncertainty

1. Recursive update of the <i>a priori</i> estimation error covariance matrix	
$\mathbf{P}_k^{ij-} = \sum_{u \in \mathcal{I}, v \in \mathcal{J}} \mathbf{F}_{k-1}^{iu} \mathbf{P}_{k-1}^{uv+} \mathbf{F}_{k-1}^{jv\top} + \mathbf{\Upsilon}_{k-1}^i \mathbf{D}^{ij} \mathbf{\Upsilon}_{k-1}^{j\top} + \dots$ $\sum_{u \in \mathcal{I}} \mathbf{F}_{k-1}^{iu} \mathbf{L}_{k-1}^{uj} \mathbf{D}^j \mathbf{\Upsilon}_{k-1}^{j\top} + \mathbf{\Upsilon}_{k-1}^i \mathbf{D}^i \sum_{v \in \mathcal{J}} \mathbf{L}_{k-1}^{vi\top} \mathbf{F}_{k-1}^{jv\top}$ (4.17)	
2. Recursive update of the <i>a posteriori</i> estimation error covariance matrix	
$\mathbf{P}_k^{ij+} = \mathbf{P}_k^{ij-} - \mathbf{K}_k^i \sum_{u \in \mathcal{I}} \mathbf{H}_k^{iu} \mathbf{P}_k^{uj-} - \sum_{v \in \mathcal{J}} \mathbf{P}_k^{iv-} \mathbf{H}_k^{jv\top} \mathbf{K}_k^{j\top} + \dots$ $\mathbf{K}_k^i \sum_{u \in \mathcal{I}, v \in \mathcal{J}} \mathbf{H}_k^{iu} \mathbf{P}_k^{uv-} \mathbf{H}_k^{jv\top} \mathbf{K}_k^{j\top} + \mathbf{K}_k^i \mathbf{\Lambda}_k^i \mathbf{B}^{ij} \mathbf{\Lambda}_k^{j\top} \mathbf{K}_k^{j\top} - \dots$ $\left(\mathbf{M}_k^{ij} - \mathbf{K}_k^i \sum_{u \in \mathcal{I}} \mathbf{H}_k^{iu} \mathbf{M}_k^{uj} \right) \mathbf{B}^j \mathbf{\Lambda}_k^{j\top} \mathbf{K}_k^{j\top} - \mathbf{K}_k^i \mathbf{\Lambda}_k^i \mathbf{B}^i \left(\mathbf{M}_k^{ji} - \mathbf{K}_k^j \sum_{v \in \mathcal{J}} \mathbf{H}_k^{jv} \mathbf{M}_k^{vi} \right)^\top$ (4.18)	
3. Recursive update of the influence matrix $\mathbf{L}$	
$\mathbf{L}_{k+1}^{ij} = \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbf{L}_k^{uj} - \mathbf{K}_k^i \mathbf{H}_k^{iu} \left[\sum_{v \in \mathcal{U}} \mathbf{F}_k^{uv} \mathbf{L}_k^{vj} - \mathbf{\Upsilon}_k^u \mathbf{\Psi}^{uj} \right] + \mathbf{\Upsilon}_k^i \mathbf{\Psi}^{ij}$ (4.19)	
4. Recursive update of the influence matrix $\mathbf{M}$	
$\mathbf{M}_{k+1}^{ij} = \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \left[\mathbf{M}_k^{uj} - \mathbf{K}_k^u \sum_{v \in \mathcal{U}} \mathbf{H}_k^{uv} \mathbf{M}_k^{vj} - \mathbf{K}_k^u \mathbf{\Lambda}_k^u \mathbf{\Phi}^{uj} \right]$ (4.20)	

Table 4.1 summarizes the recursive quantification of estimation uncertainty propagation between two arbitrary nodes, given that a fault—either an observation fault $\mathbf{f}_b^i$ or a process fault $\mathbf{f}_d^i$ —with a known distribution occurs at nodes i and j . For bias-type faults, the influence matrices $\mathbf{L}_k^{ij}$ and $\mathbf{M}_k^{ij}$ are updated recursively according to (4.19) and (4.20), respectively. In contrast, for excessive noise faults, the influence matrices are set to $\mathbf{0}$.

Typically, estimated states at a node serve as virtual measurements, $\hat{\mathbf{y}}$, for descendant nodes. In such instances, these estimates are moved to the right-hand side of the output equation in the state-space model (4.3) and treated as states, thereby enabling the use of

uncertainty propagation in (4.17)–(4.20).

4.2.1 Computational Complexity

This subsection evaluates the computational efficiency of the proposed analytical UQ methodology and compares it with traditional sampling-based approaches, such as Monte Carlo (MC) simulations.

For a standard Kalman filter applied to a system with a state vector of size n , the computational complexity of computing the error covariance matrix is $\mathcal{O}(n^3)$; see [119]. In the context of the proposed estimation graph, where $\mathbf{v}$ nodes each represent a state estimator with n states, the computational complexity of computing the auto-covariance for a fully connected node by (4.17)–(4.20) scales as $\mathcal{O}(\mathbf{v}^2 n^3)$. For nodes with fewer connections, the complexity is correspondingly reduced due to the sparsity in cross-covariance computations. Notably, the additional computational cost introduced by real-time UQ is of the same order as that of the Kalman estimators themselves.

In contrast, Monte Carlo method with N samples incur significantly higher computational cost. Specifically, for each of the $\mathbf{v}$ estimators, the Kalman filter prediction and update steps must be executed N times, resulting in a total complexity of $\mathcal{O}(N\mathbf{v}n^2)$. In addition, computing the cross-covariance matrix from these samples introduces an extra $\mathcal{O}(Nn^2)$ operations per node. Thus, the overall computational complexity for obtaining the auto-covariance of a node (assuming connectivity to all other nodes) becomes $\mathcal{O}(N\mathbf{v}n^2)$.

Given that in practical settings $N \in \{10^3, 10^4\}$ and typically $\mathbf{v} \cdot n \ll N$, the proposed analytical approach offers a substantial computational advantage over sampling-based methods in terms of both scalability and efficiency.

The analytical approach developed for Kalman-based estimation graphs provides significantly greater computational efficiency compared to sampling-based methods such as MC. Moreover, the same methodology is applicable to a broader class of linear observers. For example, Appendix B presents a detailed derivation of uncertainty propagation for a general Luenberger observer, and for a specific case of a linear tire force observer, using the same analytical push-forward strategy. In contrast, uncertainty propagation in nonlinear estimators is considerably more complex. Several techniques have been proposed to quantify uncertainty in nonlinear, data-driven, or black-box models, among which sampling-based approaches—particularly MC simulations—have received the most attention in the literature. However, due to their high computational cost, such methods are generally unsuitable for real-time implementation.

One alternative is to use a Taylor series expansion to approximate the nonlinear estimator. Assuming the uncertainties are small relative to the signal values, linearization techniques—such as Taylor expansion or perturbation methods—can be employed to express the estimator as a linear function of the fault source. This enables the application of analytical uncertainty quantification similar to the linear case. However, in practice, this linearization can be difficult to implement reliably. Therefore, there is a need to develop alternative methods that are both computationally efficient and suitable for real-time uncertainty quantification in nonlinear or black-box observers.

4.3 Uncertainty Quantification Using Polynomial Chaos Expansion

Nonlinear and black-box estimators present significant challenges for uncertainty quantification. These estimators often incorporate embedded logic, switching mechanisms, or data-driven components, which make analytical reformulation intractable and render intrusive uncertainty propagation techniques impractical. Furthermore, in many practical cases, faults are introduced into the estimator in a nonlinear or state-dependent manner, further complicating the analysis.

A state estimator can be viewed as a dynamical system that recursively fuses current measurements with model-based predictions from the previous time step to estimate the current state. To propagate uncertainty in dynamical system, several approaches have been developed, differing in terms of their assumptions, computational cost, and compatibility with model accessibility:

1. Sampling-Based Methods

- Monte Carlo Simulation
- Latin Hypercube Sampling

2. Perturbation Methods

- Method of Adjoints
- First-Order Taylor Approximation

3. Generalized Polynomial Chaos Expansion

- Intrusive Methods (e.g., Galerkin Projection)
- Non-Intrusive Methods (e.g., Stochastic Collocation)

Sampling-based methods, while widely applicable, are computationally intensive and thus unsuitable for real-time applications. Perturbation methods are efficient but rely on simplifying assumptions, making them appropriate only when the system is mildly nonlinear, uncertainties are small, and the model is explicitly available. In contrast, generalized polynomial chaos (gPC) provides a powerful framework for uncertainty quantification in both linear and nonlinear systems by representing uncertain parameters using orthogonal polynomial bases, enabling accurate and efficient propagation of uncertainty. The gPC expansion techniques can be broadly categorized into intrusive and non-intrusive methods, based on how they interact with the system model.

Intrusive gPC modifies the system’s governing equations by projecting them onto an orthogonal polynomial basis using the Galerkin method. This approach yields a set of deterministic equations that capture the propagation of uncertainty through the system. While intrusive methods offer high accuracy and computational efficiency, their application is often limited to systems with accessible and tractable analytical models. In practice, they are best suited to well-defined physical systems with complete modeling transparency.

By contrast, non-intrusive gPC methods, such as stochastic collocation, treat the system as a black box. These approaches avoid altering the system equations and instead rely on sampling the input uncertainties at selected collocation points, using the resulting simulations to construct a surrogate model through polynomial interpolation or regression. Non-intrusive approaches are especially attractive for systems involving black-box estimators, strong nonlinearities, or cascaded structures—scenarios where analytical access to the governing equations is limited or unavailable. For instance, in state estimation architectures—common in vehicle systems or interconnected dynamical networks—the estimator may include nonlinear observers, neural networks, or heuristic modules that do not lend themselves to Galerkin projection. In such cases, stochastic collocation offers a flexible and effective means for uncertainty quantification by leveraging simulation outputs rather than analytic derivations.

4.3.1 Mathematical Foundations of Polynomial Chaos Expansion

This section presents the mathematical foundations necessary for understanding gPC. An introduction to Hilbert spaces, inner products, and norms is provided, as these concepts form the basis for constructing orthogonal polynomial bases used in uncertainty quantification.

Definition 2 (Vector Space of Square Integrable Functions). *Let X be a random vector with probability density function $p(x)$. The vector space of square-integrable functions, denoted by $L^2(\Omega, \mathcal{F}, P)$, is defined as the set of all functions $f : \mathbb{R}^d \rightarrow \mathbb{R}$ such that:*

$$\mathbb{E}_p [f^2(X)] = \int_{\Omega} f^2(x)p(x) dx < \infty$$

This means that for every function f in $L^2(\Omega, \mathcal{F}, P)$, the expected value of its square under the probability distribution $p(x)$ is finite. The space L^2 is a vector space because, for any two functions $f, g \in L^2$ and scalars $\mu, \nu \in \mathbb{R}$, the linear combination $\mu f + \nu g$ is also in L^2 .

Definition 3 (Hilbert Spaces). *A Hilbert space is a complete inner product space, which generalizes the notion of Euclidean space to an infinite-dimensional setting. Formally, a Hilbert space $\mathcal{H}$ is a vector space equipped with an inner product $\langle \cdot, \cdot \rangle$ that induces a norm $\| \cdot \|$, where the space is complete with respect to the norm.*

Definition 4 (Inner Product). *The inner product on a Hilbert space $\mathcal{H}$ is a bilinear map $\langle \cdot, \cdot \rangle : \mathcal{H} \times \mathcal{H} \rightarrow \mathbb{R}$, satisfying the following properties for all $f, g, h \in \mathcal{H}$ and $\alpha \in \mathbb{R}$:*

- *Linearity:* $\langle \alpha f, g \rangle = \alpha \langle f, g \rangle$
- *Symmetry:* $\langle f, g \rangle = \langle g, f \rangle$
- *Positive definiteness:* $\langle f, f \rangle \geq 0$, and $\langle f, f \rangle = 0$ if and only if $f = 0$

Definition 5 (Norm). *The norm induced by the inner product is given by $\|f\| = \sqrt{\langle f, f \rangle}$. This norm measures the length of vectors in the Hilbert space.*

Properties of Hilbert Spaces: Hilbert spaces possess several key properties that make them useful in the context of uncertainty quantification and PCE. These include:

- **Orthogonality:** Two vectors f and g in $\mathcal{H}$ are orthogonal if $\langle f, g \rangle = 0$.
- **Orthonormal basis:** A set of vectors $\{\varphi_i\}$ forms an orthonormal basis of $\mathcal{H}$ if $\langle \varphi_i, \varphi_j \rangle = \delta_{ij}$, where δ_{ij} is the Kronecker delta, and every vector $f \in \mathcal{H}$ can be written as a linear combination of the basis vectors:

$$f = \sum_i c_i \varphi_i, \quad c_i = \langle f, \varphi_i \rangle$$

- **Completeness:** A space is complete if every Cauchy sequence in the space converges to a point in the space. This is a critical property for ensuring that the PCE expansion converges to a valid solution.

The gPC approach for uncertainty quantification relies on expanding a stochastic process or random variable in terms of orthogonal polynomials that form a basis in a Hilbert space of square-integrable functions. Let X be a random variable with a probability density function $p(x)$, and consider the Hilbert space of square-integrable functions $L^2(\Omega, \mathcal{F}, P)$, defined as:

$$L^2(\Omega, \mathcal{F}, P) = \{f : \Omega \rightarrow \mathbb{R} \mid \mathbb{E}[f(X)^2] < \infty\}$$

In this space, the inner product is defined as:

$$\langle f, g \rangle = \int_{\Omega} f(x)g(x)p(x) dx.$$

The norm in the Hilbert space L^2 is given by:

$$\|f\|^2 = \int_{\Omega} f(x)^2 p(x) dx.$$

The series expansion in gPC converges in the L^2 norm, ensuring that the approximation error decreases as more terms are included in the expansion.

Definition 6 (Orthonormal Basis in Hilbert Space). *In a Hilbert space of square-integrable functions $L^2(\Omega, \mathcal{F}, P)$, a set of $\{\varphi_0, \varphi_1, \varphi_2, \dots\}$ is said to form an orthonormal basis if they are orthogonal with respect to the inner product in L^2 , and each basis is normalized. Specifically, the basis satisfy:*

$$\langle \varphi_i, \varphi_j \rangle = \delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Here, $\langle \varphi_i, \varphi_j \rangle = \int_{\Omega} \varphi_i(x)\varphi_j(x)p(x) dx$ is the inner product in $L^2(\Omega, \mathcal{F}, P)$, and $p(x)$ is the probability density function associated with the random variable X .

The key idea of PCE is to expand a random variable $f(X)$ as a series of orthonormal polynomials $\{\varphi_i(X)\}$, where the polynomials are orthogonal with respect to the probability measure $p(x)$:

$$f(X) = \sum_{i=0}^{\infty} c_i \varphi_i(X), \quad c_i = \langle f, \varphi_i \rangle$$

The orthonormal polynomials can be constructed using methods such as Gram-Schmidt orthogonalization [120].

Theorem 2 (Parseval's Identity). *Let $\{\varphi_i\}$ be an orthonormal basis of a Hilbert space $L^2(\Omega, \mathcal{F}, P)$, where $\varphi_i \in L^2(\Omega, \mathcal{F}, P)$ and $f \in L^2(\Omega, \mathcal{F}, P)$. The Parseval's identity states that:*

$$\|f\|^2 = \sum_{i=0}^{\infty} c_i^2$$

where the coefficients c_i are given by:

$$c_i = \langle f, \varphi_i \rangle = \int_{\Omega} f(x) \varphi_i(x) p(x) dx$$

and $\|f\|^2$ is the norm of f in L^2 , i.e.,

$$\|f\|^2 = \int_{\Omega} f^2(x) p(x) dx.$$

This identity states that the total energy (or norm) of the function f is equal to the sum of the squares of its Fourier coefficients with respect to the orthonormal basis $\{\varphi_i\}$.

Polynomial Basis Functions for Random Variables. Different types of polynomials, such as Hermite for Gaussian distributions and Legendre for uniform distributions, are used in gPC expansion; see Table. 4.2. These polynomials are selected based on the distribution of the uncertain parameters.

Definition 7 (Orthonormal Scalar Polynomials). *Let X be a random variable with probability distribution P on $\mathbb{R}$, and consider the Hilbert space $L^2(\mathbb{R}, P)$ of square-integrable functions. A set of polynomials $\{\varphi_0(x), \varphi_1(x), \varphi_2(x), \dots\}$, each expressed as:*

$$\varphi_i(x) = a_{0i} + a_{1i}x + a_{2i}x^2 + \dots + a_{\rho_i i}x^{\rho_i}$$

forms an orthonormal set if:

$$\langle \varphi_i, \varphi_j \rangle = \int_{\mathbb{R}} \varphi_i(x) \varphi_j(x) p(x) dx = \delta_{ij},$$

where δ_{ij} is the Kronecker delta and $p(x)$ is the probability density function of X . The polynomials are orthonormal, meaning $\langle \varphi_i, \varphi_j \rangle = 0$ for $i \neq j$ and $\langle \varphi_i, \varphi_i \rangle = 1$. The coefficients a_{ji} are chosen to ensure orthonormality with respect to the inner product in $L^2(\mathbb{R}, P)$.

The Gram-Schmidt procedure can be used to generate orthonormal polynomials for any given probability distribution $p(x)$ by starting with a set of linearly independent polynomials and applying the process to create polynomials that are orthogonal with respect to the inner product defined by $p(x)$. This method ensures that the resulting polynomials are both orthogonal and normalized for the specific distribution.

Table 4.2: Hermite, Legendre, and Laguerre Orthonormal Polynomials

Polynomial Basis	Hermite	Legendre	Laguerre
$\varphi_0(x)$	1	1	1
$\varphi_1(x)$	x	$\sqrt{3}x$	$1 - x$
$\varphi_2(x)$	$\frac{1}{\sqrt{2}}(x^2 - 1)$	$\frac{\sqrt{5}}{2}(3x^2 - 1)$	$\frac{1}{2}(\frac{1}{2}x^2 - 2x + 1)$
$\varphi_3(x)$	$\frac{1}{\sqrt{6}}(x^3 - 3x)$	$\frac{\sqrt{7}}{2}(5x^3 - 3x)$	$\frac{1}{\sqrt{6}}(-\frac{x^3}{6} + \frac{3x^2}{2} - 3x + 1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\varphi_n(x)$	$\frac{1}{\sqrt{n!}}H_n(x)$	$\sqrt{2n+1}P_n(x)$	$\frac{1}{\sqrt{n!}}L_n(x)$

Figure 4.2 represents orthonormal Hermite, Legendre, and Laguerre polynomial bases of degrees 0 to 3, which are commonly used in generalized Polynomial Chaos (gPC) expansions depending on the distribution of the underlying random variables.

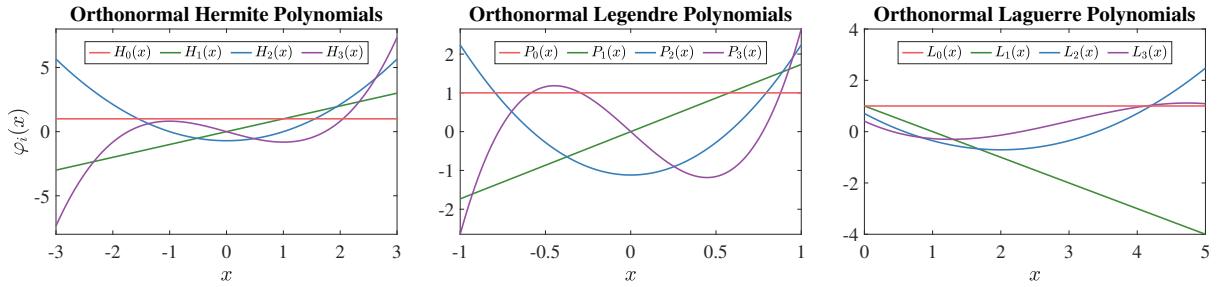


Figure 4.2: Orthonormal Hermite, Legendre, and Laguerre polynomials of degrees 0 to 3.

In the case of multi-dimensional random variables, the construction of orthogonal polynomials can be extended from one-dimensional cases to higher dimensions. Let $X = (X_1, X_2, \dots, X_d)$ be a random vector with a joint probability density function $p(x_1, x_2, \dots, x_d)$. The Hilbert space $L^2(\mathbb{R}^d, P)$ consists of square-integrable functions $f : \mathbb{R}^d \rightarrow \mathbb{R}$ under the measure defined by $p(x_1, x_2, \dots, x_d)$.

A *multivariate polynomial* of total degree ρ is a scalar function of the form:

$$f(\mathbf{x}; a) = \sum_{|\mathbf{i}| \leq \rho} a_{\mathbf{i}} x_1^{i_1} x_2^{i_2} \cdots x_d^{i_d},$$

where $\mathbf{i} = (i_1, i_2, \dots, i_d)$ is a multi-index, $|\mathbf{i}| = i_1 + i_2 + \cdots + i_d$, and $a_{\mathbf{i}}$ are the corresponding coefficients. These polynomials form the basis for constructing orthonormal polynomial functions in higher dimensions.

For each i , let $\varphi_{i,j}(x_i)$ denote a set of orthonormal polynomials in the i -th variable with respect to its marginal density $p(x_i)$. The multi-dimensional orthonormal polynomial can be constructed as a product of one-dimensional polynomials:

$$\varphi_{\mathbf{i}}(\mathbf{x}) = \prod_{k=1}^d \varphi_{i_k}(x_k),$$

where $\mathbf{i} = (i_1, i_2, \dots, i_d)$ is a multi-index, and $\mathbf{x} = (x_1, x_2, \dots, x_d)$ is a point in $\mathbb{R}^d$. These polynomials form an orthonormal basis in the Hilbert space $L^2(\mathbb{R}^d, P)$.

The inner product in this space is defined as:

$$\langle f, g \rangle = \int_{\mathbb{R}^d} f(\mathbf{x})g(\mathbf{x})p(\mathbf{x}) d\mathbf{x}.$$

The polynomials satisfy the orthonormality condition:

$$\langle \varphi_{\mathbf{i}}, \varphi_{\mathbf{j}} \rangle = \delta_{\mathbf{i}, \mathbf{j}},$$

where $\delta_{\mathbf{i}, \mathbf{j}}$ is the Kronecker delta for two multi-indices $\mathbf{i}$ and $\mathbf{j}$.

Example 1 (2D Orthonormal Hermite Polynomials). *Let X_1 and X_2 be independent normal random variables. The corresponding orthogonal polynomials are Hermite polynomials, as shown in Table 4.3. For d -dimensional polynomials of order n , there are $\binom{n+d}{d}$ orthogonal polynomials. For $n = 2$ in 2D, this results in six orthogonal polynomials. The tensor product of $\varphi_i(x_1)$ and $\varphi_j(x_2)$ gives the 2D orthonormal polynomials $\varphi_k(x_1, x_2)$:*

$$\begin{aligned} \varphi_0(x) &= \varphi_{10}(x_1)\varphi_{20}(x_2) = 1 \\ \varphi_1(x) &= \varphi_{11}(x_1)\varphi_{20}(x_2) = x_1 \\ \varphi_2(x) &= \varphi_{12}(x_1)\varphi_{20}(x_2) = \frac{1}{\sqrt{2}}(x_1^2 - 1) \\ \varphi_3(x) &= \varphi_{10}(x_1)\varphi_{21}(x_2) = x_2 \\ \varphi_4(x) &= \varphi_{11}(x_1)\varphi_{21}(x_2) = x_1x_2 \\ \varphi_5(x) &= \varphi_{10}(x_1)\varphi_{22}(x_2) = \frac{1}{\sqrt{2}}(x_2^2 - 1). \end{aligned} \tag{4.21}$$

Table 4.3: Hermite polynomials for two random variables X_1 and X_2 .

Polynomial degree	$\varphi_{1i}(x_1)$	$\varphi_{2i}(x_2)$
$i = 0$	1	1
$i = 1$	x_1	x_2
$i = 2$	$\frac{1}{\sqrt{2!}}(x_1^2 - 1)$	$\frac{1}{\sqrt{2!}}(x_2^2 - 1)$

For more details, the reader is referred to the algorithms developed by Gautschi [121] for deriving orthonormal polynomial bases.

Proposition 1 (Mean and Covariance Propagation). *Let $\mathbf{X} \in \mathbb{R}^n$ be a random vector with joint probability density function $p(\mathbf{x})$, and let $\mathbf{f}(\mathbf{x}) \in L^2(p(\mathbf{x}); \mathbb{R}^d)$ be a square-integrable vector-valued function. The function $\mathbf{f}(\mathbf{x})$ can be approximated using a generalized Polynomial Chaos Expansion (PCE) with respect to a set of multivariate orthonormal polynomials $\varphi_i(\mathbf{x})$ as:*

$$\mathbf{f}(\mathbf{x}) = \sum_{i=0}^{\infty} \mathbf{c}_i \varphi_i(\mathbf{x}),$$

where $\mathbf{c}_i \in \mathbb{R}^d$ are the vector-valued expansion coefficients, and the basis functions satisfy

$$\mathbb{E}_p[\varphi_i(\mathbf{x})\varphi_j(\mathbf{x})] = \delta_{ij}.$$

Then, the first and second moments of $\mathbf{f}(\mathbf{x})$ are given by:

$$\begin{aligned} \mathbb{E}_p[\mathbf{f}(\mathbf{x})] &= \mathbf{c}_0, \\ \mathbb{C}_p[\mathbf{f}(\mathbf{x})] &= \sum_{i=1}^{\infty} \mathbf{c}_i \mathbf{c}_i^\top. \end{aligned} \tag{4.22}$$

Proof. See Appendix C for proof. □

4.3.2 Uncertainty Quantification via Intrusive gPC (Galerkin Method)

A state estimator can be viewed as a dynamical system that recursively fuses current measurements with model-based predictions from the previous time step to estimate the

current state. In the presence of uncertainty due to fault parameters $\mathbf{f} \in \mathbb{R}^d$, the estimator dynamics can be modeled as:

$$\frac{d\hat{\mathbf{x}}(t)}{dt} = \mathbf{g}(\hat{\mathbf{x}}(t), \mathbf{y}(t), t; \mathbf{f}), \quad \hat{\mathbf{x}}(0) = \mathbf{x}_0(\mathbf{f}), \quad (4.23)$$

where $\hat{\mathbf{x}}(t) \in \mathbb{R}^n$ is the estimated state vector, $\mathbf{y}(t) \in \mathbb{R}^m$ is the observation vector, $\mathbf{g}$ is a vector-valued function defining the estimator dynamics, and the initial condition $\mathbf{x}_0(\mathbf{f}) \in L^2(\Omega; \mathbb{R}^n)$ depends on the uncertain parameters.

The goal is to quantify the uncertainty in $\hat{\mathbf{x}}$ over time by computing its statistical moments (e.g., mean and covariance), given the randomness introduced by $\mathbf{f}$. This is achieved by propagating the uncertainty through the estimator dynamics defined above using a suitable uncertainty quantification method.

Assuming that $\hat{\mathbf{x}}(t, \mathbf{f}) \in L^2(\Omega; \mathbb{R}^n)$, it can be approximated using a truncated gPC expansion:

$$\mathbf{x}(t, \mathbf{f}) \approx \sum_{i=0}^P \mathbf{c}_i(t) \varphi_i(\mathbf{f}), \quad (4.24)$$

where:

- $\varphi_i(\mathbf{f})$ are multivariate orthonormal polynomials up to order p , selected according to the distribution of the random vector $\mathbf{f}$,
- P is the total number of retained basis terms,
- $\mathbf{c}_i(t) \in \mathbb{R}^n$ are time-varying deterministic coefficient vectors to be computed.

Galerkin Projection. To derive the evolution equations for the coefficients $\mathbf{c}_i(t)$, the expansion is substituted into the ODE:

$$\frac{d\hat{\mathbf{x}}(t, \mathbf{f})}{dt} = \sum_{i=0}^P \frac{d\mathbf{c}_i(t)}{dt} \varphi_i(\mathbf{f}), \quad (4.25)$$

The nonlinear function $\mathbf{g}$ is then evaluated as:

$$\mathbf{g}(\hat{\mathbf{x}}(t, \mathbf{f}), \mathbf{y}(t), t; \mathbf{f}) = \mathbf{g}\left(\sum_{j=0}^P \mathbf{c}_j(t) \varphi_j(\mathbf{f}), \mathbf{y}(t), t; \mathbf{f}\right). \quad (4.26)$$

Taking the inner product with the basis $\varphi_i(\mathbf{f})$, the evolution equations for the coefficients are obtained as:

$$\frac{d\mathbf{c}_i(t)}{dt} = \left\langle \varphi_i(\mathbf{f}), \mathbf{g} \left(\sum_{j=0}^P \mathbf{c}_j(t) \varphi_j(\mathbf{f}), \mathbf{y}(t), t; \mathbf{f} \right) \right\rangle, \quad i = 0, \dots, P, \quad (4.27)$$

where the inner product is defined as:

$$\langle f(\mathbf{f}), h(\mathbf{f}) \rangle = \int_{\Omega} f(\mathbf{f}) h(\mathbf{f}) p(\mathbf{f}) d\mathbf{f}, \quad (4.28)$$

with $p(\mathbf{f})$ being the joint probability density function of the fault parameters.

The initial coefficients are obtained by projecting the random initial condition onto the basis:

$$\mathbf{c}_i(0) = \langle \varphi_i(\mathbf{f}), \hat{\mathbf{x}}_0(\mathbf{f}) \rangle, \quad i = 0, \dots, P. \quad (4.29)$$

Once the deterministic system governing the coefficients $\mathbf{c}_i(t)$ is solved, the statistical moments of the state $\hat{\mathbf{x}}(t, \mathbf{f})$ can be computed using the expressions provided in Proposition D:

- Mean:

$$\mathbb{E}[\hat{\mathbf{x}}(t)] = \mathbf{c}_0(t) \quad (4.30)$$

- Covariance Matrix:

$$\mathbb{C}[\hat{\mathbf{x}}(t)] = \sum_{i=1}^P \mathbf{c}_i(t) \mathbf{c}_i^\top(t). \quad (4.31)$$

The Galerkin-based gPC method transforms the original stochastic dynamical system into a deterministic system of higher dimension by projecting the solution onto an orthogonal polynomial basis. This intrusive approach yields accurate moment estimations for systems with moderate uncertainty dimensionality and smooth dynamics. However, it requires direct access to the governing equations, which may not be feasible in all applications. For high-dimensional or non-smooth systems, non-intrusive alternatives may be more appropriate.

The detailed formulation and derivation of the intrusive generalized polynomial chaos (gPC) method via Galerkin projection for stochastic ordinary differential equations (ODEs) are provided in Appendix D. This includes the mathematical foundation of the gPC expansion, the derivation of the coefficient dynamics, and the procedure for evaluating the required inner products using numerical quadrature. Furthermore, the extended formulation for uncertainty propagation in systems subject to both parametric uncertainty and additive process noise is presented in Appendix D.2.

Uncertainty Quantification in Linear Estimators via Intrusive gPC. Consider the stochastic system:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{R}_1\mathbf{f} \\ \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{R}_2\mathbf{f} \end{cases} \quad (4.32)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector and $\mathbf{f} \in \mathbb{R}^d$ is the unknown bias fault with a given distribution acting on the system. The continuous-time Kalman filter can be written as:

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{K}(\mathbf{y}(t) - \mathbf{C}\hat{\mathbf{x}}(t)) \quad (4.33)$$

while the dynamic of true state $\mathbf{x}(t)$ is given by:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{R}_1\mathbf{f} + \mathbf{K}(\mathbf{y}(t) - \mathbf{C}\mathbf{x}(t)) \\ &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + (\mathbf{R}_1 + \mathbf{K}\mathbf{R}_2)\mathbf{f}. \end{aligned} \quad (4.34)$$

Then, the dynamic of estimation uncertainty, $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$, will be:

$$\dot{\tilde{\mathbf{x}}}(t) = (\mathbf{A} - \mathbf{K}\mathbf{C})\tilde{\mathbf{x}}(t) + \mathbf{R}\mathbf{f} \quad (4.35)$$

where $\mathbf{A} - \mathbf{K}\mathbf{C}$ and $\mathbf{R} = (\mathbf{R}_1 + \mathbf{K}\mathbf{R}_2)$ are deterministic matrices, and the fault vector $\mathbf{f}$ is modeled as a random variable with known distribution. The initial condition $\hat{\mathbf{x}}_0(\mathbf{f}) \in L^2(\Omega; \mathbb{R}^n)$ may also be a function of $\mathbf{f}$.

Assuming that $\tilde{\mathbf{x}}(t, \mathbf{f}) \in L^2(\Omega; \mathbb{R}^n)$, it can be approximated using a truncated polynomial chaos expansion:

$$\tilde{\mathbf{x}}(t, \mathbf{f}) \approx \sum_{i=0}^P \mathbf{c}_i(t) \varphi_i(\mathbf{f}), \quad (4.36)$$

where $\varphi_i(\mathbf{f})$ are multivariate orthonormal polynomials with respect to the distribution of $\mathbf{f}$, P is the number of retained basis terms, and $\mathbf{c}_i(t) \in \mathbb{R}^n$ are deterministic coefficient vectors.

Substituting the gPC expansion into the linear system (4.35):

$$\frac{d}{dt} \left(\sum_{i=0}^P \mathbf{c}_i(t) \varphi_i(\mathbf{f}) \right) = (\mathbf{A} - \mathbf{K}\mathbf{C}) \sum_{i=0}^P \mathbf{c}_i(t) \varphi_i(\mathbf{f}) + \mathbf{R}\mathbf{f}. \quad (4.37)$$

Projecting both sides onto each basis function $\varphi_k(\mathbf{f})$ yields:

$$\frac{d\mathbf{c}_k(t)}{dt} = (\mathbf{A} - \mathbf{K}\mathbf{C}) \mathbf{c}_k(t) + \langle \varphi_k(\mathbf{f}), \mathbf{R}\mathbf{f} \rangle, \quad k = 0, \dots, P. \quad (4.38)$$

The inner product term can be evaluated component-wise. Let the fault signal $\mathbf{f}$ be represented by its gPC expansion as $\mathbf{f} = \sum_{j=0}^P \boldsymbol{\alpha}_j \varphi_j(\mathbf{f})$. Then,

$$\langle \varphi_k(\mathbf{f}), \mathbf{R}\mathbf{f} \rangle = \mathbf{R} \sum_{j=0}^P \boldsymbol{\alpha}_j \langle \varphi_k, \varphi_j \rangle = \mathbf{R}\boldsymbol{\alpha}_k, \quad (4.39)$$

where the final equality follows from the orthonormality condition $\langle \varphi_k, \varphi_j \rangle = \delta_{kj}$. The coefficients $\boldsymbol{\alpha}_k$ can be computed by projection as $\boldsymbol{\alpha}_k = \mathbb{E}_p[\mathbf{f} \varphi_k(\mathbf{f})]$, which may be evaluated either analytically for known distributions or numerically using sampling-based integration.

Hence, the evolution equations for the gPC coefficients become:

$$\frac{d\mathbf{c}_k(t)}{dt} = (\mathbf{A} - \mathbf{K}\mathbf{C}) \mathbf{c}_k(t) + \mathbf{R}\boldsymbol{\alpha}_k, \quad k = 0, \dots, P. \quad (4.40)$$

The initial condition for coefficients are obtained through projection, as shown in (4.29). Finally, the mean $\mathbb{E}[\tilde{\mathbf{x}}(t)]$ and covariance $\mathbb{C}[\tilde{\mathbf{x}}(t)]$ of the estimation error due to the fault source $\mathbf{f}$ are computed using (4.30) and (4.31), respectively.

4.3.3 Uncertainty Quantification via Non-intrusive gPC (Stochastic Collocation Method)

To quantify the impact of faults on state estimation accuracy, the stochastic collocation method offers a non-intrusive and efficient framework. Unlike intrusive approaches such as Galerkin projection, stochastic collocation does not require access to or modification of the internal structure of the state estimator. Instead, it treats the estimator as a black-box function and evaluates it at selected points in the input uncertainty space to construct a surrogate model using polynomial chaos expansion. This section focuses on the application of stochastic collocation to propagate uncertainty in the presence of fault inputs with known probability distributions.

Consider a general estimator defined by the following dynamics:

$$\hat{\mathbf{x}}(t) = \mathbf{g}(\hat{\mathbf{x}}(t), \mathbf{y}(t), \mathbf{f}(t), t), \quad \hat{\mathbf{x}}(0) = \mathbf{x}_0 \quad (4.41)$$

where $\hat{\mathbf{x}}(t) \in \mathbb{R}^n$ is the estimated state vector, $\mathbf{y}(t) \in \mathbb{R}^m$ is the measurement vector, and $\mathbf{f}(t) \in \mathbb{R}^d$ represents fault inputs modeled as random variables with known joint probability distribution. The true state $\mathbf{x}(t)$ is assumed to be governed by a fault-free or faulty process

model. The goal is to quantify the propagation of the estimation error $\tilde{\mathbf{x}}(t) = \hat{\mathbf{x}}(t) - \mathbf{x}(t)$ due to uncertainties in the fault variables $\mathbf{f}(t)$.

Stochastic collocation requires the selection of discrete collocation points in the uncertainty space, over which the estimator is evaluated. These points are determined using quadrature rules that balance accuracy and computational cost. Common one-dimensional quadrature rules include:

- **Newton–Cotes rules**, which use evenly spaced points and are easy to implement but can suffer from poor convergence for high-order expansions.
- **Gaussian quadrature**, which achieves exact integration for polynomials of degree up to $2n - 1$ using only n points, offering high accuracy but non-nested grids.
- **Clenshaw–Curtis quadrature**, which uses Chebyshev points of the first kind and is particularly advantageous due to its nested structure. Nested grids enable efficient construction of sparse collocation schemes and are especially useful when adaptivity or hierarchical refinement is required. Therefore, Clenshaw–Curtis quadrature is adopted in this study to construct the collocation points.

For multivariate uncertainties, tensor-product or sparse grid constructions can be employed using one-dimensional rules as building blocks.

Projection-Based Derivation of gPC Coefficients. Given a set of collocation points $\{\boldsymbol{\xi}^{(i)}\}_{i=1}^N$ in the fault space, the estimator is simulated for each realization of $\mathcal{F}(t) = \boldsymbol{\xi}^{(i)}$ to obtain the corresponding estimation $\hat{\mathbf{x}}(t; \boldsymbol{\xi}^{(i)})$. The estimation error is then approximated using a polynomial chaos expansion:

$$\hat{\mathbf{x}}(t; \boldsymbol{\xi}) \approx \sum_{k=0}^P \mathbf{c}_k(t) \varphi_k(\boldsymbol{\xi}) \quad (4.42)$$

where $\{\varphi_k(\boldsymbol{\xi})\}$ are multivariate orthogonal polynomials with respect to the distribution of $\mathcal{F}$, and $\mathbf{c}_k(t) \in \mathbb{R}^n$ are the PCE coefficients. In the projection-based approach, these coefficients are computed using weighted quadrature:

$$\mathbf{c}_k(t) = \frac{1}{\langle \varphi_k, \varphi_k \rangle} \sum_{i=1}^N \hat{\mathbf{x}}(t; \boldsymbol{\xi}^{(i)}) \varphi_k(\boldsymbol{\xi}^{(i)}) w^{(i)} \quad (4.43)$$

where $w^{(i)}$ are the quadrature weights associated with each collocation point, and $\langle \cdot, \cdot \rangle$ denotes the weighted inner product.

Once the coefficients $\mathbf{c}_i(t)$ is obtained, the statistical moments of the states $\mathbf{x}(t, \mathbf{f})$ can be computed using the expressions provided in Proposition D:

- Mean:

$$\mathbb{E}[\widehat{\mathbf{x}}(t)] = \mathbf{c}_0(t) \quad (4.44)$$

- Covariance Matrix:

$$\mathbb{C}[\widehat{\mathbf{x}}(t)] = \sum_{i=1}^P \mathbf{c}_i(t) \mathbf{c}_i^\top(t). \quad (4.45)$$

Regression-Based Derivation of gPC Coefficients. As an alternative to projection, a regression-based approach can be used to compute the gPC coefficients. Instead of requiring quadrature weights and orthogonality conditions, this method formulates the problem as a least squares fit:

$$\min_{\{\mathbf{c}_k(t)\}} \sum_{i=1}^N \left\| \widehat{\mathbf{x}}(t; \boldsymbol{\xi}^{(i)}) - \sum_{k=0}^P \mathbf{c}_k(t) \varphi_k(\boldsymbol{\xi}^{(i)}) \right\|^2. \quad (4.46)$$

This method offers greater flexibility for black-box models, noisy outputs, or irregular sampling strategies, as it does not rely on predefined quadrature rules. While the projection-based method is adopted in this work due to its analytical precision and compatibility with the fault distribution structure, the regression-based formulation remains a valuable alternative in scenarios where analytical integration or structured sampling is impractical.

4.3.4 Computational Complexity

The surrogate-based UQ method based on gPC provides an approach that balances accuracy and computational cost. The computational complexity of the gPC method primarily depends on the number of quadrature points N_q and the number of time steps T . For a univariate stochastic input with polynomial degree P , projection-based stochastic collocation requires $N_q = P + 1$ simulations and computes the coefficients $c_k(t)$ by orthogonal projection at each time step. As shown in Section 4.3.3, the coefficients are obtained via weighted inner products involving polynomial basis evaluations, resulting in a per-time-step cost of $\mathcal{O}(PN_q)$. Over all time steps, the total cost scales as $\mathcal{O}(PN_qT)$, which remains linear in both the number of basis functions and simulation length.

4.4 Uncertainty Quantification for Learning-Based Estimators

Uncertainty quantification (UQ) for deep neural network-based estimators can be achieved using methods such as Monte Carlo (MC) dropout and conformal prediction.

4.4.1 Monte Carlo Dropout

Monte Carlo dropout performs approximate Bayesian inference by enabling dropout during inference and sampling the network multiple times. This generates an empirical distribution of predictions from which the predictive mean and variance are estimated. Despite its ease of implementation, MC dropout requires tens to hundreds of forward passes per input, making it computationally expensive for real-time applications, especially on embedded systems.

4.4.2 Conformal Prediction

Conformal prediction provides statistical coverage guarantees by calibrating prediction intervals on a held-out validation dataset. It constructs prediction sets that contain the true output with a specified confidence level, under the assumption of data exchangeability. However, this assumption does not hold for time-series data due to the inherent temporal dependencies and non-stationarity in dynamical systems. To address this, modified versions such as sliding-window or weighted conformal prediction can be employed, where only recent data points are used for calibration or higher weights are assigned to recent samples. While these adaptations improve suitability for time-dependent state estimation, the practical use of conformal prediction in real-time settings remains limited by the need to maintain calibration statistics and compute nonconformity scores during inference, especially in high-rate or resource-constrained applications.

These approaches are model-agnostic and widely applicable but often introduce considerable computational challenges in real-time applications.

4.5 Summary

This chapter presented a comprehensive framework for quantifying estimation uncertainty under fault conditions across various estimator types, including both model-based and

learning-based approaches. The focus was on developing computationally efficient methods suitable for real-time applications in complex interconnected systems.

In Section 4.2, an analytical UQ method was introduced for Kalman-based estimators. By exploiting the linear structure and statistical properties of process and measurement faults, closed-form expressions for the estimation covariance were derived. This enabled fast and reliable UQ with minimal computational overhead, making it suitable for online deployment.

Section 4.3 introduced generalized Polynomial Chaos (gPC) methods for UQ in systems where analytical formulations are either unavailable or impractical. Both intrusive (Galerkin-based) and non-intrusive (stochastic collocation-based) formulations were presented. These methods enabled real-time UQ in nonlinear or black-box estimators without relying on computationally expensive sampling-based techniques such as Monte Carlo simulation.

Section 4.4 extended the discussion to learning-based estimators, such as neural networks. Two widely used uncertainty quantification techniques were presented: Monte Carlo (MC) dropout and conformal prediction. While these methods are flexible and compatible with black-box models, they come with significant computational burdens, particularly in real-time or embedded scenarios.

Overall, this chapter contributed a unified and scalable approach to UQ in both structured (Kalman-based) and unstructured (black-box) estimators, with an emphasis on runtime feasibility. These tools enable reliable, real-time estimation in complex dynamical systems and support downstream tasks such as fault-tolerant path selection and estimator reconfiguration.

Chapter 5

A Framework for Fault Detection, Isolation, and Identification

This chapter introduces a unified framework for fault detection, isolation, and identification, combining model-based statistical tests with data-driven learning methods. A three-stage framework for fault detection and diagnosis (FDD) is proposed in this work. The first stage detects the onset time of the fault, denoted by t_f . The second stage performs fault isolation to determine which specific fault i_f has occurred. Finally, the third stage carries out fault identification to estimate the fault magnitude $\hat{f}$. Figure 5.1 illustrates the proposed FDD architecture along with the different approaches applicable at each stage.

Fault detection is addressed through three approaches: the χ^2 -test, neural network-based classifiers, and a hybrid voting scheme that leverages both. For fault isolation, model-based likelihood testing, deep learning classifiers, and hybrid methods are explored. The chapter concludes with fault identification, focusing on estimating fault characteristics after isolation.

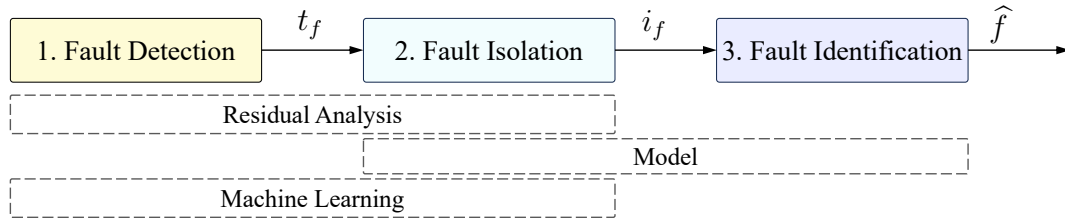


Figure 5.1: Overview of proposed 3-stage fault detection and diagnosis (FDD) framework.

5.1 Fault Detection

Fault detection is the first stage in the proposed fault diagnosis process and involves determining when a fault has occurred in the system. This section presents both model-based and data-driven approaches for detecting faults, as well as a hybrid strategy that integrates their strengths to improve detection accuracy and robustness.

5.1.1 Residual-Based Fault Detection

The χ^2 -test [50] is a statistical method for fault detection, effective in well-modeled systems with accurately predicted residual behavior under normal conditions. In fault diagnosis, a residual is defined as the inconsistency between the actual behavior of the system and its predicted behavior derived from a mathematical model. The residual sequences $\mathbf{r}_k$ are expected to have zero mean and covariance Σ_k under no fault. The precomputed Σ_k represents the expected residual distribution in non-faulty conditions. The detection criterion is given by the sum of χ^2 statistics over a sliding window of size m

$$Z_k = \sum_{t=k-m+1}^k \mathbf{r}_t^\top \Sigma_t^{-1} \mathbf{r}_t \underset{H_0}{\overset{H_1}{\geq}} \epsilon \quad (5.1)$$

where the null hypothesis H_0 denotes normal system condition, and the alternative hypothesis H_1 indicates a fault, shifting the mean of residuals from zero. The statistic Z_k follows a χ^2 distribution with $p \times m$ degrees of freedom ($p = \dim(\mathbf{r})$). A fault is flagged if Z_k exceeds a threshold ϵ , chosen from the χ^2 distribution table for the desired significance level (e.g., 5%). The sliding window helps smooth out noise or transients, improving detection reliability and reducing false alarms.

The χ^2 -test, though widely used, has certain limitations. In practice, residuals often deviate from a white, zero-mean distribution due to unmodeled dynamics, uncertainties, or disturbances, leading to false positives. Although the sliding window technique helps mitigate these issues, it can introduce delays in detection. Consequently, the χ^2 -test requires supplementation with additional techniques to handle the complexities of real-world applications. Data-driven approaches, which can better capture patterns in residuals and other system signals under both normal and faulty conditions, provide a more accurate alternative to traditional model-based methods.

5.1.2 Learning-Based Fault Detection

This part explores using a neural network for fault detection, offering a robust alternative to traditional methods like the χ^2 -test. Neural networks excel at capturing complex data patterns and, unlike threshold-based methods, can be trained on extensive datasets to distinguish between normal deviations and actual faults, reducing false alarms.

The proposed fault detection model in Figure 5.2 is designed to capture both spatial and temporal dependencies in time series data, crucial for accurate fault onset detection. It starts with a 1D convolutional layer:

$$z_i^{(k)} = \sigma \left(\sum_{j=0}^{K-1} w_j^{(k)} \cdot x_{i+j} + b^{(k)} \right), \quad (5.2)$$

where $x_{i+j} \in \mathbb{R}^p$ is the input at position $i + j$, $w_j^{(k)}$ is the weight of the j -th position of the k -th kernel of width K , $b^{(k)}$ is the corresponding bias, and $\sigma(\cdot)$ is a non-linear activation function such as ReLU. The output $z_i^{(k)}$ is the activation at position i in the k -th feature map. This layer extracts local spatial features from each time window of the multivariate signal. It is followed by a recurrent neural network (RNN) layer, which captures temporal dependencies by maintaining an internal state updated at each time step. The RNN layer depicted in Figure 5.3 operates as:

$$h_t = \sigma(W_h \cdot h_{t-1} + W_x \cdot x_t + b_h), \quad (5.3)$$

where h_t is the hidden state at time t , W_h and W_x are the weight matrices for the hidden state and input x_t , and b_h is the bias. The activation function $\sigma(\cdot)$, typically a hyperbolic tangent ($\tanh$), introduces non-linearity. This combination of convolutional and RNN layers effectively learns spatial and temporal patterns, key for distinguishing between faulty and normal conditions. The network ends with a dense layer and a sigmoid-activated output layer for binary classification. A binary cross-entropy loss function is employed to optimize the classification task.

To prepare the raw time-series data for training and testing, a sliding window approach is applied; see Figure 5.4. The original data consists of p features sampled over T time steps. Using a window of fixed length l and sliding step m , overlapping segments are extracted sequentially from the time series. Each windowed segment forms a sample $\mathbf{X}_i \in \mathbb{R}^{p \times l}$, and a total of n such windows are generated. In this work, a window length of $l = 40$ and step size $m = 1$ are used, resulting in a dataset $\mathcal{X} \in \mathbb{R}^{p \times l \times n}$. Each window is assigned a label corresponding to the label at its final time step. This preprocessing strategy enables the

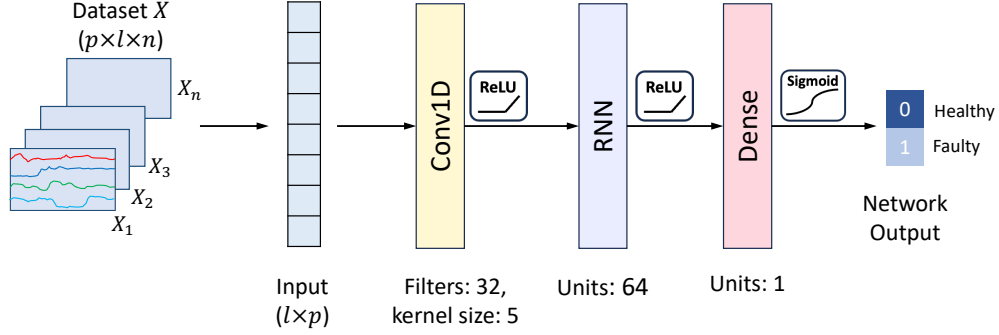


Figure 5.2: Network architecture for fault detection. The model combines a 1D convolutional layer to capture local spatial features, and an RNN layer to capture temporal dependencies. A dense layer with a sigmoid output performs binary classification to detect faults.

neural network to learn temporal patterns from fixed-length time series segments while preserving the multi-feature structure of the original signals.

The ability to learn from data and adapt to unforeseen uncertainties makes learning-based detection essential in ensuring the reliability and safety of large-scale systems. Despite their high accuracy and low false positive rate, normal zero-mean residuals can still trigger false alarms, especially due to other system signals used as features. Therefore, combining threshold-based and learning-based methods via a voting logic can further enhance detection accuracy.

5.1.3 Hybrid Voting-Based Fault Detection

To further enhance fault detection accuracy, the threshold-based χ^2 -test is integrated with the neural network using voting logic. This approach leverages the robustness of the χ^2 -test and the pattern recognition of the neural network. A fault is declared when both methods agree, using AND logic to combine their detection outputs at each time step. This integration reduces false positive alarms and enhances detection accuracy by addressing the limitations of each method individually.

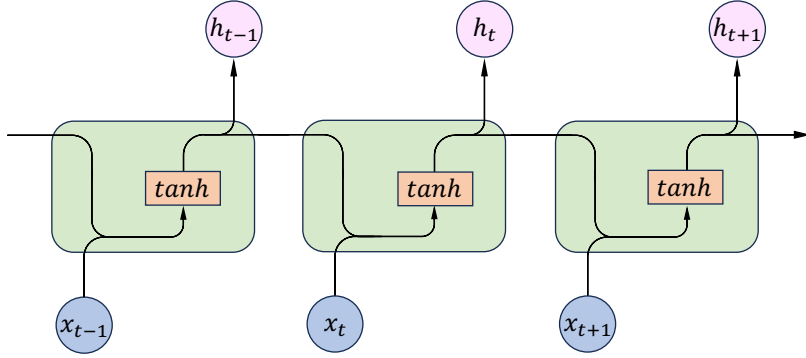


Figure 5.3: The Architecture of simple recurrent neural network (RNN) layer.

5.2 Fault Isolation

5.2.1 Residual-Based Fault Isolation

To facilitate fault isolation, a *fault signature matrix* is constructed, encoding the qualitative sensitivity of each residual to different fault sources. This matrix characterizes how specific faults (e.g., sensor bias, actuator failure, or external disturbances) affect the residuals, enabling systematic reasoning for identifying the most likely fault cause based on observed deviations.

An example fault signature matrix is shown in Table 8.1, where each entry indicates whether a given residual r_i is sensitive to a specific fault f_j . Each column represents a potential fault, and each row represents a residual or indicator sensitive to different faults. Entries of 1 represent a detectable influence of the fault on the residual (i.e., the fault affects the residual), whereas 0 indicates no detectable effect. In practice, residuals can be analytically or empirically designed to follow structured sensitivity patterns, allowing for unique localization of faults using Boolean decision logic or threshold-based classification. A residual is typically considered impacted by a fault if it exceeds a predefined threshold, enabling practical decision-making based on observed deviations.

From a theoretical standpoint, the fault signature matrix can also be interpreted as the fault-to-residual transfer matrix defined in Section A.5:

$$\mathbf{G}_{rf}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{R}_1 + \mathbf{R}_2,$$

which maps the effect of each fault f_j to the residual output. For a system to be capable of isolating all faults, this matrix must satisfy certain properties:

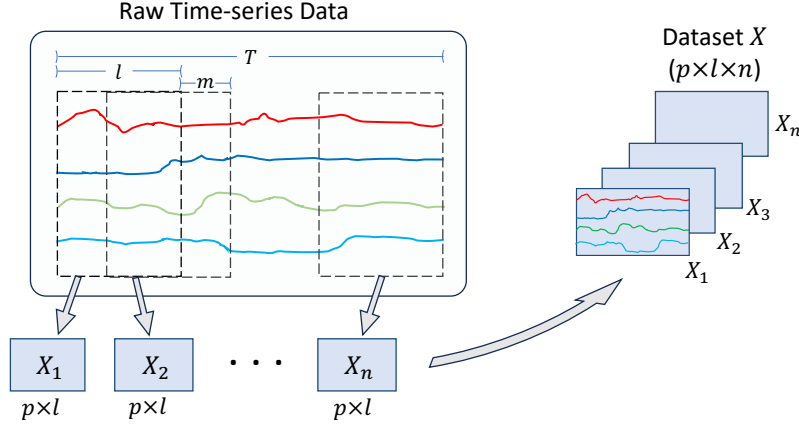


Figure 5.4: Preprocessing of raw time-series data for neural network training and testing.

- **Full Column Rank:** For all faults to be isolable, the fault signature matrix must have full column rank equal to the number of faults, i.e.,

$$\text{rank}(\mathbf{G}_{rf}(s)) = \ell,$$

where ℓ is the number of faults. This ensures that each fault produces a linearly independent signature in the residual space, making it distinguishable from the others. For instance, the fault signature matrix in Table 8.1 has full column rank, indicating that all faults are uniquely isolable.

- **Column Distinctness:** A fault f_i is isolable if the i -th column of the fault signature matrix, $\mathbf{G}_{rf}^{(i)}$, is unique—i.e., it is not a linear combination of the other columns. This guarantees that fault f_i can be distinguished from the remaining faults.
- **Sensor Limit Constraint:** The number of isolable faults is fundamentally limited by the number of independent residuals (or sensors). Specifically,

$$\text{rank}(\mathbf{G}_{rf}(s)) \leq \min\{m, \ell\},$$

where m is the number of available sensors or residuals. This imposes a practical constraint on the maximum number of faults that can be reliably isolated.

These conditions provide a theoretical foundation for designing effective fault isolation schemes and evaluating the isolability of faults in a given system structure.

Table 5.1: Example fault signature matrix indicating the sensitivity of each residual r_i to the fault scenarios f_j .

	f_1	f_2	f_3	f_4	f_5	f_6
r_1	0	1	1	1	1	1
r_2	0	1	1	1	1	1
r_3	0	1	0	1	1	1
r_4	0	0	0	1	1	1
r_5	1	0	1	1	0	1
r_6	1	0	1	0	1	1
r_7	1	1	1	0	1	0
r_8	1	1	0	1	1	0

5.2.2 Model-Based Likelihood Test for Fault Isolation

Fault isolation with multiple fault sources can be efficiently managed using multiple hypothesis testing, provided the residual distributions are known for each hypothesis. The proposed approach leverages prior knowledge of fault distributions to quantify how each fault affects the residuals, enabling localization of the most likely fault. The hypotheses are defined as H_i for each fault type.

Likelihood Test. Given the residuals vector $\mathbf{r}_k$ at time step k , the likelihood $\mathcal{L}(\mathbf{r}_k \mid H_i)$ of the residuals belonging to fault type H_i is calculated as:

$$\mathcal{L}(\mathbf{r}_k \mid H_i) = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}_k(i)|^{1/2}} \exp \left(-\frac{1}{2} \mathbf{r}_k^T \boldsymbol{\Sigma}_k^{-1}(i) \mathbf{r}_k \right) \quad (5.4)$$

where $\boldsymbol{\Sigma}_k(i)$ is the auto-covariance matrix of the residuals $\mathbf{r}_k$ for fault type H_i , derived in (5.7). To improve robustness and reduce false alarms, a sliding window of size l is used, summing the log-likelihoods over the window while ignoring the constant term:

$$\sum_{t=k-l+1}^k \log \mathcal{L}(\mathbf{r}_t \mid H_i) = -\frac{l}{2} \log |\boldsymbol{\Sigma}_k(i)| - \frac{1}{2} \sum_{t=k-l+1}^k \mathbf{r}_t^T \boldsymbol{\Sigma}_k^{-1}(i) \mathbf{r}_t \quad (5.5)$$

The fault type is identified by selecting the hypothesis H_i that maximizes the cumulative log-likelihood of the residuals over the window $[k-l+1 : k]$:

$$H^* = \arg \max_i \sum_{t=k-l+1}^k \log \mathcal{L}(\mathbf{r}_t \mid H_i).$$

The likelihood test isolates faults by leveraging how each fault uniquely affects residuals. In the following, the residual covariance matrix $\Sigma(i)$ in (5.5) is quantified for fault class i across the estimation graph.

Quantification of Residual Uncertainty Covariance. The state estimation error covariance due to fault propagation is quantified in a Kalman-based estimation graph in Section 4.2. A similar approach is used here to quantify residual uncertainty as a fault indicator.

Consider the estimation graph in Figure 4.1. The measurement residual, $\mathbf{r}_k := \mathbf{y}_k - \widehat{\mathbf{y}}_k^-$, of node i at time k is given by

$$\mathbf{r}_k^i = \sum_{u \in \mathcal{I}} \mathbf{H}_k^{iu} \tilde{\mathbf{x}}_k^{i-} + \Lambda_k^i \mathbf{f}_{b,k}^i + \boldsymbol{\eta}_k^i \quad (5.6)$$

where $\mathcal{I} = \mathcal{N}^{in}(i)$ and $\tilde{\mathbf{x}}_k := \mathbf{x}_k - \hat{\mathbf{x}}_k^-$. Since the focus is solely on fault effects, measurement noise η_k is neglected under normal operating conditions. The process fault $\mathbf{f}_b^i \in \mathbb{R}^p$ and output fault $\mathbf{f}_d^i \in \mathbb{R}^q$ follow the following properties given by (A.24); and the cross-covariances $\mathbf{D}^{ij} = \mathbb{E}[\mathbf{f}_d^i \mathbf{f}_d^{j\top}]$ and $\mathbf{B}^{ij} = \mathbb{E}[\mathbf{f}_b^i \mathbf{f}_b^{j\top}]$ are non-zero if nodes i and j share same outputs or process faults, and $\mathbb{E}[\mathbf{f}_b^i \mathbf{f}_d^{j\top}] = \mathbf{0}$. Both excessive noise and bias faults are addressed similarly. The time subscript k is neglected for simplicity. Given the linearity of Kalman-based estimators and the fault distributions in (A.24), residuals follow a zero-mean distribution. Thus, quantifying the residual covariance suffices to determine their distribution.

The cross-covariance matrix between residuals of two arbitrary nodes i and j at step k is defined as $\Sigma_k^{ij} = \mathbb{E}[\mathbf{r}_k^i \mathbf{r}_k^{j\top}]$. Using the factorization $\mathbb{E}[\tilde{\mathbf{x}}_k^{i-} \mathbf{f}_{b,k}^{j\top}] = \mathbf{M}_k^{ij} \mathbf{B}_k^j$, it simplifies to

$$\begin{aligned} \Sigma_k^{ij} = & \sum_{u \in \mathcal{I}, v \in \mathcal{J}} \mathbf{H}_k^{iu} \mathbf{P}_k^{uv-} \mathbf{H}_k^{jv\top} + \sum_{u \in \mathcal{I}} \mathbf{H}_k^{iu} \mathbf{M}_k^{uj} \mathbf{B}_k^j \Lambda_k^{j\top} + \dots \\ & \Lambda_k^i \mathbf{B}_k^i \sum_{v \in \mathcal{J}} \mathbf{M}_k^{vi\top} \mathbf{H}_k^{jv\top} + \Lambda_k^i \mathbf{B}_k^{ij} \Lambda_k^{j\top} \end{aligned} \quad (5.7)$$

where the *a priori* error covariance, $\mathbf{P}_k^{ij-} = \mathbb{E}[\tilde{\mathbf{x}}_k^{i-} \tilde{\mathbf{x}}_k^{j-\top}]$, is derived in (4.17), and recursive equation to calculate the influence matrix, $\mathbf{M}_k^{ij}$, is derived in Appendix.

In practice, not all estimation residuals are used for isolation. Instead, a sub-vector $\mathbf{r}_k^{\mathcal{P}_i}$ may be selected from $\mathbf{r}_k^i$ for each estimator i in the estimation graph, where $\mathcal{P}_i \subseteq$

$\{p_1, p_2, \dots, p_{n_i}\}$ denotes the indices of the selected residuals. The selected residual vector $\mathbf{r}_k$ from all estimators in the estimation graph is given by

$$\mathbf{r}_k = [\mathbf{r}_k^{\mathcal{P}_1} \quad \mathbf{r}_k^{\mathcal{P}_2} \quad \dots \quad \mathbf{r}_k^{\mathcal{P}_N}].$$

The covariance matrix for $\mathbf{r}_k$, denoted as Σ_k , reflects the covariance between the selected residuals from different nodes. It is given by:

$$\Sigma_k = \begin{bmatrix} \Sigma_k^{\mathcal{P}_1 \mathcal{P}_1} & \Sigma_k^{\mathcal{P}_1 \mathcal{P}_2} & \dots & \Sigma_k^{\mathcal{P}_1 \mathcal{P}_N} \\ \Sigma_k^{\mathcal{P}_2 \mathcal{P}_1} & \Sigma_k^{\mathcal{P}_2 \mathcal{P}_2} & \dots & \Sigma_k^{\mathcal{P}_2 \mathcal{P}_N} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_k^{\mathcal{P}_N \mathcal{P}_1} & \Sigma_k^{\mathcal{P}_N \mathcal{P}_2} & \dots & \Sigma_k^{\mathcal{P}_N \mathcal{P}_N} \end{bmatrix} \quad (5.8)$$

where $\Sigma_k^{\mathcal{P}_i \mathcal{P}_j}$ is a sub-matrix of Σ_k^{ij} corresponding to indices $\mathcal{P}_i$ and $\mathcal{P}_j$.

5.2.3 Learning-Based Fault Isolation

Model-based likelihood tests are prone to false alarms, particularly under unmodeled dynamics, unexpected disturbances, or system nonlinearities that distort the statistical assumptions of the residuals. Additionally, for small fault magnitudes the likelihoods might be close to each other, complicating isolation. To address these challenges, learning-based methods using historical data offer an alternative capable of capturing complex residual patterns that arise under various fault conditions. These methods do not rely on strict analytical fault models and are particularly useful when the fault signatures are nonlinear or difficult to characterize using first-principles models.

A supervised neural network architecture is designed to perform multi-class fault isolation, where each class corresponds to a distinct fault source. The architecture, shown in Figure 5.5, combines a 1D convolutional layer to extract spatial features from multivariate residual signals and an RNN layer to model temporal dependencies in the fault progression over time. This combination enables the model to effectively capture both local and sequential patterns in the data that are indicative of different fault types. The final dense layer uses a *softmax* activation function to produce a probability distribution over possible fault sources for multi-class classification.

Despite their advantages, purely data-driven approaches suffer from several limitations. The availability of labeled fault data is often limited, especially for rare or safety-critical faults, leading to class imbalance and biased learning. Furthermore, training data may not

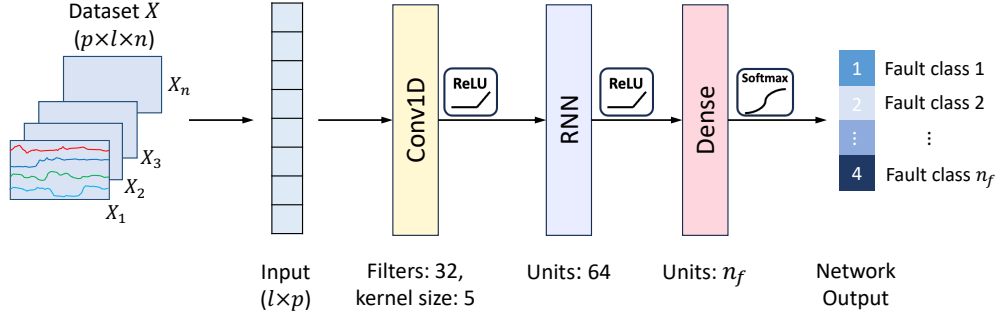


Figure 5.5: Network architecture for fault isolation. The model combines a 1D convolutional layer to capture local spatial features, and an RNN layer to capture temporal dependencies. A dense layer with a softmax output performs multi-class classification to isolate the fault source.

adequately represent the full range of operating conditions or fault magnitudes, reducing the classifier’s generalization ability. Centralized approaches also face scalability issues, as the computational complexity and memory requirements grow rapidly with system size, making real-time deployment difficult. Additionally, the need for complete retraining when new fault types or uncertainties arise limits adaptability in evolving systems. These limitations motivate the integration of prior system knowledge into the learning process, leading to a hybrid approaches.

5.2.4 Hybrid Fault Isolation via Feature Integration

A hybrid fault isolation method is proposed, wherein model-based residuals and likelihood features are integrated into a neural network to enhance fault classification accuracy. This approach is especially effective when data is insufficient or imbalanced, as it incorporates prior knowledge of fault distributions and model information with the available data. The neural network captures complex patterns in the features, enhancing robustness against uncertainties and unmodeled dynamics.

The hybrid neural network model, shown in Figure 5.6, has two parallel branches: (1) the first processes original features from the residuals, using a 1D Convolutional Layer, a Simple RNN Layer and a flattening step; (2) the second processes features from the likelihood quantifiers in (5.5), following the same structure with a 1D Convolutional Layer,

Simple RNN Layer, and flattening step. After processing in both branches, the outputs are concatenated and passed through Dense layers. The final output layer uses a *softmax* activation function for multi-class fault classification.

The hybrid fault isolation approach offers significant advantages, especially in scenarios where data is limited or imbalanced. By integrating prior knowledge of fault distributions with model-based results, this approach supplements the data, leading to higher classification accuracy. Model-based likelihoods offer a solid foundation, while the data-driven neural network captures complex patterns in the residuals, improving robustness against uncertainties and unmodeled dynamics. This hybrid approach compensates for the limitations of pure model-based methods by creating enriched feature sets that combine the strengths of both approaches.

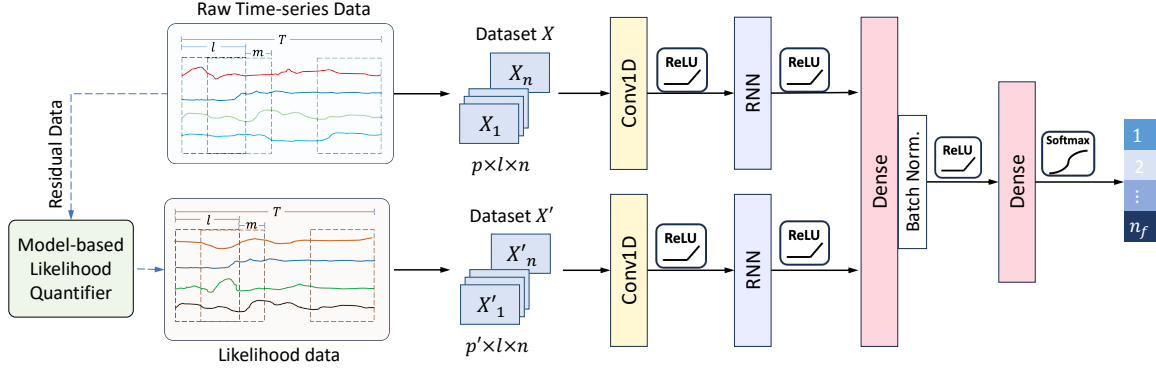


Figure 5.6: Hybrid structure for fault isolation. Likelihoods for each fault are quantified mathematically by (5.4). A sliding window is applied to time-series features, with each window labeled by the last time step. The two network branches are then integrated.

While integrating model-based likelihood testing with data-driven methods can improve isolation accuracy, this benefit relies on the availability of accurate subsystem models and prior knowledge on fault distributions. On the other hand, purely learning-based centralized fault diagnosis schemes become computationally intractable and inflexible in large-scale systems. This is because, as the number of potential faults grows, so does the dimensionality of required fault signature inputs, leading to exponentially larger models, longer training times, and heavier inference demands. Moreover, class imbalance and scarce training data for rare faults hinder generalization, while adding new fault types forces costly retraining or re-design of the entire model. These limitations are well recognized in [122] and highlight the need for a distributed and structure-aware fault isolation framework that

is both computationally efficient and capable of operating under partial supervision. The next two subsections present the design and evaluation of such a framework.

5.2.5 Hybrid Consensus-Based Fault Isolation

Figure 5.7 provides an overview of the proposed distributed fault isolation framework. The process begins with the residual generation as fault signatures, followed by the extraction of a fault signature matrix that structurally encodes fault-residual dependencies. This matrix is then used to build the fault interaction graph (FIG), which captures the structural overlap between fault signatures. Each fault node in the FIG is assigned either a supervised classifier (if labeled data are available) or an unsupervised autoencoder (if only healthy data are present) to perform local fault detection. Finally, a consensus-based decision making mechanism fuses the local scores across the graph to achieve a robust and interpretable global fault isolation decision. The remaining sections describe these components in detail.

Bipartite Fault Signature Graph. The fault signatures are structurally connected to the underlying fault sources through the system’s physical and analytical relationships. To capture this connectivity, a bipartite fault signature graph (BFSG) is constructed, representing the structured dependencies between faults and residuals. This graph is derived from the fault signature matrix, where one node set corresponds to fault sources and the other to residual signals. An edge connects a fault node to a residual node if the corresponding residual is sensitive to that fault. The BFSG enables systematic fault isolation by mapping observed residual patterns to known fault signatures. An illustrative example of the BFSG for a vehicle system is shown in Figure 5.8a, where each fault node is connected to the residuals it influences based on the fault signature matrix.

Fault Interaction Graph. To capture the direct relationships between fault sources based on shared residual influences, the *Fault Interaction Graph* (FIG) is defined. The FIG is derived from the BFSG by projecting the bipartite structure onto the fault nodes; see Figure 5.8b. This process, known as a *one-mode projection*, eliminates the intermediate residual nodes and connects two fault nodes with an undirected edge if they share at least one common residual in the BFSG. Formally, given a bipartite graph $\mathcal{B} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$, where $\mathcal{U}$ represents residual nodes and $\mathcal{V}$ represents fault nodes, the one-mode projection onto $\mathcal{V}$ results in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}')$, where an edge $(v_i, v_j) \in \mathcal{E}'$ exists if $\mathcal{N}(v_i) \cap \mathcal{N}(v_j) \neq \emptyset$, with $\mathcal{N}(v)$ denoting the neighborhood of node v in $\mathcal{G}$; see [123]. This structural connectivity reflects the potential ambiguity in fault isolation due to overlapping residual signatures.

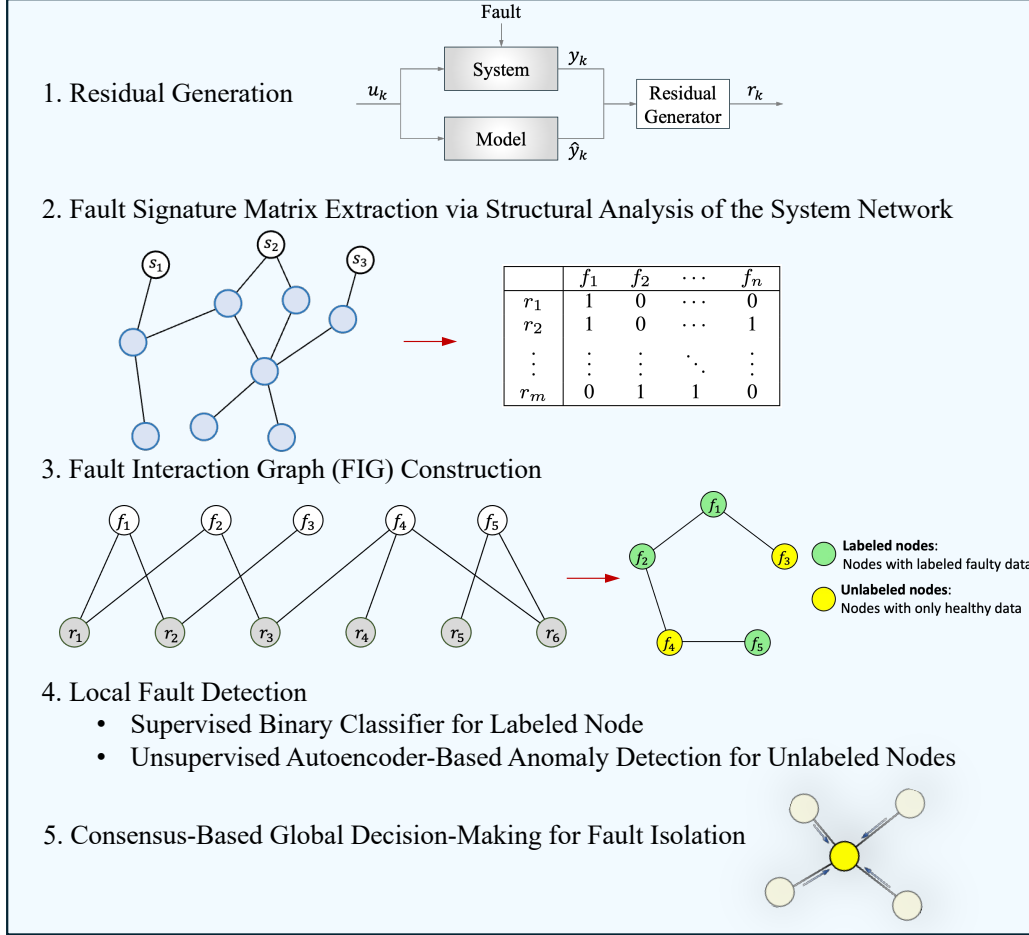


Figure 5.7: Framework for distributed fault isolation in networked dynamical systems.

To derive the FIG from the BFGS, a weighted one-mode projection onto the fault nodes is performed. Details on the one-mode projection process are provided in Appendix E. This projection captures the direct structural relationships between faults based on their shared influence on residual signals. Given the weighted adjacency matrix $\mathbf{A} \in \mathbb{R}^{n_r \times n_f}$ of the BFGS, where n_r and n_f denote the number of residuals and faults, respectively, the one-mode projection proceeds as follows:

1. **Construction of the Bipartite Graph:** A bipartite graph $\mathcal{B} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ is constructed, where $\mathcal{U}$ and $\mathcal{V}$ represent residual and fault nodes, respectively, and $\mathcal{E}$

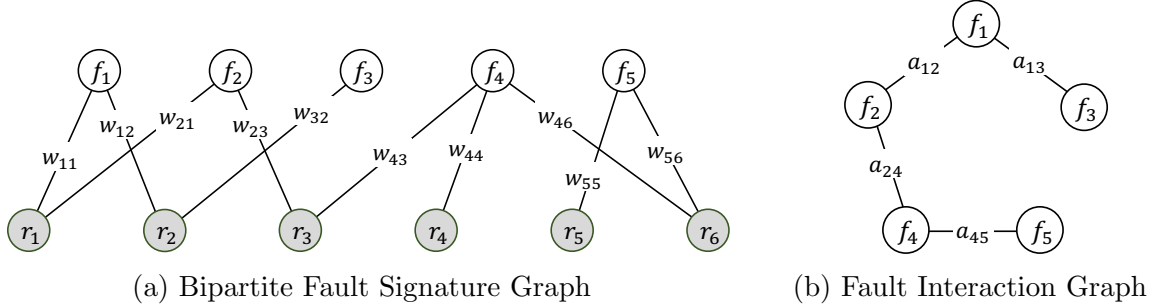


Figure 5.8: Illustration of (a) Bipartite Fault Signature Graph (BFGS), showing the relationships between faults and their associated residual signatures; and (b) Fault Interaction Graph (FIG), depicting direct relationships between faults based on shared residuals via one-mode projection of the BFGS.

denotes the set of weighted edges based on $\mathbf{A}$. In this work, a binary (0-1) adjacency matrix is assumed, where an edge weight of 1 indicates a connection between a residual and a fault, and 0 otherwise.¹

2. **One-Mode Projection onto Fault Nodes:** The projection onto $\mathcal{V}$ is defined such that two fault nodes v_i and v_j are connected if they share at least one common neighbor in $\mathcal{U}$. The weight of the projected edge between faults v_i and v_j is computed as the average of their connection weights to each shared residual:

$$a_{ij} = \sum_{u \in \mathcal{N}(v_i) \cap \mathcal{N}(v_j)} \frac{w_{v_i u} + w_{v_j u}}{2},$$

where $w_{v_i u}$ and $w_{v_j u}$ are the weights of edges from faults v_i and v_j to the shared residual node u , respectively.

3. **Projected Adjacency Matrix:** The resulting projected graph is represented by an adjacency matrix $\mathbf{A}_{\text{proj}} \in \mathbb{R}^{n_f \times n_f}$, where each entry (i, j) corresponds to the computed projected weight w_{ij} . An identity matrix $\mathbf{I}$ is added to $\mathbf{A}_{\text{proj}}$ to introduce self-loops: $\mathbf{A}_{\text{proj}} \leftarrow \mathbf{A}_{\text{proj}} + \mathbf{I}$, facilitating improved numerical stability during subsequent graph processing steps.
4. **Symmetric Normalization:** Finally, symmetric normalization is applied to the projected adjacency matrix to balance node degrees and enhance training stability

¹Alternatively, edge weights can be derived from sensitivity analyses of local neural networks trained for each fault node.

for graph-based learning algorithms. The normalized adjacency matrix is computed as: $\mathbf{A}_{\text{proj}} \leftarrow \mathbf{D}^{-1/2} \mathbf{A}_{\text{proj}} \mathbf{D}^{-1/2}$, where $\mathbf{D}$ is the diagonal degree matrix with entries $D_{ii} = \sum_j A_{\text{proj},ij}$, and $\mathbf{D}^{-1/2}$ denotes the elementwise inverse square root of $\mathbf{D}$.

This systematic projection process preserves the fault-residual relationships embedded in the BFSG while quantifying the degree of interaction between faults, which is critical for fault isolation task in the next steps.

In the proposed FIG, as shown in Figure 5.9, nodes are categorized into two types: *labeled nodes*, which have access to labeled faulty data, and *unlabeled nodes*, which only contain healthy data. For each labeled node, a supervised binary classifier has been trained, whereas each unlabeled node is equipped with an unsupervised autoencoder model.

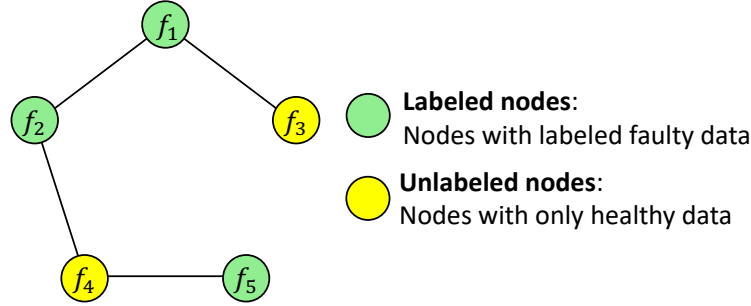


Figure 5.9: Fault Interaction Graph (FIG) with labeled and unlabeled nodes. Labeled nodes have labeled faulty data, while unlabeled nodes contain only healthy data.

Supervised Fault Probability Prediction for Labeled Node

When labeled faulty data are available for a node in the FIG, supervised learning is employed to detect occurrence of the corresponding fault. The binary model trained for each fault node is referred to as a *local detector*. The similar network architecture as Figure 5.2 is adopted here for binary fault classification, where the output label is 0 for healthy and 1 for fault occurrence. To reduce computational complexity in large-scale systems, only the residuals that are structurally sensitive to the fault—i.e., those identified in the fault signature matrix—are used as input features to the local detector.

It is important to note that some, or even all, of the residuals sensitive to a specific fault may also be sensitive to other faults. To address this, the binary classification task

for each node in the FIG is formulated to ensure accurate fault isolation. Specifically, for the fault associated with node i , the label 1 indicates the occurrence of fault i , while label 0 corresponds to either the healthy condition or the presence of faults associated with neighboring nodes $\mathcal{N}(i)$ in the FIG. This formulation ensures that the classifier is trained to distinguish fault i not only from nominal behavior but also from structurally related fault scenarios.

The binary classification model, depicted in Figure 5.10, is designed to detect faults from multivariate time-series data by learning both spatial and temporal patterns. The input to the model is a three-dimensional tensor $\mathbf{X} \in \mathbb{R}^{T \times D \times N}$, where T denotes the number of time steps, D the number of features, and N the number of data samples. The first layer of the network is a one-dimensional convolutional layer that applies F filters of size k to extract local spatial patterns across the feature dimensions.

Following the convolutional layer, a max pooling operation is applied to reduce the temporal resolution and improve computational efficiency. The resulting sequence of pooled activations, denoted by $\mathbf{p}_t$, is then processed by an RNN layer, which captures temporal dependencies across time steps; see Section 5.1.2 for further details.

To prevent overfitting, a dropout layer is included after the RNN layer, randomly zeroing out a fraction of the activations during training. Finally, the last hidden state $\mathbf{h}_T$ is passed to a dense (fully connected) output layer with a sigmoid activation function to compute the predicted fault probability:

$$p_i = \sigma(\mathbf{w}^\top \mathbf{h}_T + b),$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$, and $p_i \in [0, 1]$ quantifies the likelihood that fault i has occurred. A value of p_i close to 1 indicates a high probability of fault presence at node i , while a value near 0 suggests either healthy operation or the presence of a neighboring fault. Finally, the model is trained using the Adam optimizer and binary cross-entropy loss.

Feature Reduction via Sensitivity Analysis. To enhance the model’s efficiency and interpretability, a gradient-based sensitivity analysis was conducted to identify and retain only the most influential input features. This analysis evaluates how sensitive the network’s output is with respect to small perturbations in each input feature, thereby quantifying each feature’s relative importance.

After training the fault isolation network, the gradient of the predicted output with respect to each input feature was computed using TensorFlow’s automatic differentiation. Specifically, for a batch of validation inputs $\mathbf{X} \in \mathbb{R}^{T \times D \times B}$, where T is the time window length, D is the number of input features, and B is the batch size, the gradient of the

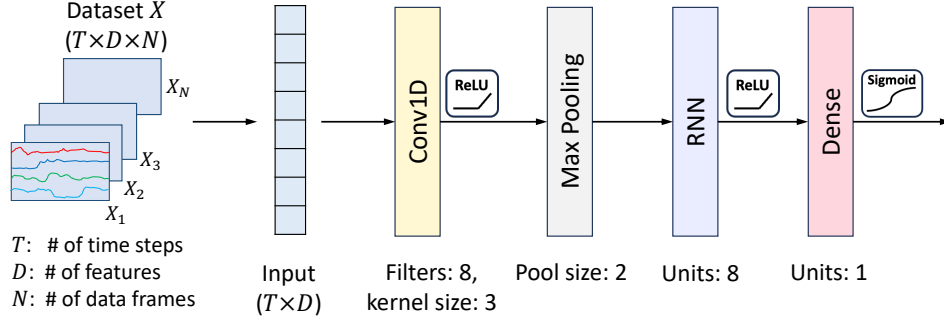


Figure 5.10: Architecture of the supervised binary classifier at node i .

model output $\hat{y}$ is calculated as $\nabla_{\mathbf{x}} \hat{y}$. The absolute value of these gradients is averaged across all samples and time steps to yield a global sensitivity score for each feature:

$$S_f = \frac{1}{B \cdot T} \sum_{b=1}^B \sum_{t=1}^T \left| \frac{\partial \hat{y}_b}{\partial X_{b,t,f}} \right|. \quad (5.9)$$

These scores are normalized so their sum equals one, and features are ranked accordingly. Rather than selecting a fixed subset based on a cumulative sensitivity threshold (e.g., 0.9), an iterative feature reduction strategy is employed. Starting with the full set of features, the least sensitive feature is removed in each iteration, and the model is re-evaluated to monitor classification performance. This process continues until a noticeable drop in accuracy is observed, at which point the last removed feature is restored and the feature reduction process is halted. This ensures that the final feature set maintains the original classification performance while achieving dimensionality reduction.

It was observed through simulations that some features with low sensitivity scores may still contribute significantly to classification accuracy. In some cases, removing a low-sensitivity feature led to a larger accuracy drop than removing a feature with a higher sensitivity score. This highlights a limitation of relying solely on gradient-based sensitivity. Moreover, in overparameterized networks, sensitivity scores can be misleading. To address these issues, the gradient-based sensitivity serves as a guide for pruning, while iterative validation ensures performance is preserved.

Eventually, feature dimensionality reduction makes the classifier more computationally efficient and interpretable, facilitating real-time deployment. Furthermore, the selected features will later be used to refine the fault signature matrix in Table 3.3, making it sparser and more effective than the one derived solely from structural analysis.

Unsupervised Anomaly Scoring for Unlabeled Nodes

For faults without labeled faulty data, unsupervised autoencoders are employed. The nodes associated with these faults are referred to as *unlabeled nodes* in Figure 5.9. For each unlabeled node, the corresponding autoencoder is trained using only healthy data and faulty data from neighboring labeled nodes, without exposure to faults occurring at the node itself. Consequently, if a fault occurs in the node, thresholding the reconstruction error enables its detection. To achieve this, an RNN-based sequence-to-sequence autoencoder is trained to learn a compressed temporal representation of the healthy behavior, making it well-suited for anomaly or fault detection based on reconstruction error.

The autoencoder architecture, illustrated in Fig. 5.11, is designed to classify input time-series data as either healthy or faulty based on reconstruction errors. The input consists of windowed time-series segments with dimensionality $T \times D$, where T is the number of time steps and D is the number of features. The encoder maps the input into a lower-dimensional latent space using two cascaded RNN layers with 32 and 16 units, respectively. A **RepeatVector** layer is then employed to match the temporal dimension required by the decoder. The decoder reconstructs the input using two cascaded RNN layers followed by a **TimeDistributed Dense** layer, ensuring the output has the same dimensionality $T \times D$ as the input. During inference, the reconstruction error is computed, and a thresholding method is applied to determine whether the input data is healthy (in-distribution) or faulty (out-of-distribution).

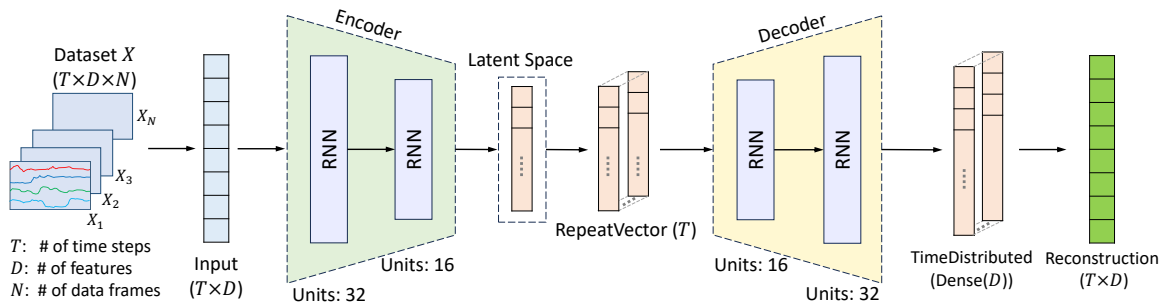


Figure 5.11: Architecture of the unsupervised autoencoder for binary fault classification at node i .

For each unlabeled node, the anomaly score is computed by measuring the discrepancy between the original and reconstructed input sequences, followed by a non-linear normalization to bound the score within $[0, 1]$.

Given an input sequence $\mathbf{X} \in \mathbb{R}^{T \times D}$, the corresponding reconstructed output from the trained autoencoder is denoted by $\hat{\mathbf{X}}$. The reconstruction error ϵ is computed as the mean absolute error over all features and time steps:

$$\epsilon = \frac{1}{T \cdot D} \sum_{i=1}^T \sum_{j=1}^D |\mathbf{X}_{ij} - \hat{\mathbf{X}}_{ij}|. \quad (5.10)$$

To obtain a normalized anomaly score $s \in [0, 1]$, the reconstruction error is transformed using a sigmoid-based function parameterized by a node-specific threshold τ (empirically selected from healthy validation data) and a scaling factor α :

$$s = \frac{1}{1 + \exp\left(-\alpha \frac{\epsilon - \tau}{\tau}\right)}. \quad (5.11)$$

The thresholds τ are precomputed and fixed for each unlabeled node based on a desired quantile (e.g., the 90th percentile) of the reconstruction error distribution obtained from healthy validation data. The resulting anomaly scores s_k serve as soft indicators of potential fault occurrences and are subsequently used in downstream decision-making processes such as consensus-based fault localization.

Node Score Consensus for Global Fault Isolation

In the proposed Fault Interaction Graph (FIG), shown in Figure 5.9, nodes are categorized into two types: *labeled nodes*, which have access to labeled faulty data, and *unlabeled nodes*, which only contain healthy data. For each labeled node, a supervised binary classifier has been trained as described in 5.2.5, whereas each unlabeled node is equipped with an unsupervised autoencoder model as detailed in 5.2.5.

Each node $i \in \mathcal{V}$ outputs a local decision score in the range $[0, 1]$. Specifically, a labeled node outputs a *predicted fault probability* $p_i \in [0, 1]$, while an unlabeled node yields an *anomaly score* $s_i \in [0, 1]$. To enable holistic decision-making across the graph, a *consensus-based score aggregation* mechanism is introduced that aggregates local scores through graph-based interactions.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}')$ denote the FIG, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}'$ defines their connections. The binary adjacency matrix $A \in \mathbb{R}^{N \times N}$ encodes the neighborhood structure of the graph. For each node $i \in \mathcal{V}$, a scalar node score $\rho_i \in [0, 1]$ is defined such that:

$$\rho_i = \begin{cases} p_i, & \text{if node } i \text{ is labeled,} \\ s_i, & \text{if node } i \text{ is unlabeled.} \end{cases} \quad (5.12)$$

To propagate these scores across the graph, the *consensus score*, ρ_i^{con} , for each node i is defined as:

$$\rho_i^{\text{con}} = \frac{1}{|\mathcal{N}_i| + 1} \left(\rho_i + \sum_{j \in \mathcal{N}_i} \phi_i(\rho_j) \right), \quad (5.13)$$

where $\mathcal{N}_i = \{j \in \mathcal{V} \mid A_{ij} \neq 0, j \neq i\}$ is the set of neighboring nodes, and the transformation function $\phi_i(\cdot)$ is defined based on the type of node i :

$$\phi_i(\rho_j) = \begin{cases} 1 - \rho_j, & \text{if node } i \text{ is labeled,} \\ \rho_j, & \text{if node } i \text{ is unlabeled.} \end{cases} \quad (5.14)$$

The transformation function $\phi_i(\cdot)$ captures the direction of belief propagation based on the node type and the way they are trained; see Figure 5.12.

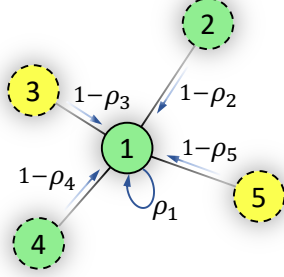
For a *labeled node*, which has been trained using a supervised binary classifier on its own fault data (labeled as 1) and on healthy as well as neighbor-fault data (labeled as 0), the output $\rho_i = p_i$ reflects the predicted probability of fault at that node. In this case, using the *complement* of each neighbor's score, $1 - \rho_j$, is meaningful: if neighboring nodes yield low scores (indicating they are likely healthy), this reinforces the hypothesis that the fault is likely centered at node i . Conversely, if the neighbors exhibit high scores, it weakens the confidence that node i is faulty, since neighboring nodes may themselves be candidates for fault occurrence.

For an *unlabeled node*, which is trained in an unsupervised manner using only healthy data and neighbor-fault data, the output $\rho_i = s_i$ reflects the anomaly score. In this case, it is reasonable to propagate the neighbor scores *as-is*, since these nodes rely on indirect information. When neighbors exhibit high scores, it may indicate that the anomaly source lies with node i , especially if fault labels are unavailable. Thus, direct aggregation of neighboring scores provides informative consensus that supports fault inference despite the absence of explicit supervision. This aggregation strategy allows the framework to meaningfully interpret and combine the scores of supervised and unsupervised nodes under a unified score aggregation mechanism.

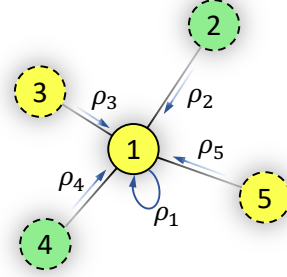
The final decision is made by selecting the node with the highest consensus score:

$$i^* = \arg \max_{i \in \mathcal{V}} \rho_i^{\text{con}}, \quad (5.15)$$

and comparing the score $\rho_{i^*}^{\text{con}}$ against a predefined threshold $\bar{\tau}$. The global decision label y is then defined as:



(a) Aggregation of node scores for labeled node 1.



(b) Aggregation of node scores for unlabeled node 1.

Figure 5.12: Consensus-based node score aggregation for global fault isolation.

$$y = \begin{cases} i^*, & \text{if } \rho_{i^*}^{\text{con}} > \bar{\tau}, \\ 0, & \text{otherwise} \end{cases} \quad (5.16)$$

where $y = 0$ indicates no fault (healthy condition), and $y = i^*$ indicates a fault at node i .

This consensus-based strategy enables a unified decision process across heterogeneous models by leveraging both local prediction confidence and graph-level dependencies.

5.2.6 Hybrid Fault Isolation via Graph Convolutional Network

Graph Convolutional Networks (GCNs), an extension of (CNNs), are designed to operate on graph-structured data by directly incorporating relational information through convolution over nodes and their neighbors. This ability to integrate both node features and graph topology makes GCNs particularly well-suited for tasks such as node classification and graph classification link prediction where relationships play a crucial role.

The convolution operation is defined in a spatial context. Let $\mathbf{X} \in \mathbb{R}^{N \times F}$ denote the input feature matrix (with N nodes and F features per node) and $\mathbf{A} \in \mathbb{R}^{N \times N}$ be the corresponding adjacency matrix of the graph. To ensure that a node's own features are included during the convolution, an augmented adjacency matrix is defined $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$, where $\mathbf{I}_N$ is the identity matrix. The corresponding degree matrix $\tilde{\mathbf{D}}$ is defined such that $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. The propagation rule for a single GCN layer is given by

$$\mathbf{H}^{(l+1)} = \sigma \left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(l)} \mathbf{W}^{(l)} \right),$$

where $\mathbf{H}^{(l)}$ denotes the node feature matrix at layer l (with $\mathbf{H}^{(0)} = \mathbf{X}$), $\mathbf{W}^{(l)}$ is the layer-specific trainable weight matrix, and σ is a non-linear activation function such as ReLU. This normalized formulation ensures that the features aggregated from neighboring nodes are appropriately scaled, which helps stabilize the training process and mitigate issues related to varying node degrees.

To incorporate both spatial and temporal information for fault localization, a GCN-RNN network is developed to process time-series node scores in two stages: spatial reasoning via GCN and temporal modeling via a RNN. As illustrated in Figure 5.13, the input to this network is a sequence of scalar node scores over T time steps, where each snapshot represents the node features for all N fault nodes at a given time. These node features correspond to the predicted fault probabilities p_i , generated by the associated local classifiers for each node, as defined in 5.2.5.

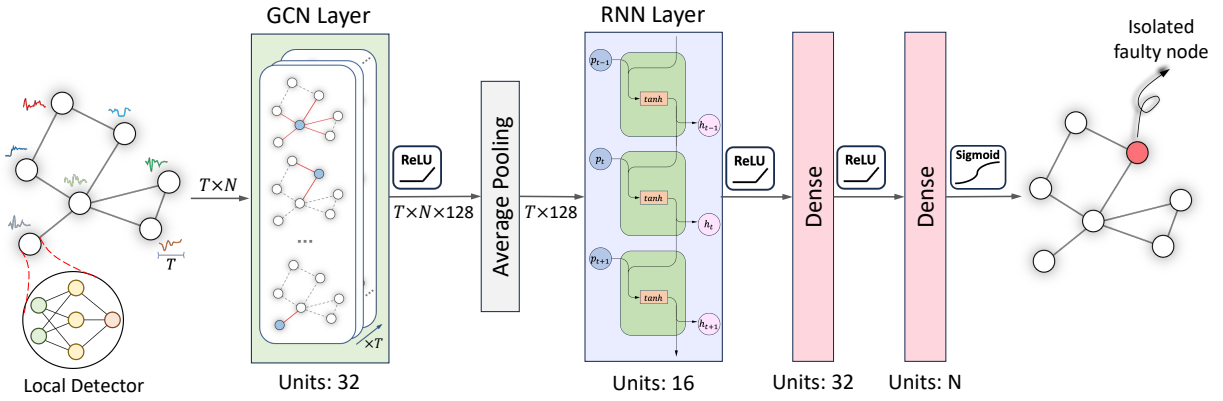


Figure 5.13: Architecture of the GCN-based model for global fault localization. Each graph node corresponds to a local detector, as illustrated in Figure 5.10.

The spatial reasoning is handled by applying a shared GCN to each snapshot independently using a `TimeDistributed` wrapper. This allows the GCN to extract topologically-aware node representations for each time step while preserving the temporal order of the data. Each of the T input frames—of shape $N \times 1$ —is passed through the GCN layer to produce a transformed sequence of node embeddings of shape $T \times N'$, where N' is the GCN output dimension (e.g., 128). These embeddings encode the structural relationships among faults at each time instance.

Following spatial encoding, an recurrent neural network (RNN) is applied across time to capture the temporal evolution of the spatially-informed node states. The RNN outputs a global temporal representation, which is then passed through a fully connected layer for

further feature compression. To prevent overfitting and improve generalization, regularization techniques such as dropout and weight decay are incorporated during training. This framework enables our model to effectively leverage the rich inter-dependencies in the data, ultimately leading to improved performance on the target application. Finally, through a sigmoid-activated output layer that performs multi-label classification across all N fault nodes. Each output value indicates the likelihood that the corresponding fault is active. However, this global supervised fault classification approach assumes that labeled data are available for all fault nodes during training, which limits its applicability in partially supervised or data-scarce scenarios.

This spatial-temporal modeling pipeline enables the network to reason over both the graph structure and the temporal progression of node-level fault indicators, making it particularly effective for global fault localization in distributed systems with evolving dynamics.

5.3 Fault Identification

This section presents the model-based formulation for identifying fault magnitude.

Let $\mathbf{f}_{b,k}$ represent deterministic, possibly time-varying measurement faults, and $\boldsymbol{\eta}_k$ denote zero-mean white noise. The expectation of residual at node i , defined in (5.6), is given by:

$$\mathbb{E}[\mathbf{r}_k^i] = \sum_{u \in \mathcal{I}} \mathbf{H}_k^{iu} \mathbb{E}[\tilde{\mathbf{x}}_k^{u-}] + \boldsymbol{\Lambda}_k^i \mathbf{f}_{b,k}^i. \quad (5.17)$$

Assuming that the *a priori* estimation error $\tilde{\mathbf{x}}_k^{i-}$ at node i is solely caused by a measurement fault $\mathbf{f}_{b,k}^j$ and a process fault $\mathbf{f}_{d,k}^j$ at an arbitrary node j , the expected error can be factorized as

$$\mathbb{E}[\tilde{\mathbf{x}}_k^{i-}] = \mathbf{M}_k^{ij} \mathbf{f}_{b,k}^j + \mathbf{L}_k^{ij} \mathbf{f}_{d,k}^j, \quad (5.18)$$

where the recursive time-update equations for the influence matrices $\mathbf{M}_k^{ij}$ and $\mathbf{L}_k^{ij}$ are derived in Subsection 5.3.2 and given by:

$$\begin{aligned} \mathbf{M}_{k+1}^{ij} &= \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \left[\mathbf{M}_k^{uj} - \mathbf{K}_k^u \sum_{v \in \mathcal{U}} \mathbf{H}_k^{uv} \mathbf{M}_k^{vj} - \mathbf{K}_k^u \boldsymbol{\Lambda}_k^u \boldsymbol{\Phi}^{uj} \right] \\ \mathbf{L}_{k+1}^{ij} &= \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \left[\mathbf{L}_k^{uj} - \mathbf{K}_k^u \sum_{v \in \mathcal{U}} \mathbf{H}_k^{uv} \mathbf{L}_k^{vj} \right] + \Upsilon_k^i \boldsymbol{\Psi}^{ij} \end{aligned} \quad (5.19)$$

A zero-one shape matrix Φ^{ij} is introduced such that $\mathbf{f}_{b,k}^i = \Phi^{ij} \mathbf{f}_{b,k}^j$, indicating common measurement faults in nodes i and j , and also a zero-one shape matrix Ψ^{ij} is introduced such that $\mathbf{f}_{d,k}^i = \Psi^{ij} \mathbf{f}_{d,k}^j$, indicating common process faults in nodes i and j . By substituting $\mathbb{E}[\tilde{\mathbf{x}}_k^{i-}]$ from (5.18) and using $\mathbf{f}_{b,k}^i = \Phi^{ij} \mathbf{f}_{b,k}^j$ in (5.17), the following is obtained:

$$\mathbb{E}[\mathbf{r}_k^i] = \left\{ \sum_{u \in \mathcal{I}} \mathbf{H}_k^{iu} \mathbf{M}_k^{uj} + \Lambda_k^i \Phi^{ij} \right\} \mathbf{f}_{b,k}^j + \left\{ \sum_{u \in \mathcal{I}} \mathbf{H}_k^{iu} \mathbf{L}_k^{uj} \right\} \mathbf{f}_{d,k}^j \quad (5.20)$$

In fact, this equation quantifies the impacts of measurement fault $\mathbf{f}_{b,k}^j$ and process fault $\mathbf{f}_{d,k}^j$ at node j on the residual vector of node i . For implementation, $\mathbb{E}[\mathbf{r}_k^i]$ is approximated by averaging the residuals over a time window of length T , expressed as

$$\mathbb{E}[\mathbf{r}_k^i] \approx \bar{\mathbf{r}}_k^i = \frac{1}{T} \sum_{t=k-T+1}^k \mathbf{r}_t^i$$

Given the detectability condition, the recursive least squares (RLS) method is used to estimate $\mathbf{f}_{b,k}^j$ and $\mathbf{f}_{d,k}^j$ from (5.20):

$$\begin{aligned} \hat{\mathbf{f}}_k &= \hat{\mathbf{f}}_{k-1} + \mathbf{K}_k \left(\bar{\mathbf{r}}_k^i - \mathbf{G}_k^{ij} \hat{\mathbf{f}}_{k-1} \right) \\ \mathbf{K}_k &= \mathbf{V}_{k-1} \mathbf{G}_k^{ij} \left(\lambda \mathbf{I} + \mathbf{G}_k^{ij\top} \mathbf{V}_{k-1} \mathbf{G}_k^{ij} \right)^{-1} \\ \mathbf{V}_k &= \frac{1}{\lambda} \left(\mathbf{V}_{k-1} - \mathbf{K}_k \mathbf{G}_k^{ij\top} \mathbf{V}_{k-1} \right) \end{aligned} \quad (5.21)$$

where the fault vector $\mathbf{f}_k$ and matrix $\mathbf{G}_k^{ij}$ are defined as

$$\mathbf{f} = [\mathbf{f}_b^j \quad \mathbf{f}_d^j]^\top, \quad \mathbf{G}_k^{ij} = [\sum_{u \in \mathcal{I}} \mathbf{H}_k^{iu} \mathbf{M}_k^{uj} + \Lambda_k^i \Phi^{ij} \quad \sum_{u \in \mathcal{I}} \mathbf{H}_k^{iu} \mathbf{L}_k^{uj}]. \quad (5.22)$$

The initial error covariance matrix $\mathbf{V}_0$ determines the level of confidence in the initial estimate and affects the convergence rate of the algorithm. The forgetting factor $\lambda \in [0, 1]$ controls the algorithm's responsiveness to changes in fault values.

5.3.1 Fault Identifiability

To enable the identification of faults in the fault vector $\mathbf{f}$ using residuals, the regression matrix used for fault estimation, denoted $\mathbf{G}_k^{ij}$, must have full column rank:

$$\text{rank}(\mathbf{G}_k^{ij}) = \dim(\mathbf{f}_k),$$

ensuring that the fault vector $\mathbf{f}_k$ is uniquely identifiable from the residual data. In the RLS update,

$$\mathbf{K}_k = \mathbf{V}_{k-1} \mathbf{G}_k^{ij} \left(\lambda \mathbf{I} + \mathbf{G}_k^{ij\top} \mathbf{V}_{k-1} \mathbf{G}_k^{ij} \right)^{-1},$$

the matrix $\mathbf{G}_k^{ij\top} \mathbf{V}_{k-1} \mathbf{G}_k^{ij}$ must be invertible for $\mathbf{K}_k$ to be computable. This invertibility requires that the Gram matrix be positive definite, which occurs only if $\mathbf{G}_k^{ij}$ has full column rank. Notably, this condition is inherently satisfied if the faults are isolable, as isolability implies that the effects of individual faults are distinguishable and linearly independent in the residual space.

In summary, for fault identification, the following conditions must be satisfied:

1. **Strong Detectability:** The fault or disturbance must induce a persistent and non-vanishing deviation from the nominal residual behavior. This is ensured if the fault is strongly detectable which guarantees that the expected residual remains above a certain threshold over time, i.e.,

$$\mathbb{E}[\mathbf{r}_k^i] \geq \epsilon,$$

for some $\epsilon > 0$. This condition ensures the fault's effect is sustained and observable for reliable estimation and isolation.

2. **Fault Isolability:** Each fault of the fault vector $\mathbf{f}$ must be *isolable*, meaning it induces a unique and distinguishable pattern in the residual vector such that it can be differentiated from other faults in the system.

5.3.2 Derivation of Influence Matrices

The recursive equations for the influence matrices $\mathbf{M}_k^{ij}$ and $\mathbf{L}_k^{ij}$ are derived as follows. Consider the expected *a priori* estimation error given in (4.4):

$$\mathbb{E}[\tilde{\mathbf{x}}_{k+1}^{i-}] = \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbb{E}[\tilde{\mathbf{x}}_k^{u+}] + \Upsilon_k^i \mathbf{f}_{d,k}^i \quad (5.23)$$

and the expected *a posteriori* estimation error given in (4.5):

$$\mathbb{E}[\tilde{\mathbf{x}}_k^{i+}] = \mathbb{E}[\tilde{\mathbf{x}}_k^{i-}] - \mathbf{K}_k^i \sum_{u \in \mathcal{I}} \mathbf{H}_k^{iu} \mathbb{E}[\tilde{\mathbf{x}}_k^{u-}] - \mathbf{K}_k^i \mathbf{\Lambda}_k^i \mathbf{f}_{b,k}^i \quad (5.24)$$

at node i . Substituting $\mathbb{E}[\tilde{\mathbf{x}}_k^{u+}]$ from (5.24) into (5.23) gives:

$$\begin{aligned} \mathbb{E}[\tilde{\mathbf{x}}_{k+1}^{i-}] &= \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbb{E}[\tilde{\mathbf{x}}_k^{u-}] - \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbf{K}_k^u \sum_{v \in \mathcal{U}} \mathbf{H}_k^{uv} \mathbb{E}[\tilde{\mathbf{x}}_k^{v-}] \\ &\quad - \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbf{K}_k^u \mathbf{\Lambda}_k^{u, b, k} \mathbf{f}_{b, k}^u + \mathbf{\Upsilon}_k^i \mathbf{f}_{d, k}^i \end{aligned} \quad (5.25)$$

where $\mathcal{U}$ is the in-neighbor sets of node u . Substituting the factorization $\mathbb{E}[\tilde{\mathbf{x}}_k^{i-}] = \mathbf{M}_k^{ij} \mathbf{f}_{b, k}^j + \mathbf{L}_k^{ij} \mathbf{f}_{d, k}^j$ into the above equation and rearranging all the terms yields:

$$\begin{aligned} &\left[\mathbf{M}_{k+1}^{ij} - \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbf{M}_k^{uj} + \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbf{K}_k^u \sum_{v \in \mathcal{U}} \mathbf{H}_k^{uv} \mathbf{M}_k^{vj} + \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbf{K}_k^u \mathbf{\Lambda}_k^{u, b, k} \mathbf{\Phi}^{uj} \right] \mathbf{f}_{b, k}^j + \\ &\left[\mathbf{L}_{k+1}^{ij} - \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbf{L}_k^{uj} + \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \mathbf{K}_k^u \sum_{v \in \mathcal{U}} \mathbf{H}_k^{uv} \mathbf{L}_k^{vj} - \mathbf{\Upsilon}_k^i \mathbf{\Psi}^{ij} \right] \mathbf{f}_{d, k}^j = \mathbf{0} \end{aligned} \quad (5.26)$$

Since the faults $\mathbf{f}_{b, k}^j$ and $\mathbf{f}_{d, k}^j$ are assumed to be non-zero, their corresponding coefficients in (5.26) must be zero, leading to:

$$\begin{aligned} \mathbf{M}_{k+1}^{ij} &= \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \left[\mathbf{M}_k^{uj} - \mathbf{K}_k^u \sum_{v \in \mathcal{U}} \mathbf{H}_k^{uv} \mathbf{M}_k^{vj} - \mathbf{K}_k^u \mathbf{\Lambda}_k^{u, b, k} \mathbf{\Phi}^{uj} \right] \\ \mathbf{L}_{k+1}^{ij} &= \sum_{u \in \mathcal{I}} \mathbf{F}_k^{iu} \left[\mathbf{L}_k^{uj} - \mathbf{K}_k^u \sum_{v \in \mathcal{U}} \mathbf{H}_k^{uv} \mathbf{L}_k^{vj} \right] + \mathbf{\Upsilon}_k^i \mathbf{\Psi}^{ij} \end{aligned} \quad (5.27)$$

Finally, the influence matrices are computed recursively, starting with $\mathbf{M}_0^{ij} = \mathbf{0}$ and $\mathbf{L}_0^{ij} = \mathbf{0}$.

5.4 Summary

In this chapter, a unified and scalable framework for fault detection, isolation, and identification is presented. System prior knowledge was leveraged as much as possible to enable accurate and efficient fault diagnosis. To achieve this, both quantitative and structural expert knowledge of the system were integrated into the data through the proposed hybrid methods.

For fault detection, an ensemble strategy was introduced by integrating the statistical robustness of the threshold-based χ^2 -test with the pattern recognition capability of a neural

network. A fault was declared only when both methods agreed which significantly reduced false alarms and improved detection accuracy.

For fault isolation, residual signals were combined with model-based likelihoods within a hybrid neural architecture to improve performance under uncertainty and data limitations. For effective isolation in large-scale systems, distributed learning-based methods—such as feature integration, consensus-based decision-making, and graph convolutional networks—were developed to ensure scalability and support partial supervision. These methods were designed to be modular, allowing incremental retraining when new fault sources are introduced.

Fault identification was addressed by quantifying fault magnitudes and analyzing their propagation across estimation paths. The influence of faults on state estimates was characterized using influence matrices, enabling direct fault compensation when magnitudes were identifiable.

Throughout the framework, the need for observer banks or exhaustive hypothesis testing was avoided by embedding prior system knowledge directly into the learning process. The proposed detection and isolation networks included a cascade of convolutional and recurrent layers to capture temporal and structural dependencies, where the recurrent layer demonstrated strong recall performance.

Chapter 6

Numerical and Experimental Validation

This chapter presents the validation of the proposed estimation graph and uncertainty quantification (UQ) methods. The estimation graph is validated using data from an experimental test vehicle. Both the analytical and surrogate-based UQ approaches are verified through Monte Carlo simulations, and their computational complexities are assessed using a high-fidelity vehicle model.

6.1 Validation of Vehicle State Estimation Graph

This section experimentally validates the estimators proposed in Chapter 3 under both normal and harsh driving conditions.

6.1.1 Experimental Test Setup

The test platform is an electric Chevrolet Equinox equipped with an all-wheel-drive system and independent electric motors on each wheel. The key vehicle specifications are summarized in Table 6.1. Sensor measurements are sampled at 200 Hz.

The hardware architecture of the experimental setup is illustrated in Figure 6.1. The test vehicle is equipped with an RT2500 6-axis GNSS/IMU, which provides high-accuracy ground truth measurements of the longitudinal velocity, as well as longitudinal and lateral

Table 6.1: Specifications of Chevrolet Equinox test vehicle.

Parameter	Description	Value	Unit
m	Vehicle total Mass	2271	kg
l_f	Distance from front axle to the CG	1.42	m
l_r	Distance from rear axle to the CG	1.43	m
b_w	Track Width (front and rear)	1.58	m
h_{CG}	Height of CG	0.75	m
I_{zz}	Vehicle moment of inertia	4600	kg.m ²
R_e	Tire effective radius	0.312	m
SR	Ratio of steering and front wheel angle	18:1	–
I_w	Wheel moment of inertia	1.7	kg.m ²
$C_{y,ij}$	Tire cornering stiffness	56050	N
(B_x, C_x, D_x, E_x)	Longitudinal Tire Model Coefficients	(9.0, 1.56, 0.86, 0.4)	–
(B_y, C_y, D_y, E_y)	Lateral Tire Model Coefficients	(7.6, 1.27, 0.58, 0.2)	–

accelerations and yaw rate for the estimator. For validation, the RT2500 system is complemented by a Tire Force/Moment Measurement module that directly measures tire forces and moments at each wheel. Standard wheel encoders and a steering angle encoder are also integrated to provide wheel speed and steering measurements. All sensors and modules communicate over a CAN bus, and the signals are interfaced with MATLAB/SIMULINK through a dSPACE MicroAutoBox II. The estimators and algorithms are developed in MATLAB/SIMULINK and tested using the collected experimental data.

For validation, the estimators are tested on three specific driving maneuvers designed to evaluate the estimation of vehicle velocity, roll angle, tire forces, and road slope angles. The details of these test scenarios are summarized in Table 6.2. Tests 2 and 3 are conducted under challenging road and driving conditions, which pose significant difficulties for accurately estimating tire forces at each time step. In particular, Test 3 involves a harsh maneuver on an icy road where all four wheels experience substantial combined slip, making it especially difficult for the models to estimate both velocity and tire forces reliably.

Table 6.2: Maneuvers and road conditions in the experimental test dataset.

Dataset	Description	Road Condition	Friction Coefficient
Test 1	Acceleration in turn	Dry	0.9
Test 2	Driving forward (high slip)	Wet	0.6
Test 3	Acceleration in turn	Icy	0.2

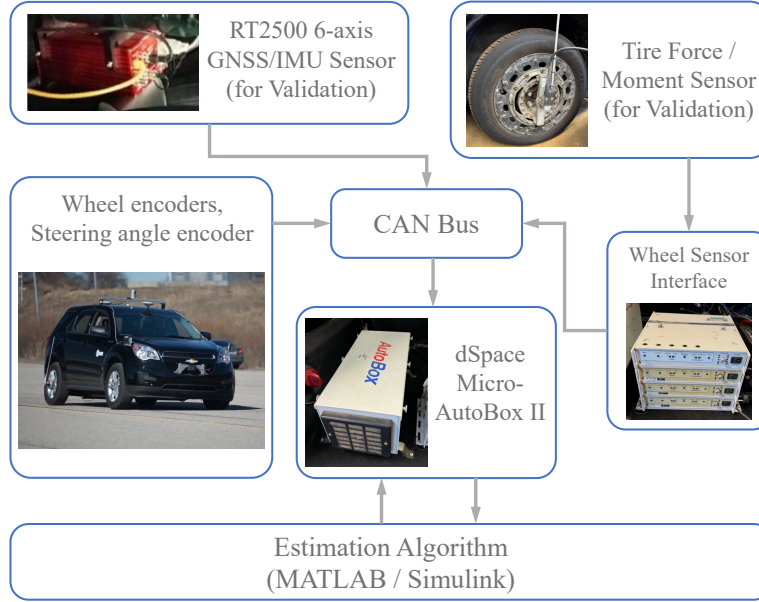


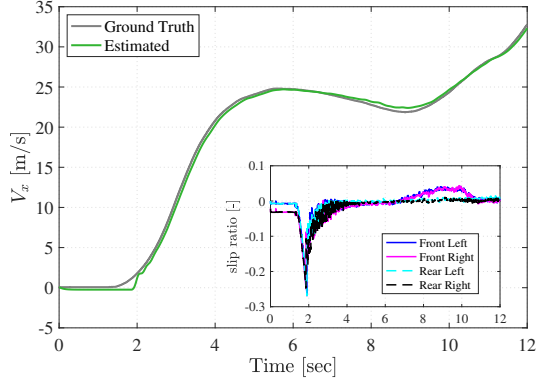
Figure 6.1: Hardware layout of the experimental setup.

6.1.2 Velocity and Longitudinal Tire Force Estimation (Test 1)

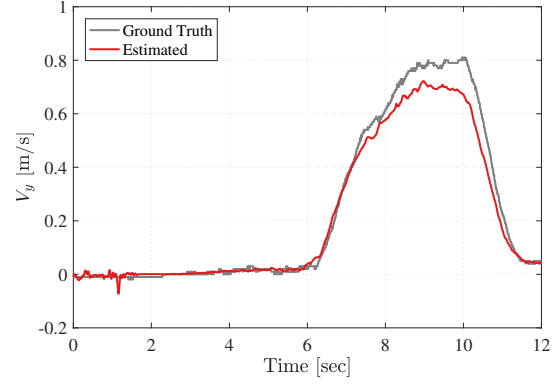
The estimation path $\mathcal{P}_3$ (including estimators E_1 , E_2 , and E_3) is implemented to estimate the longitudinal velocity and the longitudinal tire forces of the vehicle using tire model without wheel torques in the normal driving scenario of Test 1. The estimated longitudinal velocity from E_3 is fed into the estimator E_8 to estimated the lateral velocity.

Figure 6.2a illustrates the longitudinal velocity estimation along with the corresponding wheel slips for all four wheels under normal road and driving condition. The Kalman-based observer successfully tracks the actual longitudinal velocity. Figure 6.2b shows the estimated lateral velocity, which aligns closely with the actual values, with a maximum error of 10% relative to the peak lateral velocity.

The longitudinal tire forces estimated by estimator E_3 , based on the tire model, are compared with the actual forces measured by wheel force sensors, as shown in Figure 6.3. It is important to note that wheel torque measurements are not utilized in this estimator for force estimation.



(a) Longitudinal velocity estimated by E_3 .



(b) Lateral velocity estimated by E_8 .

Figure 6.2: Estimated longitudinal and lateral velocities during Test 1.

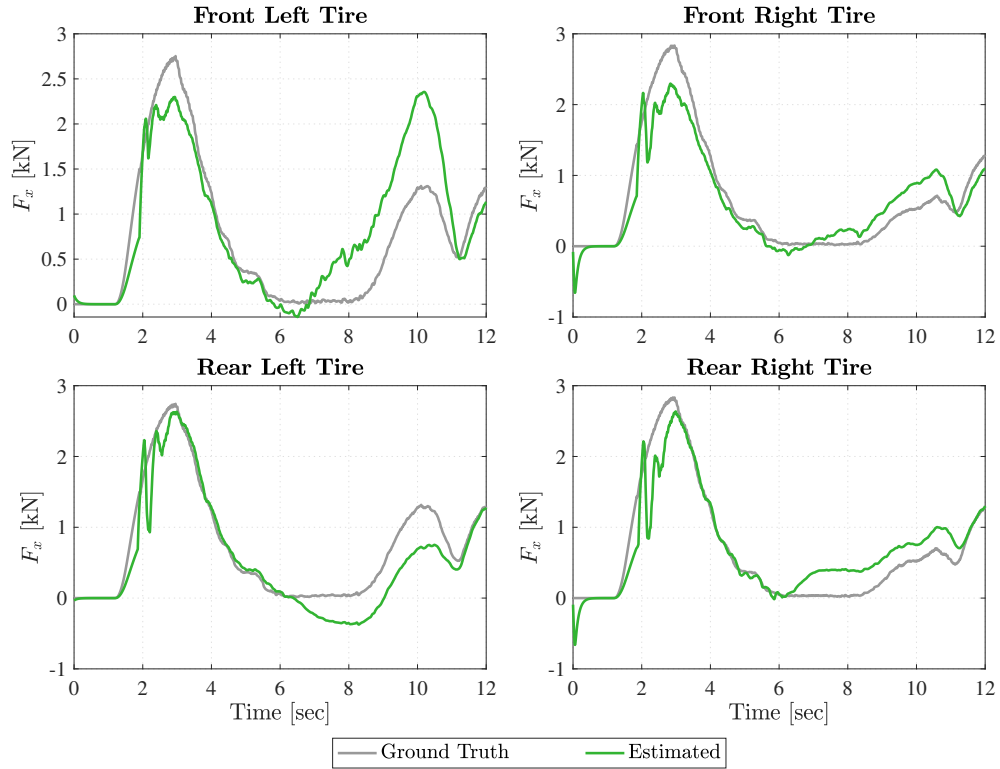


Figure 6.3: Longitudinal tire force estimation by E_3 using the tire model in Test 1.

6.1.3 Longitudinal Velocity Estimation (Test 2)

Test 2 is conducted on a wet road surface with a medium friction coefficient, where two front wheels experience high slip conditions, as shown in Figure 6.4. The figure illustrates the longitudinal velocity estimated by the model-based observer E_7 , which combines the vehicle's longitudinal kinematics with wheel rotational dynamics. Despite the presence of partial wheel slip, the estimated longitudinal velocity closely matches the actual velocity, with a maximum error of 1.2 kph.

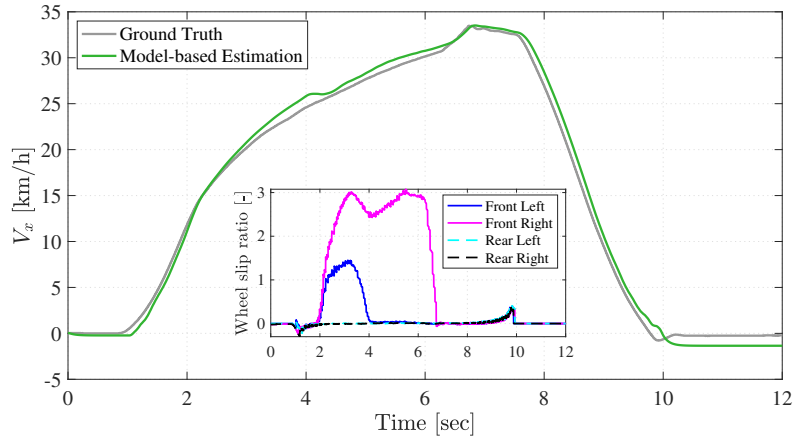


Figure 6.4: Longitudinal velocity estimation using the model-based estimator E_7 . The two front wheels are experiencing slip.

6.1.4 Velocity and Tire Force Estimation (Test 3)

Estimation path $\mathcal{P}_5$, which includes estimators E_1 , E_2 , E_4 , and E_5 , is implemented to estimate the vehicle velocity and tire forces. The estimation relies on wheel rotational dynamics and the lateral tire force model, and is evaluated under the combined-slip conditions of Test 3, where wheels experience longitudinal and significant lateral slip on an icy road.

Figure 6.5 presents the velocity estimation results during Test 3, which involves combined slip at all wheels. As shown in Figure 6.5a, the accuracy of longitudinal velocity estimation by estimator E_3 decreases after $t = 4$ s. This degradation is primarily due to the presence of lateral slip, which is not accounted for in the longitudinal velocity estimation model of

E_3 . In contrast, estimator E_5 , which incorporates a tire force model, accurately estimates the lateral velocity even under challenging road conditions, with a maximum error of only 12% relative to the peak lateral velocity, as illustrated in Figure 6.5b.

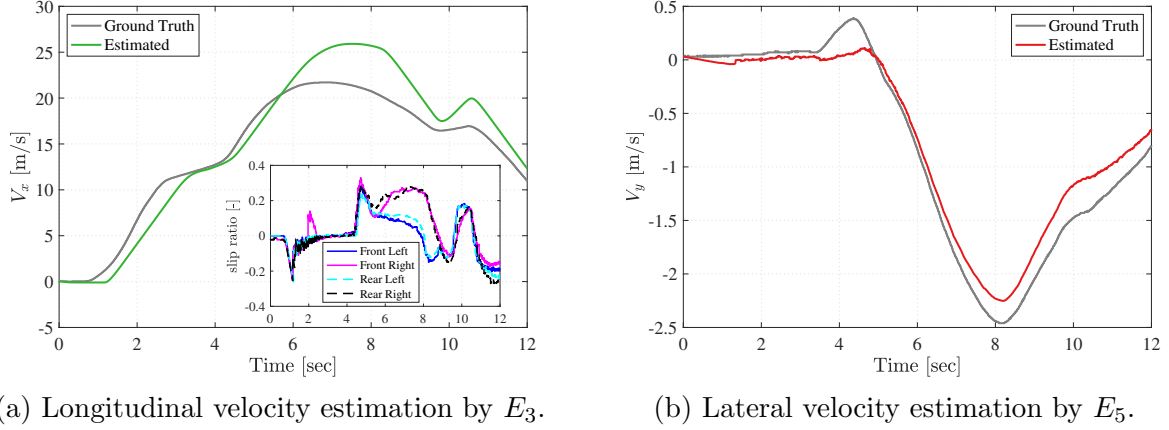


Figure 6.5: Velocity estimation results from the model-based observer on Test 3 with combined wheel slip.

Figure 6.6 presents the longitudinal tire force estimates obtained by estimators E_4 and E_6 during Test 3. Estimator E_4 uses wheel torque measurements and wheel rotational dynamics to estimate the individual longitudinal tire forces. This model-based approach does not rely on tire models and is effective under complex slip conditions—such as those on icy roads—provided that accurate wheel torque data is available. In contrast, estimator E_6 employs a kinematics-based approach, using a discrete-time Kalman filter to estimate the total longitudinal and lateral tire forces at the front and rear axles based on vehicle acceleration and yaw rate measurements. The process model assumes a random walk for the tire forces, and the measurement model incorporates vehicle kinematics and steering geometry. The estimated axle-level forces are then distributed among individual tires according to the normal load distribution formulas (A.6)–(A.7). As shown in the figure, both estimators track the ground truth tire forces with similar accuracy despite the presence of noise and high-slip conditions. Estimators E_4 and E_6 exhibit comparable estimation accuracy overall, with each slightly outperforming the other at different points during the maneuver. At certain times, E_4 provides more accurate estimates due to its use of wheel dynamics, while at other times, E_6 's kinematics-based approach yields better performance. Although both strategies are effective, the performance of E_6 may degrade in the presence of sensor faults, since it relies solely on kinematic and inertial measurements without torque

feedback and vehicle dynamics model.

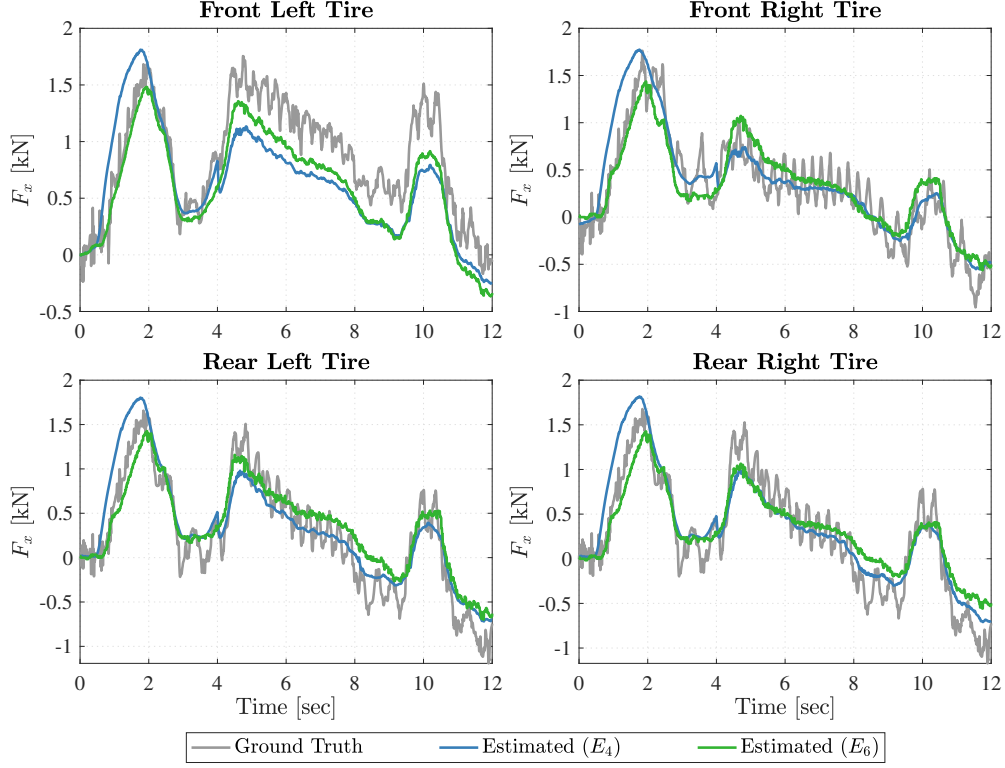


Figure 6.6: Lateral tire force estimation by E_4 (wheel rotational dynamics) and E_6 (kinematics-based) in Test 3.

Figure 6.7 shows the lateral tire force estimation results from estimators E_5 and E_6 during Test 3 on an icy road. Estimator E_5 , implemented along path $\mathcal{P}_5$, uses vertical forces from E_2 and longitudinal forces from E_4 as inputs to a nonlinear tire model. Despite the presence of combined slip and low friction, estimator E_5 tracks the actual forces reasonably well, with an average RMSE of 0.23 kN across all four tires, demonstrating the effectiveness of the cascaded model-based approach. In contrast, E_6 uses a kinematics-based method to estimate axle-level lateral forces, which are then allocated to individual tires based on normal load estimates. With an average RMSE of 0.18 kN, it offers slightly better accuracy than E_5 , while achieving similar overall performance. While E_5 captures transient responses more sharply, E_6 offers smoother estimates that better follow the general trend in some phases. These results demonstrate the redundancy of the two estimation approaches under

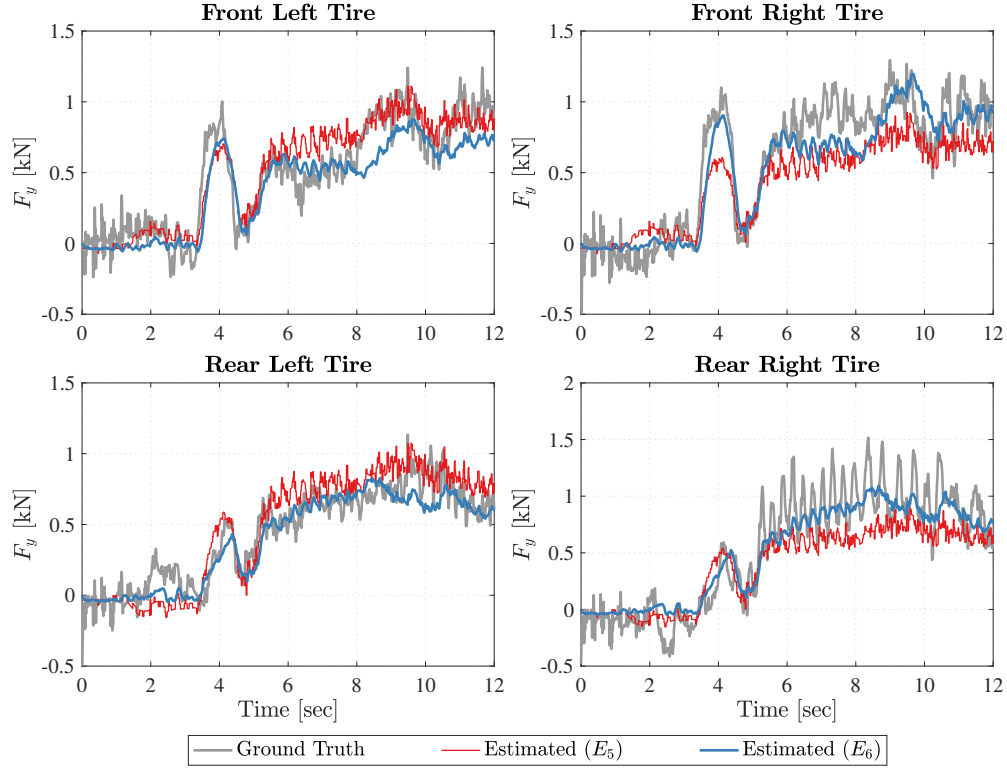


Figure 6.7: Lateral tire force estimation by E_5 (tire model) and E_6 (kinematics) in Test 3.

challenging road conditions.

6.1.5 Roll Angle Estimation Using Suspension Height Sensors

Figure 6.8 shows the estimated suspension heights at all four corners of the vehicle compared with the ground truth from the CarSim high-fidelity simulation. Each suspension height is reconstructed using the analytical redundancy relations in Section A.3.6, where the height at one corner is estimated from the remaining three based on the suspension geometry. Specifically, equation (A.11) is used to project the body plane—formed by three suspension points—onto the excluded sensor location. The results show excellent agreement with the ground truth, with an RMSE of approximately 1 mm, validating the accuracy and robustness of the redundancy-based estimation method under dynamic conditions.

Figure 6.9 presents the estimated roll angle using different combinations of suspension height sensors. Two estimations are shown: one excluding the front left sensor and the other

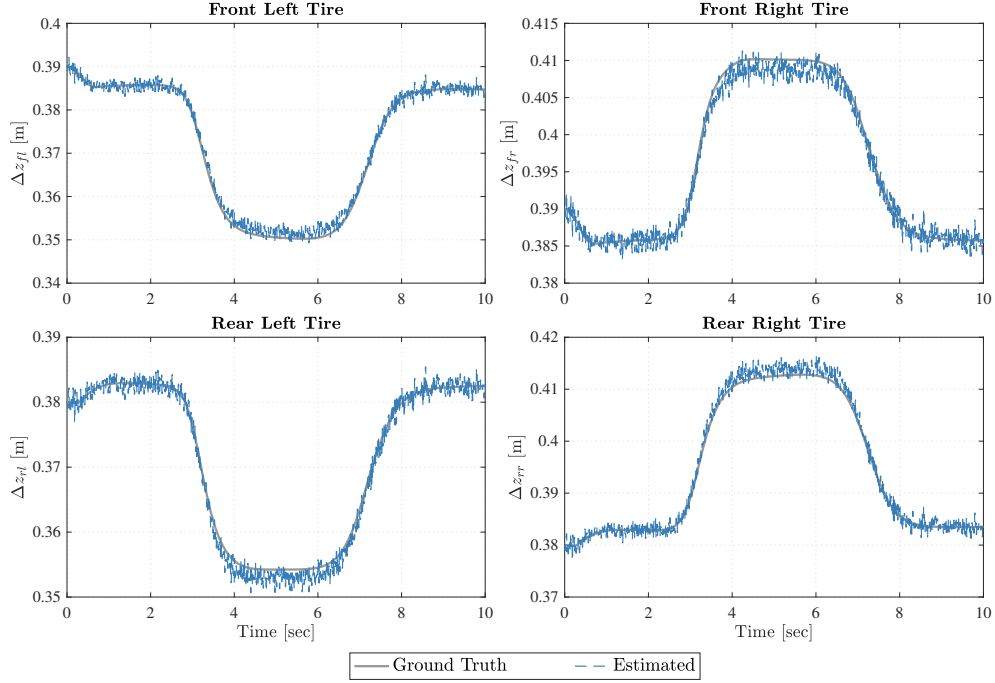


Figure 6.8: Estimated suspension heights.

excluding the rear left. The method computes the body plane normal using the remaining three sensors and estimates the roll angle via equation (A.12). Both configurations yield accurate results, with a maximum error of less than 2%, demonstrating the effectiveness of analytical redundancy. Similar accuracy is achieved when excluding any of the other two corners, allowing four redundant configurations for roll angle estimation.

6.1.6 Road Slope Angle Estimation

Figure 6.10 shows the estimated road slope angles—road grade and road bank—using unknown input observers E_9 and E_{10} , developed in Sections 3.2.9 and 3.2.10, respectively. The estimation is performed in a co-simulation environment using a high-fidelity CarSIM model of an SUV and a Simulink-based implementation of the observers. The vehicle model specifications are provided in Section 6.2.1. The results demonstrate that both observers track the ground truth road angles closely, despite the presence of dynamic variations and nonlinearities in the road profile.

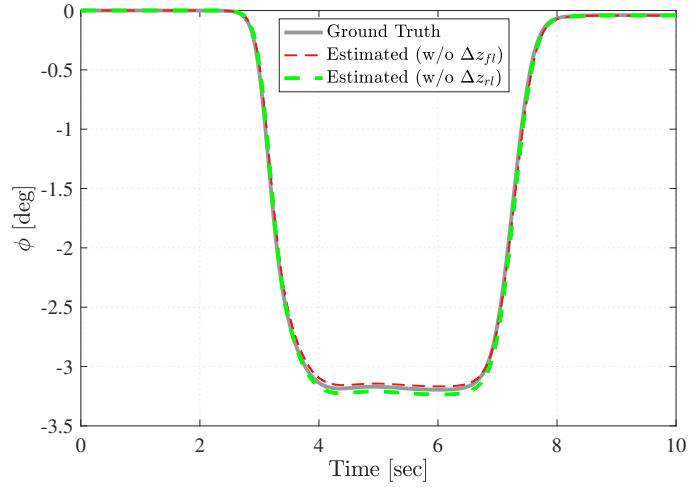


Figure 6.9: Estimated roll angle using different combinations of suspension height sensors.

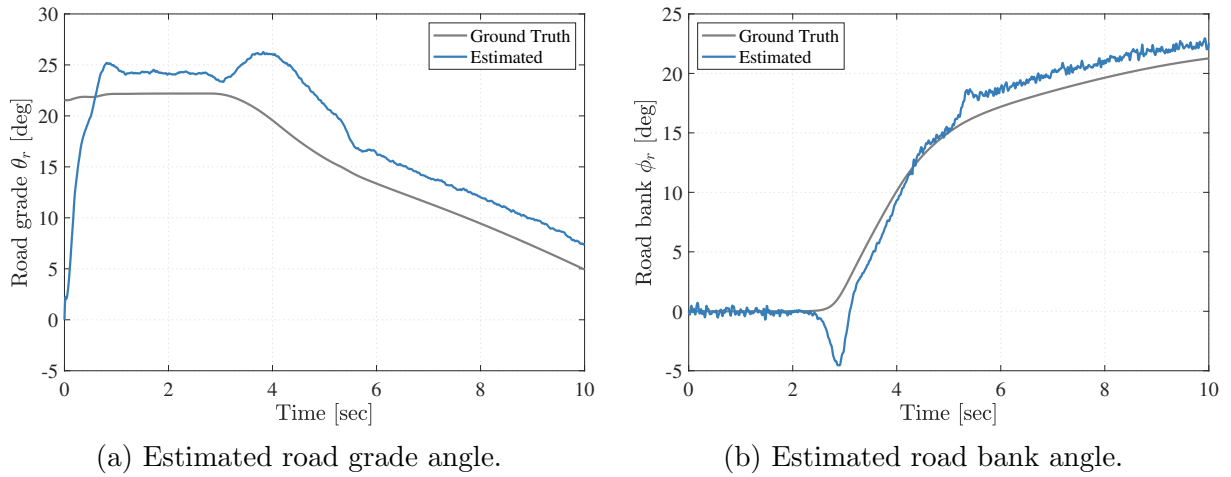


Figure 6.10: Road slope angle estimation using unknown input observer.

6.2 Validation of Uncertainty Quantification

6.2.1 Analytical Uncertainty Quantification

This section aims to numerically validate the analytical uncertainty quantification framework developed in Section 4.2 for Kalman-based estimation paths. Validation is performed along estimation path illustrated in Figure 6.12. The simulation setup consists of a high-fidelity CarSim vehicle dynamics model, along with corresponding sensor models, used to implement the estimators E_i and their associated uncertainty quantifiers UQ_i , as shown in Figure 6.14.

The vehicle model used in this study is a high-fidelity SUV from CarSim, illustrated in Figure 6.11. It is a front-axle drive model with detailed vehicle dynamics, such as suspension, steering, braking, and powertrain systems. The specification of vehicle model is listed in Table 6.3. This CarSim model is integrated with Simulink, where the real-time estimation and UQ algorithms are executed. This integrated platform creates a realistic environment for evaluating the performance of our proposed framework across various fault scenarios.



Figure 6.11: A front-axle-drive E-Class SUV model in CarSim, used as the test platform.

Figure 6.12 illustrates an estimation path for the vehicle's lateral dynamics, extracted from the full estimation graph shown in Figure 3.2. This path consists of three sequential local estimator nodes: E_1 estimates the load transfer (see Section 3.2.1), E_2 estimates the vertical tire loads (Section 3.2.2), and E_5 estimates the lateral tire forces and lateral velocity (Section 3.2.5). It is assumed that the longitudinal velocity is constant and the longitudinal tire forces are negligible; thus, the estimation of longitudinal tire forces (by E_3 or E_4) is excluded from this path.

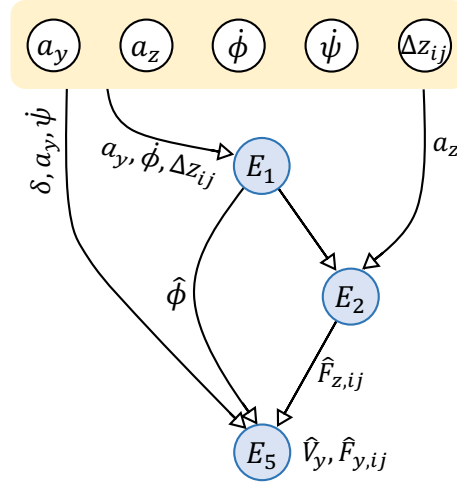


Figure 6.12: An example of a cascaded estimation path for vehicle lateral velocity and lateral tire forces. The path includes three estimator nodes: E_1 estimates roll angle and load transfer, E_2 estimates vertical loads, and E_5 estimates lateral velocity and tire forces. It is assumed that the longitudinal velocity remains constant and the longitudinal tire forces are negligible.

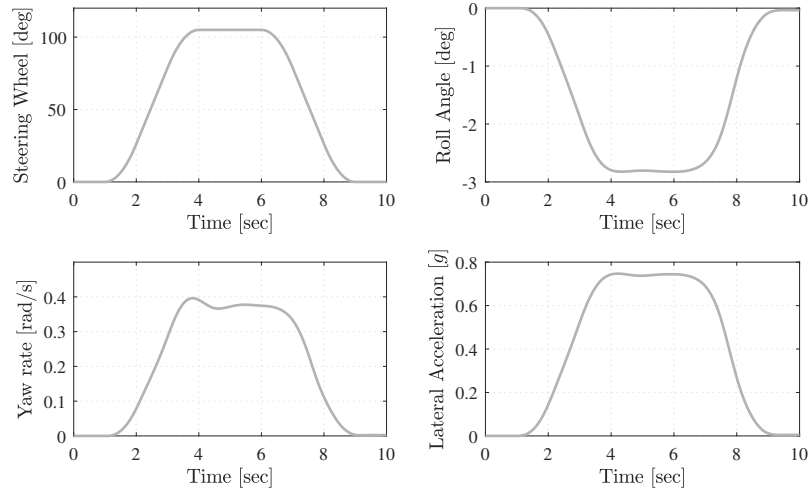


Figure 6.13: Cornering maneuver test at a constant speed of 70 km/h: Steering angle, roll angle, yaw rate, and lateral acceleration.

Table 6.3: Parameters of the Test Vehicle from CarSim

Parameter	Description	Value	Unit
m_s	Vehicle Sprung Mass	1990	kg
m_{us}	Vehicle Unsprung Mass	330	kg
I_{xx}	Roll Moment of Inertia	895	kg.m ²
I_{yy}	Pitch Moment of Inertia	2687	kg.m ²
I_{zz}	Yaw Moment of Inertia	3573	kg.m ²
l_f	Front Axle Distance from CG	1.24	m
l_r	Rear Axle Distance from CG	1.71	m
b_w	Track Width (front and rear)	1.58	m
h_{CG}	Center of Gravity Height	0.66	m
h_{RC}	Roll Center Height (front and rear)	0.43	m
k_ϕ	Roll Stiffness Coefficient	143910	N/rad
c_ϕ	Roll Damping Coefficient	41300	N.s/rad
σ	Tire Relaxation Length	0.5	m
C_y	Tire Cornering Stiffness Coefficient	11.75	—
(B_x, C_x, D_x, E_x)	Longitudinal Tire Model Coefficients	(9.0, 1.52, 0.83, 0.5)	—
(B_y, C_y, D_y, E_y)	Lateral Tire Model Coefficients	(8.9, 1.59, 0.83, 0.1)	—

As a representative case, the effect of a fault in the IMU sensor is investigated. The specifications of the IMU sensor under both healthy and faulty conditions are provided in Table 6.4. In the validation process, the propagation of two fault types in $a_{y,m}$ along the estimation path is quantified: a constant bias fault and an excessive noise fault; both are modeled as zero-mean Gaussian distributions with a variance of $0.5 \text{ m}^2/\text{s}^4$. To assess the performance of our approach, a cornering maneuver at a constant longitudinal speed of 70 km/h is conducted. The test is assumed on dry road with $\mu = 1$. Figure 6.13 depicts the time history of steer angle, roll angle, yaw rate, and lateral acceleration as the key variables for the lateral dynamics.

Figure 6.15a compares the analytically quantified uncertainties—computed using (4.17)–(4.20)—in lateral tire forces under a bias fault in $a_{y,m}$, against 1000 Monte Carlo (MC) simulations as actual values. The close alignment between the analytical $\pm 2\sigma$ uncertainty bands and the simulation results confirms the accuracy of the analytical method in capturing the propagation of estimation uncertainty due to the bias fault. Notably, the level of uncertainty varies across different states and operating conditions. For instance, during a cornering maneuver ($\delta \neq 0$), the uncertainty in the rear-right tire’s lateral force decreases, while it increases for the rear-left tire. This highlights the potential to select an optimal estimation path based on the state of interest to minimize uncertainty.

Table 6.4: IMU Sensor Specifications

Sensor	Condition	Parameter	Value
Accelerometer (3-axis)	Healthy	Noise Variance	10^{-3}
		Bias Variance	0
	Faulty	Noise Variance	0.5
		Bias Variance	0.5
Gyro (3-axis)	Healthy	Noise Variance	10^{-4}
		Bias Variance	0
	Faulty	Noise Variance	10^{-2}
		Bias Variance	10^{-2}

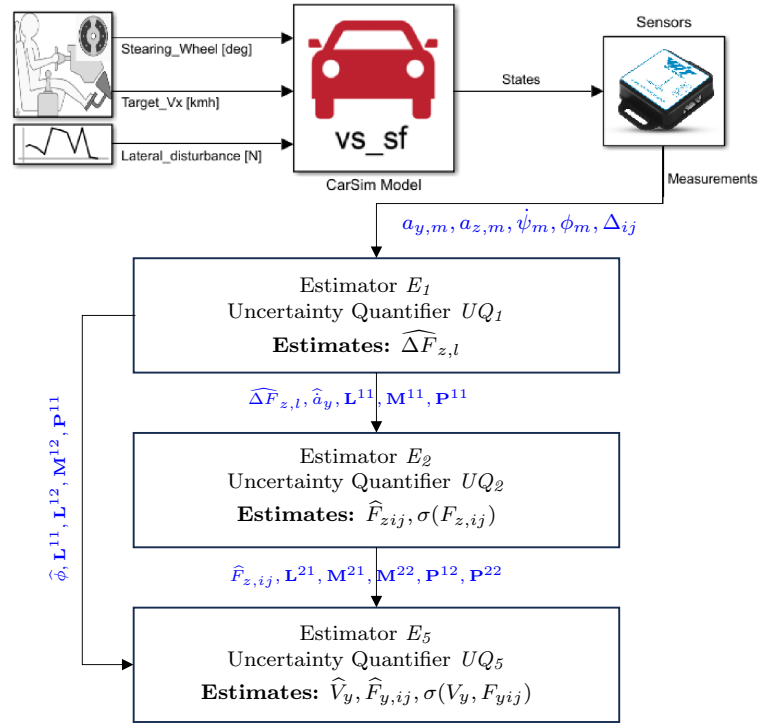


Figure 6.14: Uncertainty quantification framework for a cascaded estimation path, implemented in Simulink with CarSim integration.

Similarly, Figure 6.15b compares the analytically quantified uncertainties—computed using (4.17)–(4.20)—in lateral tire forces under an excessive noise fault in $a_{y,m}$, against 1000 Monte Carlo (MC) simulations. Once again, there is a close alignment between the analytical uncertainty bands and the simulation results. A key distinction between the uncertainty bands for bias and excessive noise faults, despite having the same fault variance, is that the estimation uncertainty in lateral forces is noticeably lower under the excessive noise fault than under the bias fault. Figure 6.16 further highlights this difference by comparing the uncertainty in lateral velocity estimation under both fault types in $a_{y,m}$, using the same fault magnitude. During the cornering maneuver ($\delta \neq 0$) in the time interval from 2 to 8 seconds, the estimation uncertainty increases significantly under the bias fault, while it remains relatively constant under the excessive noise fault.

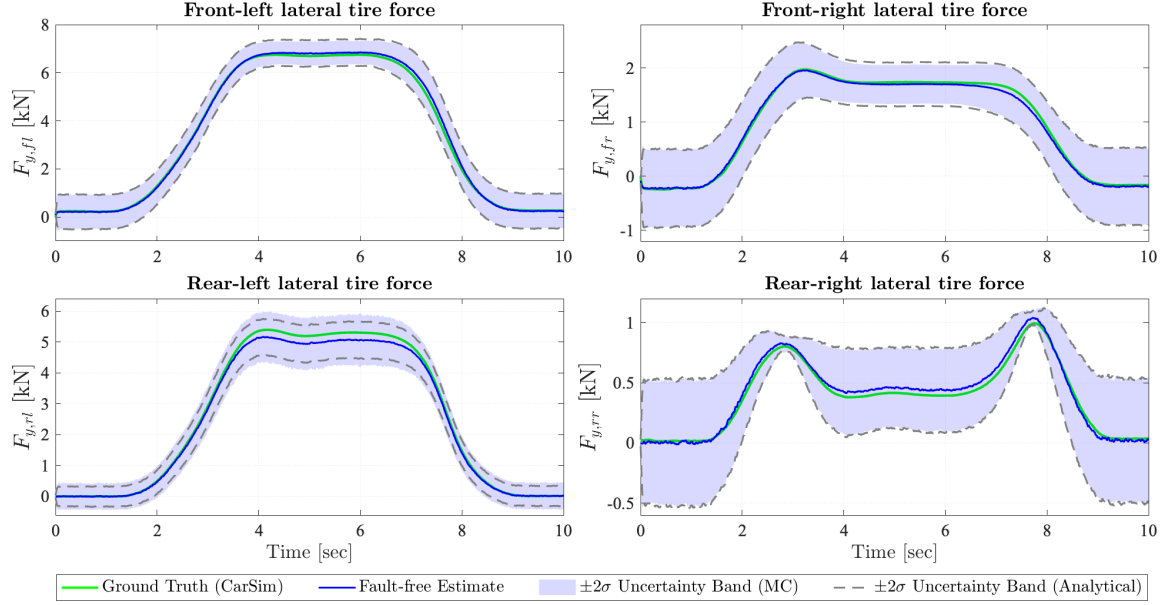
Table 6.5 compares the time-averaged analytical standard deviation (SD) and the root mean squared error (RMSE) from MC simulations over a 10-second interval for various fault sources and types. The analytical quantification of estimation uncertainty maintains the same level of accuracy for other fault sources ($a_{z,m}, \dot{\psi}_m, w_y$) and fault types (bias and excessive noise). The close match between the analytical quantified uncertainty and the MC results demonstrates the accuracy of the analytical approach. Minor discrepancies observed are likely due to unmodeled dynamics and linearization errors in the extended Kalman filter, which are not considered in the analytical fault propagation. Furthermore, the estimation uncertainty is clearly lower for noise faults compared to bias faults of the same magnitude.

In summary, it is demonstrated that the analytical derivations accurately quantify estimation uncertainty from various fault sources along an estimation path. Therefore, this approach can be applied in real-time to identify the most reliable estimation path under varying operating conditions.

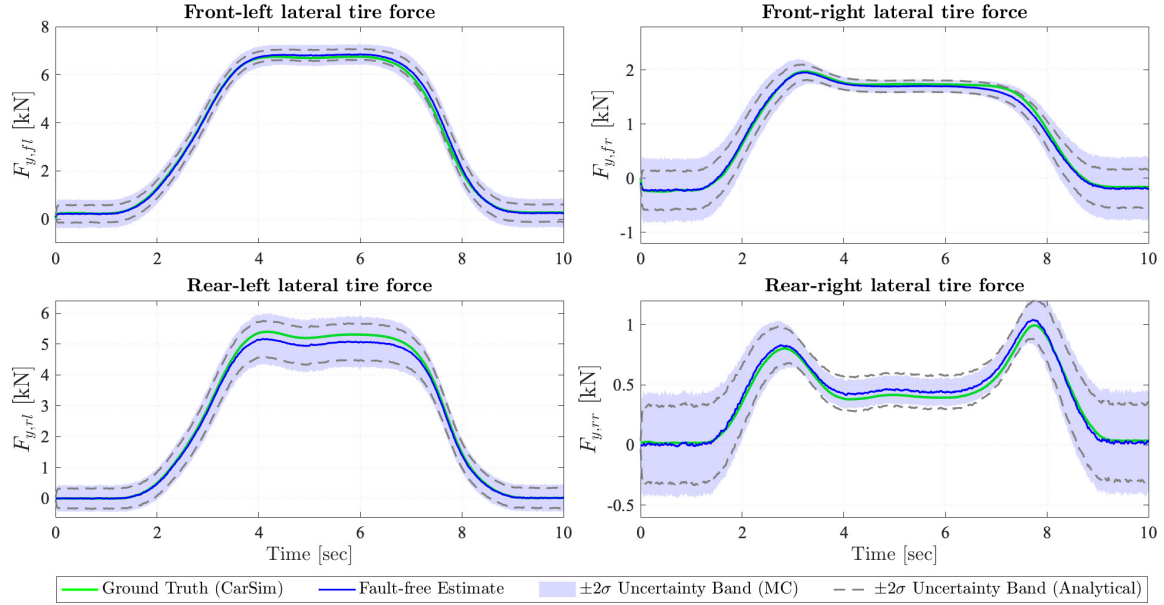
6.2.2 Surrogate-Based Uncertainty Quantification (Non-intrusive)

To validate the proposed surrogate-based uncertainty quantification framework, the projection-based stochastic collocation method is applied along the estimation path illustrated in Figure 6.12, using a co-simulation environment that integrates CarSim and Simulink.

A bias fault is introduced in the lateral acceleration measurement a_y , modeled as a zero-mean Gaussian random variable with a variance of $0.5 \text{ m}^2/\text{s}^4$. A univariate generalized polynomial chaos (gPC) expansion of order five is constructed using Hermite polynomials, which are orthogonal with respect to the Gaussian distribution.



(a) Estimation uncertainty under bias fault in $a_{y,m}$.



(b) Estimation uncertainty under excessive noise fault in $a_{y,m}$.

Figure 6.15: Comparison of analytical and Monte Carlo-based estimation uncertainty in lateral tire forces under (a) bias fault and (b) excessive noise fault in a_y . The variance of both faults is set to $0.5 \text{ m}^2/\text{s}^4$.

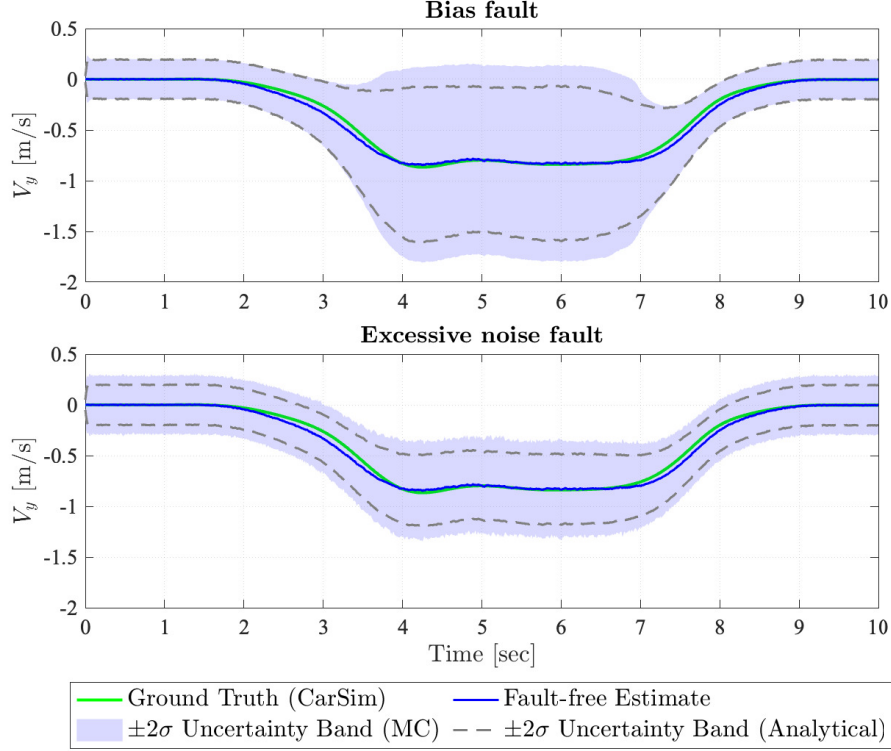


Figure 6.16: Analytical vs. Monte Carlo-simulated quantification of estimation uncertainty of lateral velocity V_y under bias and excessive noise faults in a_y measurement. The variance of both faults is set to $0.5 \text{ m}^2/\text{s}^4$.

To perform the projection, a total of $N_q = 6$ collocation points are selected using Gaussian quadrature. This choice corresponds to $P + 1$ points for a polynomial of degree $P = 5$, ensuring exact integration for polynomials of degree up to $2P + 1 = 11$. The resulting bias sample set of $a_{y,m}$ is:

$$\text{bias samples} = [-1.662 \quad -0.945 \quad -0.308 \quad 0.308 \quad 0.945 \quad 1.662]^\top.$$

For each sample, the black-box estimator is simulated over the full trajectory, resulting in a matrix of estimated states with dimensions $N_q \times T$, where T is the number of time steps.

The gPC coefficients $c_k(t)$ are computed at each time step via projection using the quadrature weights and orthogonal basis functions. These coefficients are then used to derive the mean and covariance of the estimated states $\hat{\mathbf{x}}(t)$ as given in (4.44)–(4.45). The

Table 6.5: Comparing time-averaged RMSE from Monte Carlo and analytical SD ($\bar{\sigma}_a$) for estimated states under various faults.

Fault source	Fault type	Metric	States				
			F_{yfl} (kN)	F_{yfr} (kN)	F_{yrl} (kN)	F_{yrr} (kN)	V_y (m/s)
$a_{y,m}$	Bias	RMSE	0.33	0.27	0.56	0.18	0.24
		$\bar{\sigma}_a$	0.32	0.26	0.53	0.17	0.25
	Noise	RMSE	0.28	0.17	0.31	0.13	0.18
		$\bar{\sigma}_a$	0.26	0.15	0.30	0.13	0.16
$a_{z,m}$	Bias	RMSE	0.18	0.07	0.06	0.26	0.23
		$\bar{\sigma}_a$	0.17	0.07	0.06	0.24	0.22
	Noise	RMSE	0.07	0.03	0.08	0.12	0.05
		$\bar{\sigma}_a$	0.06	0.02	0.07	0.12	0.04
$\dot{\psi}_m$	Bias	RMSE	0.33	0.23	0.43	0.23	0.03
		$\bar{\sigma}_a$	0.31	0.21	0.40	0.21	0.03
	Noise	RMSE	0.14	0.10	0.20	0.11	0.02
		$\bar{\sigma}_a$	0.13	0.10	0.21	0.11	0.02

time-varying coefficients, shown in Figure 6.17, offer additional insight into the spectral contribution of each polynomial basis to the uncertainty profile over time.

The surrogate-based uncertainty results are compared against a Monte Carlo benchmark to validate the accuracy and efficiency of the method, as shown in Figures 6.18 and 6.19. The agreement between the two confirms the validity of the non-intrusive gPC framework for quantifying estimation uncertainty under known fault distributions.

To further assess the accuracy of the surrogate-based uncertainty quantification method, Table 6.6 compares the time-averaged standard deviation obtained from the gPC surrogate ($\bar{\sigma}_{gPC}$) against the RMSE computed from Monte Carlo simulations for three fault scenarios: bias in $a_{y,m}$, $a_{z,m}$, and $\dot{\psi}_m$ along the estimation path in Figure 6.12. The results demonstrate close agreement between $\bar{\sigma}_{gPC}$ and the RMSE values across all estimated states, indicating that the gPC-based surrogate accurately captures the magnitude and distribution of the estimation uncertainty under various fault conditions. Compared to the analytical approach, however, the gPC-based standard deviations slightly underestimate the true error variances. This is an expected outcome due to the truncation of higher-order polynomial terms in the finite-order gPC expansion, which limits the surrogate's ability to capture rare or nonlinear deviations beyond the selected polynomial degree.

Due to the small number of quadrature points ($N_q = 6$) and the non-intrusive nature of the method, the proposed surrogate-based uncertainty quantification is potentially suitable

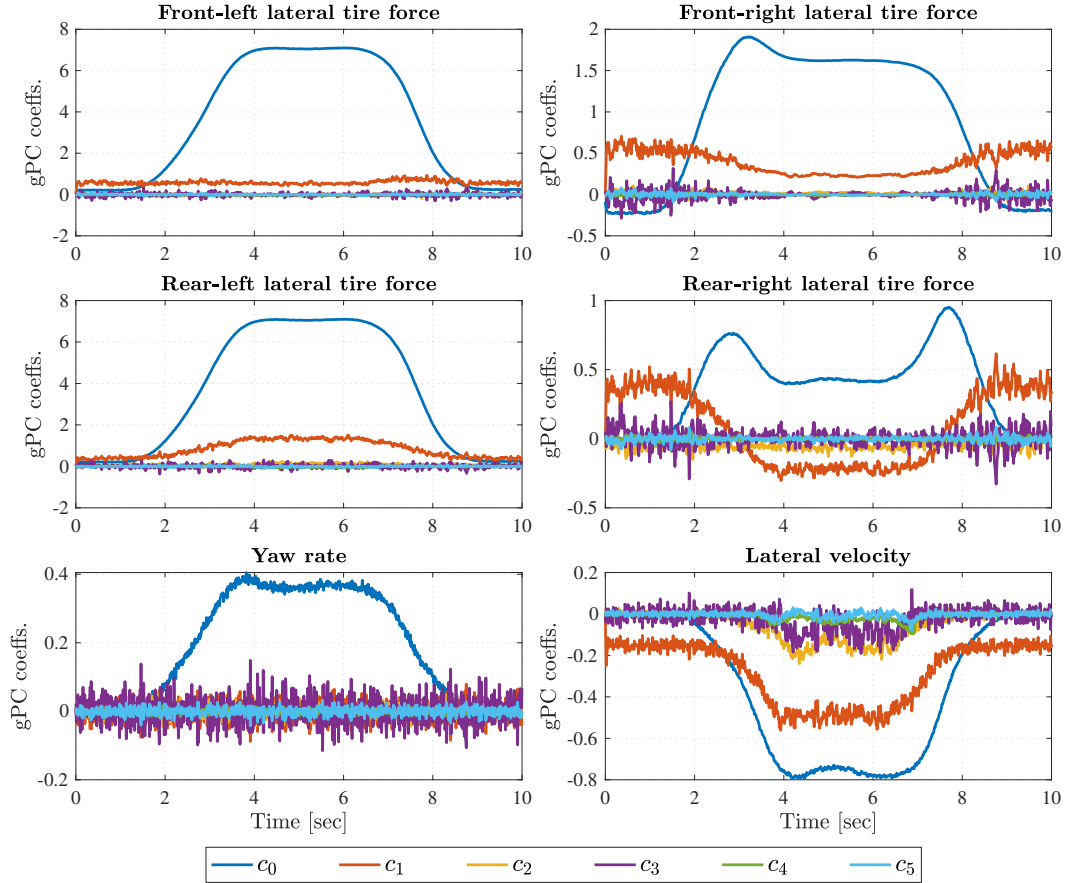


Figure 6.17: Time evolution of the generalized polynomial chaos (gPC) coefficients $c_k(t)$ associated with each orthogonal basis function.

for real-time implementation, particularly when estimator evaluations are lightweight or can be parallelized.

6.2.3 Surrogate-Based Uncertainty Quantification (Intrusive)

The Galerkin-based gPC method enables efficient and accurate propagation of uncertainty by transforming stochastic systems into deterministic ones of higher dimension. However, its intrusive nature imposes a key limitation: deriving the evolution equations for the gPC coefficients requires explicit access to the model equations. This makes the method unsuitable for cascaded or nonlinear estimators, where analytical projection becomes

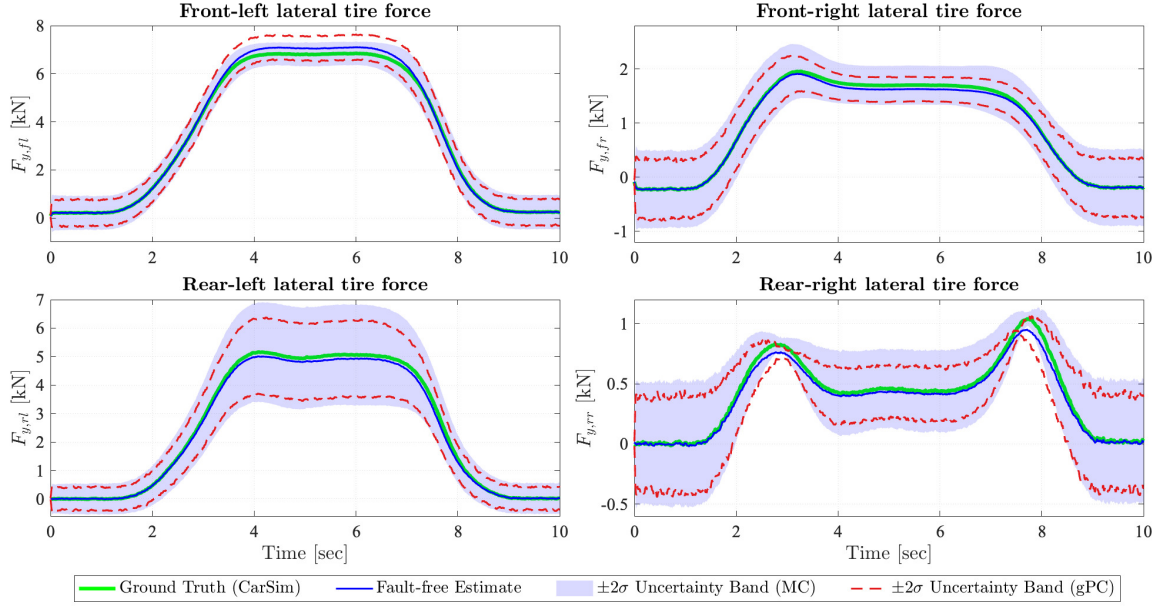


Figure 6.18: Surrogate-based vs. Monte Carlo-simulated quantification of estimation uncertainty of lateral tire forces $F_{y,ij}$ under bias fault in a_y measurement. The variance of bias fault is set to $0.5 \text{ m}^2/\text{s}^4$.

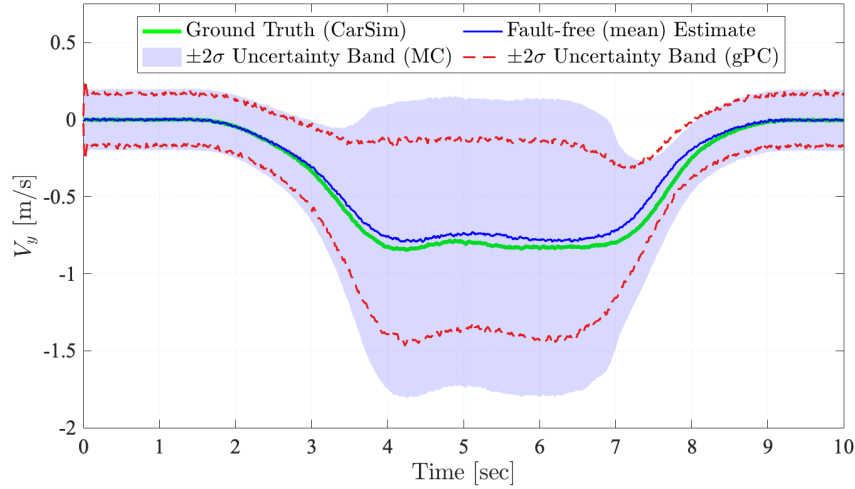


Figure 6.19: Surrogate-based vs. Monte Carlo-simulated quantification of estimation uncertainty of lateral velocity V_y under a bias fault in a_y measurement. The variance of bias fault is set to $0.5 \text{ m}^2/\text{s}^4$.

Table 6.6: Comparing time-averaged RMSE from Monte Carlo and surrogate-based SD ($\bar{\sigma}_{gPC}$) for estimated states under various faults.

Fault source	Fault type	Metric	States				
			F_{yfl} (kN)	F_{yfr} (kN)	F_{yrl} (kN)	F_{yrr} (kN)	V_y (m/s)
$a_{y,m}$	Bias	RMSE	0.33	0.27	0.56	0.18	0.24
		$\bar{\sigma}_{gPC}$	0.29	0.22	0.48	0.14	0.19
$a_{z,m}$	Bias	RMSE	0.18	0.07	0.06	0.26	0.23
		$\bar{\sigma}_{gPC}$	0.16	0.06	0.06	0.21	0.21
$\dot{\psi}_m$	Bias	RMSE	0.33	0.23	0.43	0.23	0.03
		$\bar{\sigma}_{gPC}$	0.29	0.20	0.38	0.19	0.03

intractable. Nevertheless, when applied to well-structured systems with low to moderate dimensionality and accessible dynamics, the Galerkin method provides a powerful alternative to non-intrusive and sampling-based techniques, offering accurate moment estimates with significantly reduced computational cost.

To demonstrate its applicability, the Galerkin-based approach is implemented for a linear Kalman estimator tasked with estimating the lateral velocity V_y of a vehicle using the linear bicycle model. The estimator is subjected to a bias fault in the lateral acceleration measurement a_y , modeled as a scalar random variable $f \sim \mathcal{U}(-\sqrt{6}/2, \sqrt{6}/2)$, corresponding to a variance of $0.5 \text{ m}^2/\text{s}^4$. The resulting error dynamics are given by the linear system in (4.35), where the fault propagates into the estimator via the deterministic matrix $\mathbf{R}$.

Following the formulation in Section 4.3.2, the estimation error $\tilde{\mathbf{x}}(t, f) \in L^2(\Omega; \mathbb{R}^n)$ is approximated by the truncated polynomial chaos expansion given in (4.36). Applying the Galerkin projection results in the system of deterministic equations for the gPC coefficients as expressed in (4.40), where the projection coefficients $\boldsymbol{\alpha}_k = \mathbb{E}[f \varphi_k(f)]$ reflect the contribution of the random fault to each polynomial basis term.

The system matrices used in the Galerkin implementation are defined based on the linear bicycle model and the Kalman estimator. The state transition matrix $\mathbf{A}$ corresponds to the linearized bicycle dynamics described in (A.4), and the output matrix $\mathbf{C}$ is given in (A.5). The matrix $\mathbf{K}$ represents the steady-state continuous-time Kalman gain, although a discrete-time version can be used equivalently in implementation. The fault enters through the measurement channel and is incorporated via the matrix $\mathbf{R} = \mathbf{R}_1 + \mathbf{K}\mathbf{R}_2$, where $\mathbf{R}_1 = [0 \ 0]^\top$ and $\mathbf{R}_2 = [1 \ 0]^\top$, corresponding to a bias applied to the a_y measurement.

The projection coefficients $\boldsymbol{\alpha}_k$ are computed numerically using Gauss–Legendre quadrature to reflect the uniform distribution of f . In the implementation, a quadrature order

of $N_q = 10$ is used, with nodes $\{\xi_i\}_{i=1}^{N_q}$ and weights $\{w_i\}_{i=1}^{N_q}$ defined on the interval $[-1, 1]$. The fault values are mapped to the physical domain via $f(\xi_i) = \frac{\sqrt{6}}{2} \xi_i$, and the orthonormal Legendre polynomials $\varphi_k(\xi)$ are evaluated recursively. Each projection coefficient is then approximated as

$$\alpha_k \approx \sum_{i=1}^{N_q} w_i f(\xi_i) \varphi_k(\xi_i), \quad k = 0, \dots, P.$$

These coefficients appear as deterministic forcing terms in the evolution equations of the gPC coefficients.

Once the coefficients $\mathbf{c}_k(t)$ are computed, the statistical moments of the estimation error are obtained using the expressions in (4.30) and (4.31). The results are shown in Figure 6.20, which compares the variance of the lateral velocity estimation error computed using the Galerkin method against that obtained from a Monte Carlo simulation with 1000 samples. The vehicle is assumed to travel at a constant longitudinal velocity of 70 km/h. As observed, the Galerkin-based solution closely matches the Monte Carlo benchmark, confirming both the accuracy and computational efficiency of the intrusive gPC approach. This highlights the value of surrogate-based uncertainty quantification in applications where the model is known and computational resources are limited.

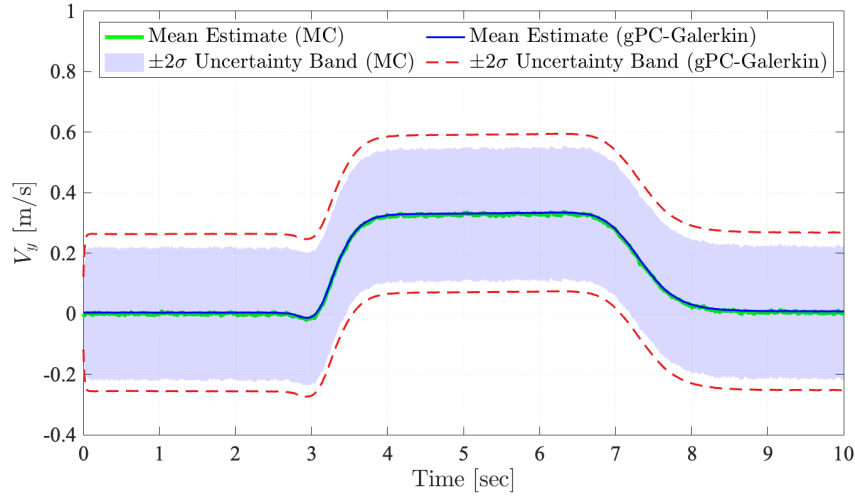


Figure 6.20: Intrusive surrogate-based (Galerkin method) vs. Monte Carlo-simulated quantification of estimation uncertainty of lateral velocity V_y under a bias fault in a_y measurement. The variance of the bias fault is set to $0.5 \text{ m}^2/\text{s}^4$. The longitudinal velocity is constant at 70 km/h.

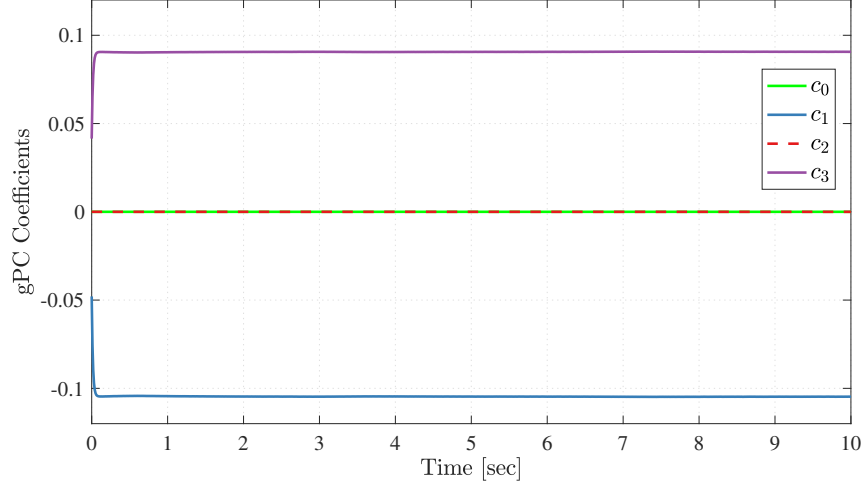


Figure 6.21: Time evolution of the generalized polynomial chaos (gPC) coefficients $c_k(t)$ associated with each orthogonal basis function.

6.2.4 Computational Performance Analysis

This subsection evaluates the computational efficiency of the uncertainty quantification (UQ) methods presented in Sections 4.2 and 4.3, using a representative case study. The computational complexities of the analytical and gPC-based methods are detailed in Sections 4.2.1 and 4.3.4, respectively. The objective here is to highlight the runtime advantages of the proposed analytical and surrogate-based UQ approaches in comparison to traditional sampling-based methods, such as Monte Carlo (MC) simulation.

Table 6.7 summarizes the actual execution times of three methods used to compute state covariance matrices across the estimation path shown in Figure 6.14. The reported runtimes reflect only the execution time of the estimator and UQ computations within the Simulink–CarSim co-simulation framework. The computational cost of the vehicle dynamics model itself is excluded. All simulations were performed on an Apple M3 chip with 16 GB of RAM.

As detailed in Section 4.2, the proposed analytical UQ method leverages the recursive structure of Kalman-based estimators to propagate covariance information efficiently. This approach completed the covariance computation in just 0.0024 s, owing to its closed-form expressions derived in (4.17)–(4.20). In contrast, the MC method—performed using $N = 10^3$ samples of a bias fault in a_y —required 1.3659 s to compute the same statistics. This

corresponds to a speed-up factor of over 600x, demonstrating the computational scalability and suitability of the analytical UQ approach for real-time or embedded applications.

In scenarios where analytical propagation is not feasible—such as nonlinear or black-box estimators—the gPC-based stochastic collocation method introduced in Section 4.3.3 provides a tractable surrogate-based alternative. In the case study, a fifth-order expansion with $N_q = 6$ quadrature points was used to quantify uncertainty due to the same bias fault. The total runtime of the gPC-based method was 0.0606 s, which is nearly 20 times faster than the MC approach, while delivering comparable accuracy in mean and variance estimates. Once constructed, the gPC surrogate enables efficient reuse for multiple queries, making it particularly advantageous in control loops or repeated simulations.

Overall, these results confirm the findings from the theoretical performance analysis: the analytical UQ method offers the lowest runtime with exact moment propagation for linear estimators, while the gPC surrogate balances flexibility and efficiency for nonlinear or black-box settings. Both methods dramatically outperform MC simulation in terms of computational cost, reinforcing their practicality for real-time uncertainty-aware estimation frameworks.

Table 6.7: Comparison of runtime for computing state covariance matrices across the estimation path in Figure 6.12.

Method	Runtime	Remarks
Monte Carlo	1.3659 s	$N = 10^3$ samples
Analytical UQ	0.0024 s	Closed-form (4.17)-(4.20)
Surrogate-based UQ	0.0606 s	gPC with $P = 5$ and $N_q = 6$

Chapter 7

Fault Detection and Diagnosis Results

This chapter presents the results of the multi-stage hybrid framework introduced in Chapter 5 for fault detection, isolation, and identification in a vehicle system. The vehicle's motion estimation is modeled as a networked dynamical system using an estimation graph. The framework integrates model-based and data-driven approaches to enhance fault detection and diagnosis (FDD) performance. Temporal and spatial distributions of residuals are mathematically quantified to enable robust fault isolation, while data-driven techniques are employed to address unforeseen uncertainties and unmodeled dynamics. The framework operates in multiple stages to ensure high accuracy and reliability in complex systems (see Figure 7.1). The remainder of this chapter is organized as follows: Section 7.1 covers the fault detection stage; Section 7.2 focuses on fault isolation; and Section 7.3 addresses fault magnitude identification. Additionally, Section 7.4 further investigates the two distributed fault isolation methods developed in 5.2.6 and 5.2.5, evaluating their accuracy and computational efficiency, particularly in the context of large-scale systems.

Based on detectability and sensitivity analysis, the selected residuals for the the multi-stage FDI framework are:

$$\mathbf{r} = \begin{bmatrix} r_{\dot{\phi}}^{(E_1)} & r_{\dot{\psi}}^{(E_5)} & r_{a_y}^{(E_5)} & r_{\dot{\psi}}^{(E_8)} & r_{a_y}^{(E_8)} \end{bmatrix}^T. \quad (7.1)$$

To generate training and testing data for the detection task, 100 simulations with an average length of 6 sec and sampling frequency of 1 Hz were run. Each simulation contains one fault occurring at a random time within the simulation interval. For the isolation task, another 100 simulations were conducted under the same conditions. Table 7.1 provides the data statistics for both tasks. In the detection task, there are 62,127 time samples, of which

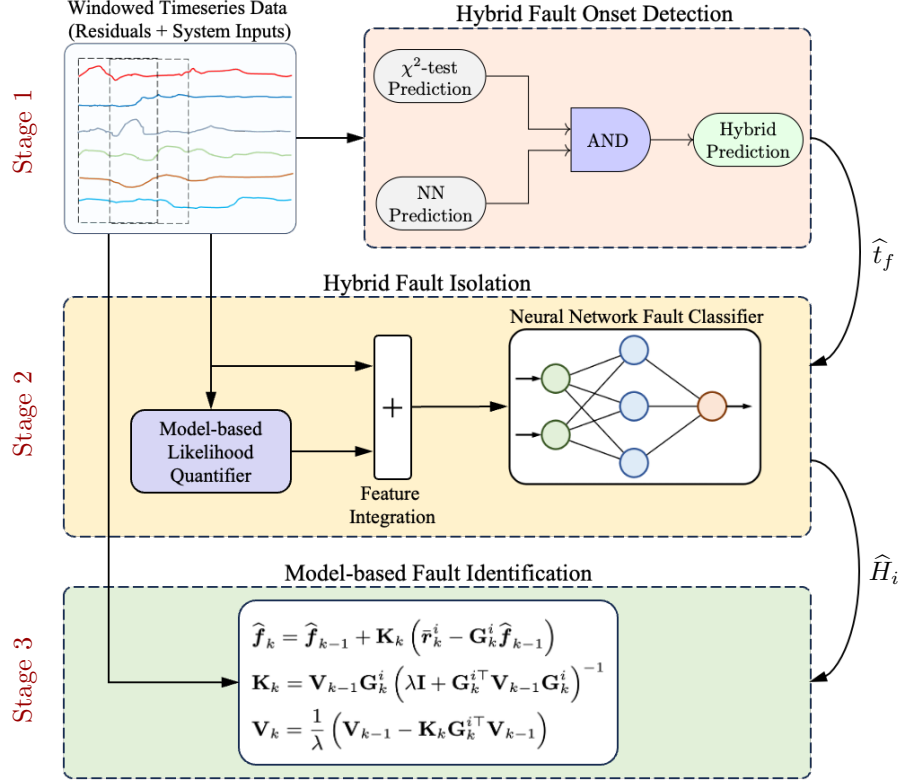


Figure 7.1: Multi-stage hybrid FDD framework. $\hat{t}_f$ is the detected fault onset time, and the hypothesis $\hat{H}_i$ denotes the isolated fault class.

86.32% are non-faulty. A significant portion of the data is non-faulty to reflect realistic conditions. The data was split into 80% for training and 20% for testing. For the isolation task, the network was trained on 46,850 time samples, with 34,391 sliding windows (40 steps each), split 80% for training and 20% for testing.

7.1 Fault Onset Detection Stage

7.1.1 Residual-based Fault Detection

Model-based fault detection is implemented using the χ^2 statistic in (5.1). A sliding window size of $l = 40$ is selected for robust detection through trial and error. The threshold

Table 7.1: Total dataset time samples and percentage for each fault case in 100 simulations for detection and isolation tasks.

Task	Samples	No fault	$a_{y,m}$	$a_{z,m}$	ψ_m	$a_{y,\text{wind}}$
Detection	62127	86.32%	4.06%	2.51%	4.58%	2.53%
Isolation	46850	–	29.61%	18.38%	18.50%	33.51%

corresponds to a 5% significance level with degrees of freedom $df = 120$. Figure 7.2 depicts residual behavior in non-faulty conditions during smooth and harsh cornering. In the smooth maneuver, residuals are nearly zero-mean, as expected. However, in the harsh maneuver, residuals deviate from zero-mean due to unmodeled dynamics during sharp steering, causing false positives in the χ^2 -test. The highlighted regions indicate where the χ^2 statistic exceeds the threshold. The χ^2 -test achieved 84.4% accuracy, with most false alarms resulting from nonlinearities and unmodeled dynamics in harsh maneuvers. This underscores the need for an alternative approach, like data-driven methods with pattern recognition capabilities to reduce false positive alarms.

7.1.2 Data-driven Fault Detection

Here, a neural network-based fault detector, designed to overcome the limitations of model-based χ^2 -test, is explored. Most false alarms in the χ^2 -test occur during sharp steering changes, so the steering angle δ along with the residuals in (7.1) are included as features:

$$\delta, \quad r_{\phi}^{(E1)}, \quad r_{\psi}^{(E5)}, \quad r_{a_y}^{(E5)}, \quad r_{\psi}^{(E8)}, \quad r_{a_y}^{(E8)}$$

for the network in Figure 5.2. At each time step k , a sliding window of $[k - l : k]$ with $l = 40$ is fed into the network.

Table 7.2 represents the contributions of Convolutional and RNN layers in the detection network’s performance. Recall is defined as

$$\frac{\text{TP}}{\text{TP} + \text{FN}},$$

where TP is true positives and FN is false negatives. It measures a classifier’s ability to identify all positive alarms. While accuracy decreases a bit, replacing the RNN layer with a Dense layer results in a significant drop in recall, highlighting the importance of RNNs for detecting positive alarms. Additionally, using a Convolutional layer to capture spatial dependencies among features improves both accuracy and recall by about 1%.

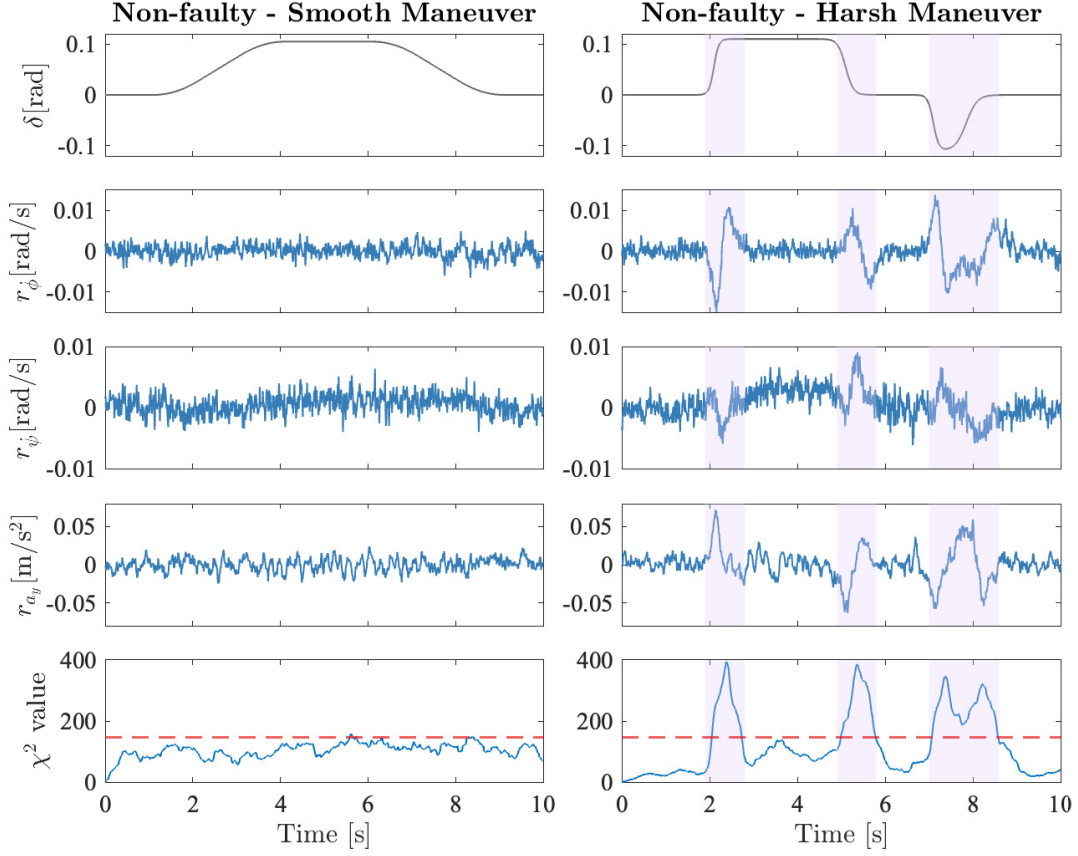


Figure 7.2: The behavior of residuals in non-faulty conditions under both smooth and harsh cornering maneuvers. For harsh maneuvers, residuals do not maintain zero mean.

7.1.3 Hybrid Fault Detection

Figure 7.3 compares the fault predictions of the χ^2 -test and the neural network. The neural network provides a range of false positives around $t = 1$ sec despite no residual deviations. In contrast, the χ^2 -test gives false positives between $2.4 - 3$ sec, triggered by sharp steering wheel angle changes, while the neural network correctly identifies these as non-faulty. The hybrid method combines both outputs using voting logic, outperforming both individual methods with an accuracy of 99.6%.

Table 7.2: Comparing the contribution of Conv and RNN layers to network performance in Figure 5.2 on the test dataset.

Metrics	Conv + RNN	Conv + Dense	Dense + RNN
Accuracy	98.7 %	97.5 %	97.6 %
Recall	98.5 %	89.9 %	97.4 %

7.2 Fault Isolation Stage

In this section, the results of fault isolation using model-based, data-driven, and hybrid approaches are presented. The bias faults in measurements a_y , a_z , and $\dot{\psi}$, as well as the unknown lateral disturbance d , are modeled as random constants with zero mean and variances of 0.5, 1, 0.01, and 0.5, respectively.

7.2.1 Model-based Likelihood Test for Fault Isolation

The model-based fault isolation results using the likelihood test from Section 5.2.2 are shown. The results are divided into two parts:

Residual distribution under faults. The residual covariance matrices for different fault cases are derived from (5.7), and the square roots of its diagonal elements are plotted as $1\text{-}\sigma$ uncertainty bands in Figure 7.4. The figure also presents the three main residuals in estimation path 1 under a $\sqrt{0.5}\text{ m/s}^2$ bias fault in a_y (each residual in a separate row). In the first column, corresponding to the a_y bias, the residuals closely follow the uncertainty bands, indicating that the model effectively captures the fault dynamics. In contrast, for the other three fault cases (columns), there is less alignment between the residuals and their uncertainty bands. This underscores the accuracy of quantified uncertainty bands in reflecting temporal distribution of residuals under a given fault. In the next part, the residual covariance matrix is used to compute the likelihood of residuals given each fault case, enabling the likelihood test for fault isolation.

Likelihood Test. The likelihood of residuals under four fault scenarios are quantified by applying the log-likelihood over a 40-step window, as given in (5.5). The covariance matrix for each fault, $\Sigma_k(i)$, is obtained by (5.7) and (5.8). The selected residual vector is $\mathbf{r} = \begin{bmatrix} r_{\dot{\phi}}^{(E_1)}, r_{\dot{\psi}}^{(E_5)}, r_{a_y}^{(E_5)} \end{bmatrix}$.

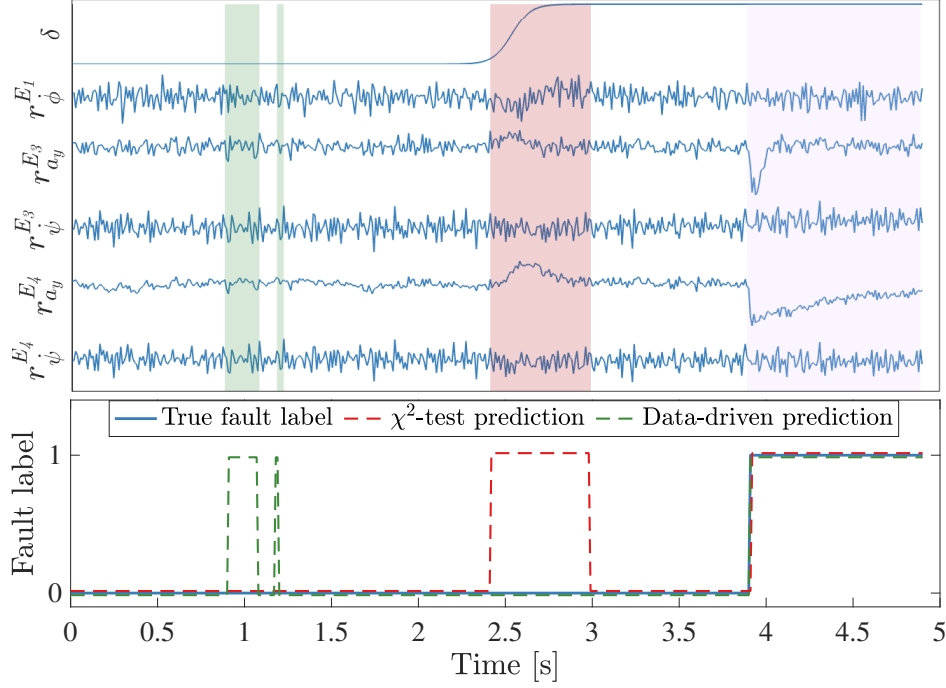


Figure 7.3: Comparing fault detection performance using residual patterns.

Figure 7.5 shows the likelihood values of residuals for four fault cases, as depicted in Figure 7.4. The likelihood values are computed over time for each fault case. Before the fault occurrence at $t = 5$ s, all likelihoods are equal. After the fault is detected at $t = 5$ s, the likelihood for the bias in $a_{y,m}$ becomes the highest, confirming it as the most probable fault. This demonstrates the effectiveness of the likelihood test in fault isolation. However, when the fault magnitude is small or under harsh conditions due to unmodeled dynamics impact, the likelihood test may struggle to reliably isolate the fault. Figure 7.6 shows the likelihoods for different fault scenarios with a small bias in yaw rate ψ . As seen, it is difficult to say the likelihood corresponding to the bias in yaw rate $\dot{\psi}_m$ is higher during the period from 0 – 2.4 s.

In Figure 7.7a, the confusion matrix for fault isolation performance using the likelihood test is shown. The test achieves high accuracy for isolating faults in $a_{y,m}$ and $\dot{\psi}_m$, correctly identifying 87.95% and 86.94% of cases, respectively. However, it exhibits significant confusion between biases in $a_{z,m}$ and external disturbance $a_{y,\text{wind}}$, with only 74.74% of $a_{z,m}$ and 33.76% of $a_{y,\text{wind}}$ correctly isolated. Notably, 55.93% of d instances are misclassified as a_z . This confusion is expected, as the effects of $a_{z,m}$ bias and external disturbance $a_{y,\text{wind}}$ on

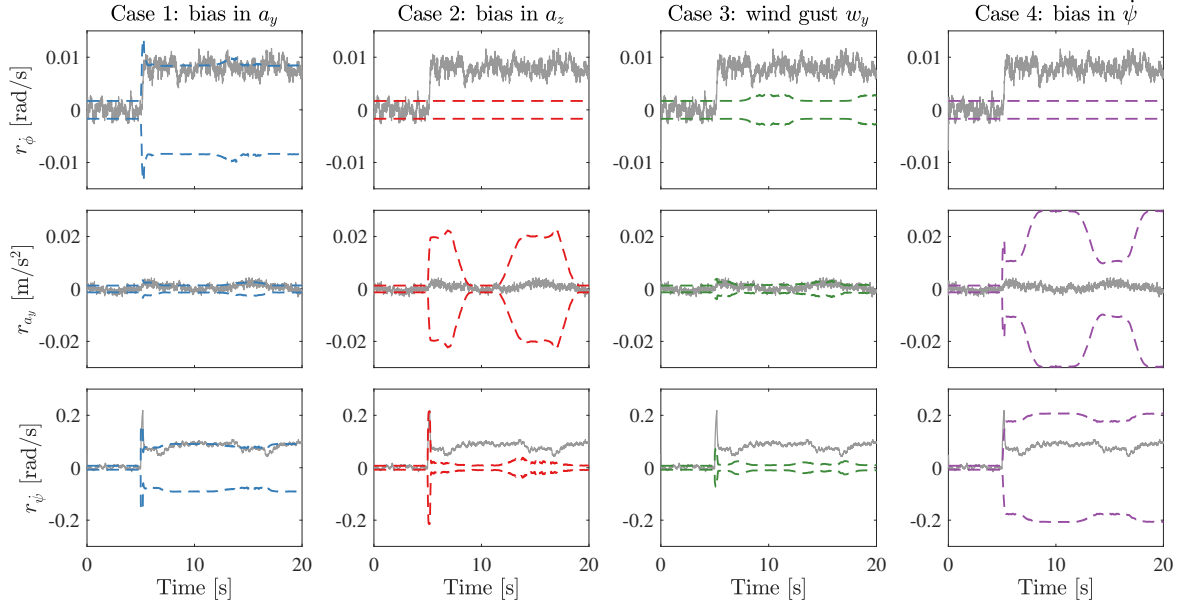


Figure 7.4: Estimation residuals under a bias fault in $a_{y,m}$ are shown versus the $1\text{-}\sigma$ uncertainty band for different fault cases. The fault occurs at $t = 5$. As expected, the residuals closely match the uncertainty bound in column 1, corresponding to bias in $a_{y,m}$.

the estimation error of lateral tire forces are nearly identical. This low accuracy highlights the limitations of model-based likelihood tests.

7.2.2 Data-driven Fault Isolation Results

An RNN-based neural network, similar to the one in Figure 5.2 and using the same input features, is implemented. The key difference is that it classifies four fault classes instead of two.

Figure 7.7b shows the confusion matrix for fault isolation performance on the test data, achieving a total accuracy of 98%. Compared to the model-based approach (Figure 7.7a), which struggled with accuracy for a_z and d due to confusion between these faults, the data-driven method significantly improves accuracy across all fault types.

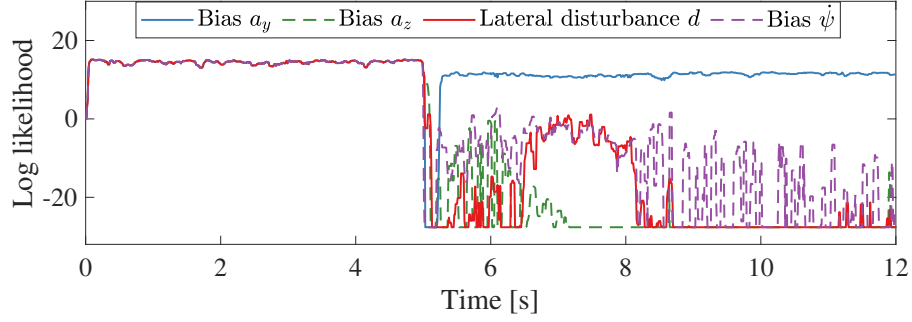


Figure 7.5: Log-likelihood values of residuals for different fault types, with a bias of 0.7 m/s^2 in $a_{y,m}$ occurred at $t = 5 \text{ s}$. As expected, the likelihood for the $a_{y,m}$ bias is highest.

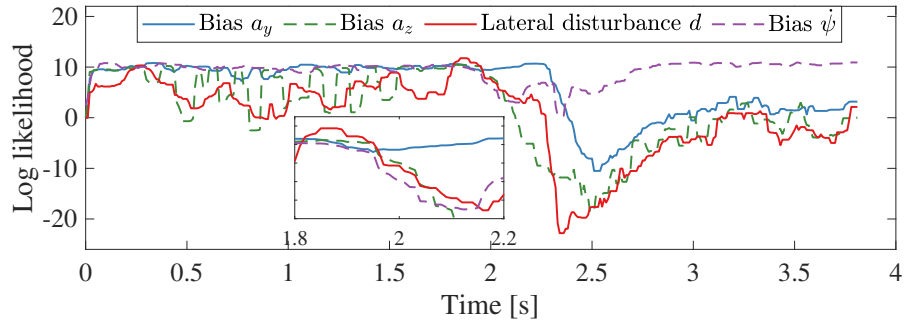


Figure 7.6: Log-likelihood values of residuals for different fault types, with a bias of 0.01 rad/sec in ψ_m occurred at $t = 0 \text{ s}$.

7.2.3 Hybrid Fault Isolation Results

This section presents the fault isolation results using the network structure in Figure 5.6. Figure 7.7c shows the confusion matrix for the hybrid approach, achieving a test accuracy of 99.5%, which is 1.5% higher than the pure data-driven method. While the data-driven approach in Section 7.2.2 performed well, its success was largely due to the richness of the dataset. In cases with limited data, the data-driven method may struggle to maintain high accuracy. The model-based likelihood values in the hybrid approach significantly enhance accuracy, particularly in data-scarce scenarios.

Figure 7.8 shows the test accuracy of model-based, data-driven, and hybrid isolation

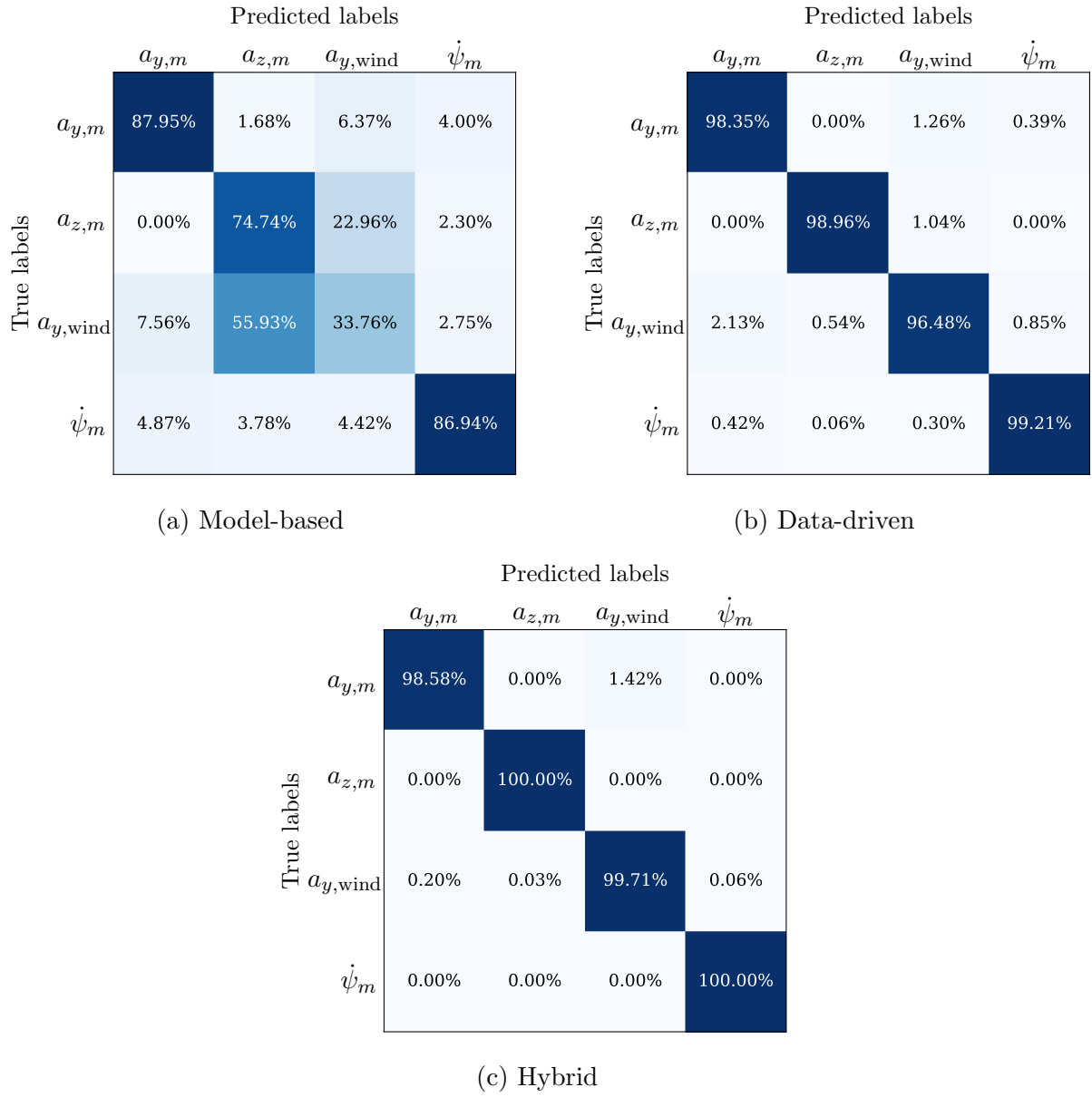


Figure 7.7: Comparison of confusion matrices for the model-based, pure data-driven, and hybrid fault isolation approaches.

approaches versus different dataset sizes. The model-based approach maintains consistently low accuracy regardless of data size, while the data-driven approach improves with larger datasets, reaching approximately 99%. The hybrid approach outperforms both, achieving high accuracy even with smaller datasets and reaching 99.5% as the data size increases. This shows the robustness of the hybrid method in combining data-driven learning with model-based likelihood information, especially in data-limited scenarios.

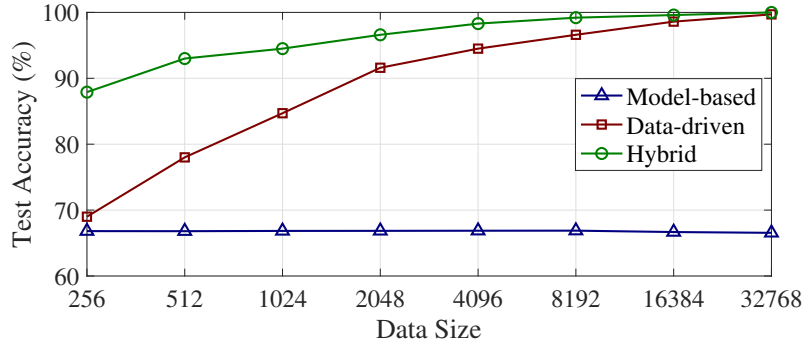


Figure 7.8: Isolation accuracy of different methods vs. data size.

7.3 Fault Identification Stage

This section presents the fault magnitude identification. After the fault type is isolated in stage 2, the formulation in (5.21) is used to identify the fault magnitude. A proper residual vector satisfying the fault detectability conditions is selected. For both $a_{y,m}$ and $a_{z,m}$ bias faults, estimator E_3 residuals, $\mathbf{r}_k^{(E_3)}$, are used.

Figure 7.9 shows the model-based identification of a bias fault in $a_{y,m}$, where the algorithm accurately captures the fault magnitude. Note that the fault matrix $\mathbf{G}_k^i$ in (5.21) accounts for the impact of $a_{y,m}$ bias in E_1 as well to ensure accurate identification. In Figure 7.10, the identification of a bias fault in $a_{z,m}$ along with the steering angle δ as the system input is shown. Two key points are noted: 1) The $a_{z,m}$ bias is undetectable in E_2 due to its random walk process, but it appears in E_3 residuals, allowing its successful identification; 2) The a_z bias fault is undetectable by E_3 residuals when there is no lateral motion (i.e., $\delta = 0$). Thus, as expected after $t = 7$ s, the identified $a_{z,m}$ bias converges to an incorrect value.

In summary, a hybrid multi-stage FDD algorithm is presented. The proposed hybrid fault detection method combines χ^2 -test with a neural network, achieving a detection accuracy

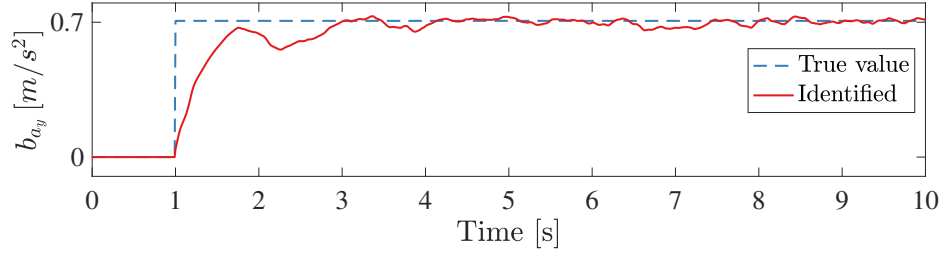


Figure 7.9: Model-based identification of a bias fault of 0.7 m/s^2 in $a_{y,m}$ occurred at $t = 1 \text{ s}$.

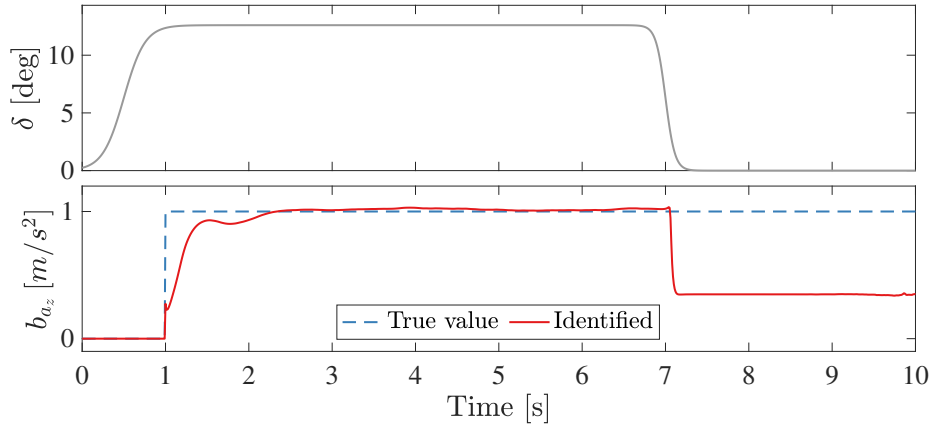


Figure 7.10: A model-based identification of a 1 m/s^2 bias in $a_{z,m}$ occurring at $t = 1 \text{ s}$. As expected, with the fault undetectable at $\delta = 0$, the value converges incorrectly after $t = 7 \text{ s}$.

of 99.6% by effectively reducing false positives. For fault isolation, the hybrid approach integrates model-based likelihoods with neural network classifiers, leading to 99.5% accuracy. This method outperforms individual likelihood test and data-driven technique, especially for fault cases with similar residual patterns. Moreover, the proposed hybrid isolation method remains effective in scenarios with limited training data, leveraging model-based priors to enhance classification. Fault impacts on residuals across the estimation graph are mathematically quantified, enabling accurate model-based identification of fault magnitudes. While $a_{z,m}$ bias is challenging to identify due to insensitivity of the $a_{z,m}$ residual in E_2 , its impact in E_3 is successfully formulated, enabling its accurate identification.

The multi-stage structure of the framework allows for flexible integration at each stage—detection, isolation, and identification—ensuring improved reliability, robustness,

and adaptability in large-scale vehicle fault diagnosis systems. This staged approach ensures that each phase is optimized for its specific task, resulting in more precise and comprehensive fault handling across different operating conditions.

The next section focuses on distributed fault isolation in large-scale systems, where structural information is leveraged to address challenges related to scalability, fault propagation, and the lack of labeled data for certain fault cases.

7.4 Distributed Fault Isolation in Large-Scale Systems

This section presents a scalable distributed fault isolation framework tailored for large-scale networked dynamical systems (NDSs). While modular and decentralized architectures alleviate the scalability and maintainability limitations of centralized estimators, they also introduce new challenges, including fault propagation, cascading failures, and coordination under communication constraints. In this context, leveraging the system’s structural redundancies becomes essential to ensure robust and reliable fault isolation, especially in the presence of subsystem heterogeneity and partial observability.

The proposed framework addresses this by assigning lightweight local detectors to fault sources, propagating information over a fault interaction graph, and exploiting structural connectivity to improve isolability, adaptability, and fault localization performance in evolving large-scale systems. Two approaches are developed to implement this framework. In Section 5.2.5, a consensus-based label propagation method is introduced, which enables fault isolation under partial supervision by diffusing label information through the graph structure. In Section 5.2.6, a GCN-based approach is proposed, which learns to aggregate features from neighboring nodes for more accurate and topology-aware fault classification.

To demonstrate the proposed fault isolation framework, a large-scale interconnected vehicle system subject to various fault types is considered. As illustrated in Figure 7.11, the system integrates numerous sensors and actuators whose measurements are used for state estimation and control. These components are susceptible to faults such as bias errors. In addition to these hardware-related faults, the vehicle is also affected by unmeasurable external disturbances (e.g., wind gusts or road slope changes) and uncertain parameters like tire-road friction. Table 7.3 summarizes the fault sources considered in this study, along with their ranges. Sensor and actuator faults are introduced as random constant biases within predefined bounds, while disturbances and parameter errors are modeled as deviations from near-zero or nominal conditions. The objective is to accurately isolate these diverse fault sources in a scalable and structure-aware manner, leveraging the system’s modular design and fault interdependencies.

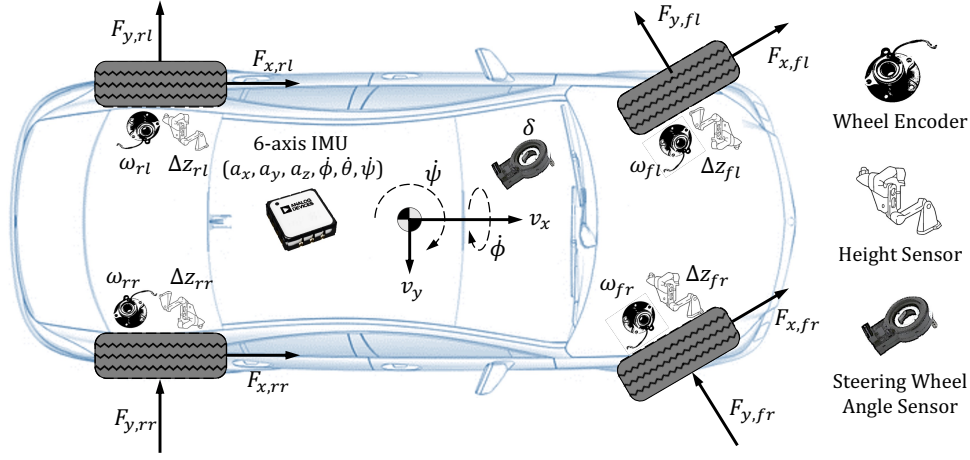


Figure 7.11: Sensor layout for fault diagnosis framework.

7.4.1 Vehicle Fault Interaction Graph

In Section 3.3, the structural fault signature matrix in Table 3.3 was derived from the estimation graph topology, capturing the binary connectivity between residuals and fault sources. While this indicates potential fault propagation paths, it does not guarantee significant or informative fault signatures. In practice, structurally connected faults may have negligible influence on residuals due to low sensitivity, weak detectability, or measurement noise. Moreover, residuals from different estimation paths may exhibit high correlation, resulting in redundancy and reduced diagnostic value. To enable accurate and efficient data-driven fault detection using local detectors, sensitivity and correlation analyses are performed to identify a compact and informative subset of residuals for each fault source. This leads to a sparser and more effective fault signature matrix, tailored for training localized fault detectors.

Reducing Feature Dimensionality via Correlation and Sensitivity Analyses

To construct a sparse and effective fault signature matrix, two complementary techniques are applied to the candidate features (residuals) associated with each fault: (1) Pearson correlation analysis and (2) gradient-based sensitivity analysis.

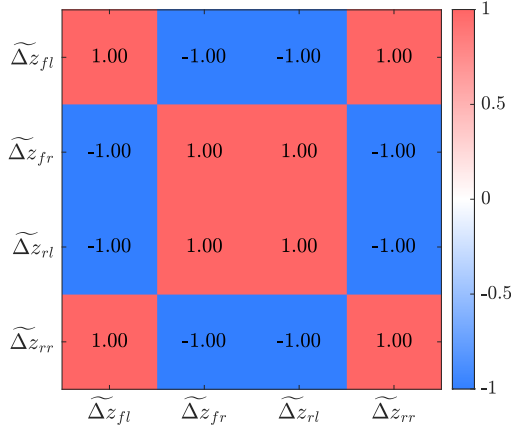
First, the Pearson correlation coefficient is computed for all residuals to identify and remove redundant features. Highly correlated residuals carry similar information and may

Table 7.3: Summary of fault sources, associated variables, and injected fault ranges. Sensor and actuator faults are simulated as constant random biases within the specified ranges. Unknown disturbances and parameter errors are modeled as deviations from near-zero or nominal values.

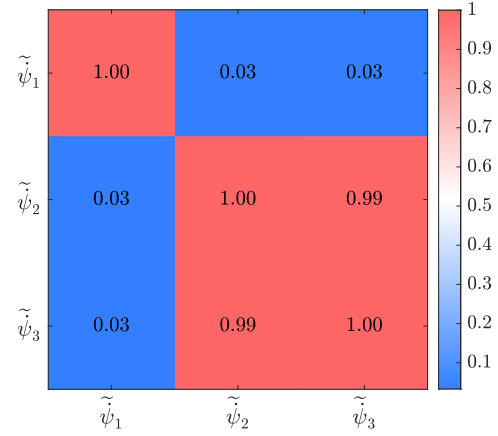
Fault Source	Variable Description	Fault Range
Sensor/Actuator Faults	δ : Measured steering wheel angle	$[0.01, 0.1]$ rad
	a_x : Measured longitudinal acceleration	$[0.1, 1]$ m/s ²
	a_y : Measured lateral acceleration	$[0.1, 1]$ m/s ²
	a_z : Measured vertical acceleration	$[0.2, 2]$ m/s ²
	$\dot{\phi}$: Measured roll rate	$[0.01, 0.1]$ rad/s
	$\dot{\psi}$: Measured yaw rate	$[0.02, 0.2]$ rad/s
	Δz_{ij} : Measured suspension height	$[0.5, 5]$ cm
Unknown External Disturbances	ω_{ij} : Measured wheel speed	$[2, 20]$ rad/s
	$a_{y,\text{wind}}$: Lateral wind disturbance	$50 \text{ N} < a_{y,\text{wind}} < 150 \text{ N}$
	θ_r : Road grade angle	$10 \text{ deg} < \theta_r < 45 \text{ deg}$
Parameter Error	ϕ_r : Road bank angle	$10 \text{ deg} < \phi_r < 45 \text{ deg}$
	μ_r : Tire-road friction coefficient	$0.1 < \mu_r < 0.6$

degrade classifier generalization due to multicollinearity. The correlation matrix in Figure 7.12a illustrates that the suspension height sensor residuals (i.e., $\widetilde{\Delta z}_{fl}$, $\widetilde{\Delta z}_{fr}$, $\widetilde{\Delta z}_{rl}$, $\widetilde{\Delta z}_{rr}$) are either perfectly positively or negatively correlated (correlation coefficient = 1 or -1). Structurally, each suspension height sensor fault affects all four residuals simultaneously, meaning the information content across them is not distinguishable. Thus, selecting one representative residual, such as $\widetilde{\Delta z}_{fl}$, while discarding the others, is sufficient for fault isolation purposes in this case. Similarly, Figure 7.12b shows that yaw rate measurement residuals $\dot{\psi}_2$ and $\dot{\psi}_3$ are highly correlated (Pearson coefficient ≈ 0.99). Therefore, retaining just one of them, such as $\dot{\psi}_2$, suffices in the fault signature matrix without loss of isolation capability.

The next step involves identifying the most significant and sensitive features for each fault listed in Table 3.3. This process is carried out in two stages. In the first stage, the sensitivity of residuals structurally connected to each fault is analyzed. Residuals that



(a) Suspension height sensor residuals.



(b) Yaw rate sensor residuals.

Figure 7.12: Correlation matrices between residuals for two sensor groups: (a) suspension height sensors, and (b) yaw rate sensors.

exhibit a noticeable change in response to the corresponding fault—visually observable in the data—are considered sensitive and retained as candidate features, while insensitive ones are discarded. After performing correlation analysis to eliminate redundant, highly correlated features, the second stage applies gradient-based sensitivity analysis to identify and remove less significant features.

In the second stage, local detectors are trained using the selected candidate features, as described in Section 5.2.5. The gradient-based sensitivity score of each classifier with respect to its input features is computed using (5.9). Based on these scores, features are ranked in descending order of importance, and an iterative feature reduction strategy is applied. Starting from the full feature set, the least sensitive feature is removed in each iteration, and the classifier is re-evaluated to monitor its accuracy. This process continues until a noticeable drop in performance is observed, at which point the last removed feature is restored.

Figure 7.13 illustrates an example sensitivity profile for one of the local fault classifiers corresponding to the a_x measurement. As expected, residuals related to lateral and vertical motions—such as $\widetilde{a}_{y,2}$, $\widetilde{a}_z$, and $\widetilde{F}_{y,f}$ —exhibit low sensitivity to this fault. After removing these features and retraining the model using only the remaining features ($\widetilde{a}_{x,1}$, $\widetilde{a}_{x,2}$, $\widetilde{V}_x$, and $\widetilde{F}_{x,f}$), the classifier preserved its high classification accuracy. This confirms that a compact

subset of features is sufficient for accurate fault detection, leading to a more efficient and interpretable models. The same procedure is repeated for all fault sources.

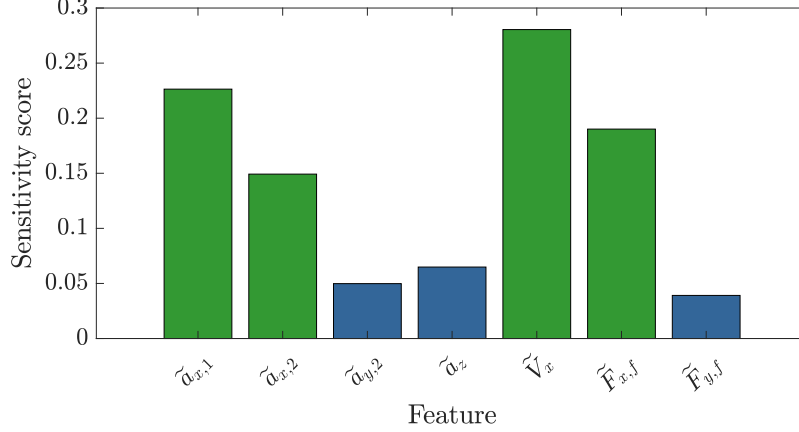


Figure 7.13: Sensitivity scores of the trained neural network model for detecting a fault in the a_x measurement with respect to different residual features.

Finally, the refined fault signature matrix resulting from the feature dimensionality reduction process is presented in Table 7.4. This matrix is sparse, with most entries equal to zero, indicating that each fault influences only a small subset of residuals. The sparse fault signature matrix can be represented as a bipartite fault-signature graph (BFSG), capturing the relationships between faults and their associated residuals. To analyze the interactions among fault sources, the one-mode projection method described in Appendix E is applied to the BFSG, resulting in the vehicle fault interaction graph (FIG) shown in Figure 7.14. The corresponding adjacency matrix of the FIG is provided in (E.1).

7.4.2 Numerical Evaluation of Local Fault Detectors

This section presents a numerical evaluation of the local supervised and unsupervised fault detection models introduced in Section 5.2.5.

To construct the dataset for training and evaluating the proposed fault diagnosis framework, 300 simulation runs were conducted using the high-fidelity CarSim–Simulink co-simulation environment. Each simulation lasted around 10 seconds and was sampled at 100 Hz, with a single fault injected per run, selected from the fault sources listed in Table 7.3. The fault type and injection time were randomly determined for each run, with the injection

Table 7.4: Fault signature matrix for a vehicle system. Rows represent residuals as fault signatures, and columns represent the fault sources.

Residual	Fault source																	
	δ	a_x	a_y	a_z	$\dot{\phi}$	$\dot{\psi}$	z_{fl}	z_{fr}	z_{rl}	z_{rr}	ω_{fl}	ω_{fr}	ω_{rl}	ω_{rr}	$a_{y,w}$	θ_r	ϕ_r	μ_r
$\tilde{a}_{x,1}$	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
$\tilde{a}_{x,2}$	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
$\tilde{a}_{y,1}$	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1
$\tilde{a}_{y,2}$	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\tilde{a}_z$	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\tilde{\dot{\phi}}_1$	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0
$\tilde{\dot{\phi}}_2$	0	0	1	0	1	0	1	1	1	1	0	0	0	0	0	0	1	0
$\tilde{\dot{\psi}}_1$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\tilde{\dot{\psi}}_2$	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
$\tilde{\Delta z_{fl}}$	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	0	1	0
$\tilde{\omega}_{fl}$	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1
$\tilde{\omega}_{fr}$	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1
$\tilde{\omega}_{rl}$	0	0	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0	1
$\tilde{\omega}_{rr}$	0	0	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0	1
$\tilde{V}_x$	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
$\tilde{V}_y$	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0
$\tilde{F}_{x,f}$	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
$\tilde{F}_{y,f}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
$\tilde{\theta}_r$	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0
$\tilde{\phi}_r$	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	1	0

time uniformly sampled within the 0–10 second window. Fault magnitudes were also randomly generated based on the specified fault ranges in Table 7.3, where sensor/actuator faults were modeled as either zero-mean Gaussian noise or constant bias errors with the same variance. External disturbances and parametric faults were simulated by deviating from nominal values within defined thresholds. This procedure ensured coverage of a wide range of fault manifestations while preserving a realistic and diverse fault scenario distribution. The collected dataset consists of 245,142 samples, with 61% corresponding to healthy conditions (label 0). Each of the 18 fault types accounts for approximately 1–3% of the total samples, offering a representative and reasonably balanced distribution across all fault classes.

To evaluate the classification performance of the proposed and baseline methods, two widely-adopted metrics are used: accuracy and F1-score. These metrics are computed based

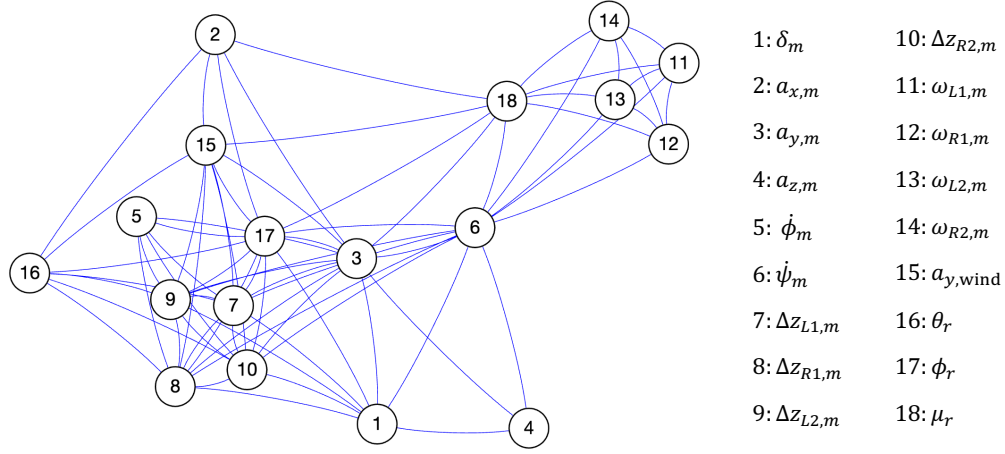


Figure 7.14: Fault interaction graph (FIG) for the vehicle system, obtained via one-mode projection of the bipartite fault-signature graph (BFSG). Each node corresponds to one of the 18 fault sources, and edges indicate fault interactions arising from shared influences on residuals.

on the classification outcomes—true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Accuracy quantifies the overall correctness of the model by measuring the proportion of all correctly classified instances among the total number of instances:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}.$$

F1-score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives. This score is particularly useful when the dataset is imbalanced, as it reflects the trade-off between the precision and recall:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Table 7.5 represents the classification accuracy of local supervised classifiers and unsupervised autoencoders for each individual fault source. Each local model is trained using only the data associated with its corresponding fault. For supervised models, only labeled faulty samples of the target fault and a balanced subset of healthy samples are used, with 80% allocated for training and 20% for testing. In contrast, the unsupervised autoencoders

Table 7.5: Accuracy of local fault detectors at each node. Supervised classifiers are trained on labeled faulty data, while unsupervised autoencoders are trained on healthy data only.

Method	Fault Sources								
	δ_m	$a_{x,m}$	$a_{y,m}$	$a_{z,m}$	ϕ_m	ψ_m	$\Delta z_{fl,m}$	$\Delta z_{fr,m}$	$\Delta z_{rl,m}$
Supervised	98.5%	98.5%	99.3%	99.2%	99.9%	99.5%	99.6%	99.5%	99.6%
Unsupervised	85.7%	90.5%	91.4%	93.5%	90.2%	93.0%	89.7%	88.5%	93.6%

Method	Fault Sources								
	$\Delta z_{rr,m}$	$\omega_{fl,m}$	$\omega_{fr,m}$	$\omega_{rl,m}$	$\omega_{rr,m}$	$a_{y,\text{wind}}$	θ_r	ϕ_r	μ_r
Supervised	99.3%	99.7%	99.8%	99.6%	99.9%	99.7%	90.3%	97.8%	98.8%
Unsupervised	88.9%	92.4%	94.8%	97.4%	96.7%	62.7%	88.4%	90.7%	60.6%

are trained solely on healthy data using a random 10% subset due to the large volume of healthy samples. All models receive input as sliding windows of 40 time steps.

Figure 7.15 illustrates an example of a local unsupervised model trained for fault detection of the $a_{y,m}$ measurement. The reconstruction error histogram and confusion matrix show how the threshold is selected to balance detection accuracy between healthy and faulty classes. This example is representative of the process applied to all 18 fault sources in the unsupervised approach, with overall detection performance summarized in Table 7.5.

On average, the supervised classifiers outperform their unsupervised counterparts by approximately 10.42%. The average accuracy across all faults is 98.68% for the supervised models and 88.26% for the unsupervised models. Significant improvements are observed in detecting faults with subtle residual patterns, such as road friction coefficient (μ_r) and lateral wind disturbance ($a_{y,\text{wind}}$), where the accuracy improves from 60.6% to 98.8% and from 62.7% to 99.7%, respectively. These results confirm the effectiveness of fault-specific supervised training, especially for faults with weak or ambiguous signatures under healthy-only training. However, in large-scale systems where labeled faulty data may not be available for all fault scenarios, the unsupervised autoencoders remain a promising alternative, achieving an average accuracy of 88.26%.

7.4.3 Consensus-Based Fault Isolation

The consensus-based fault isolation strategy proposed in Section 5.2.5 is evaluated here. This approach leverages the structure of the FIG, which is derived from the system's structural information and the refined fault signature matrix introduced earlier. Each

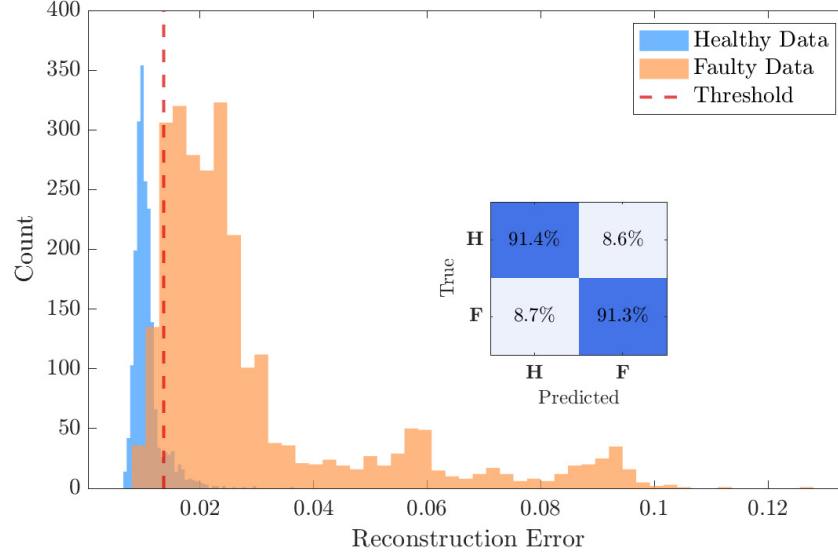


Figure 7.15: Reconstruction error histogram for fault detection of $a_{y,m}$ using an autoencoder trained on healthy data. The threshold is set to balance detection accuracy, as shown in the confusion matrix.

node in the FIG is equipped with either a supervised classifier (for labeled nodes) or an unsupervised autoencoder (for unlabeled nodes), both of which output local fault scores. The core idea is to aggregate local scores through a graph-based consensus mechanism, enabling global fault isolation even when labeled data is unavailable at some nodes. The final decision is made by selecting the node with the highest consensus score, subject to a threshold check.

Figure 7.16 presents the confusion matrix for consensus-based fault isolation when all nodes in the FIG are labeled—that is, each node is equipped with a supervised classifier trained on both healthy and faulty labeled data. The healthy (fault-free) condition is encoded as ‘H’. The results demonstrate strong isolation performance, with an overall accuracy of 93.6% across all fault types. Most faults are correctly identified with high confidence, as indicated by the strong diagonal entries in the matrix.

This result highlights the effectiveness of the proposed consensus-based global decision-making strategy. Although each local detector is trained only on its own fault and neighboring fault data, the consensus mechanism leverages the structural information embedded in the FIG to infer the correct global fault label. This localized training paradigm makes the method scalable, as new fault nodes can be added without retraining

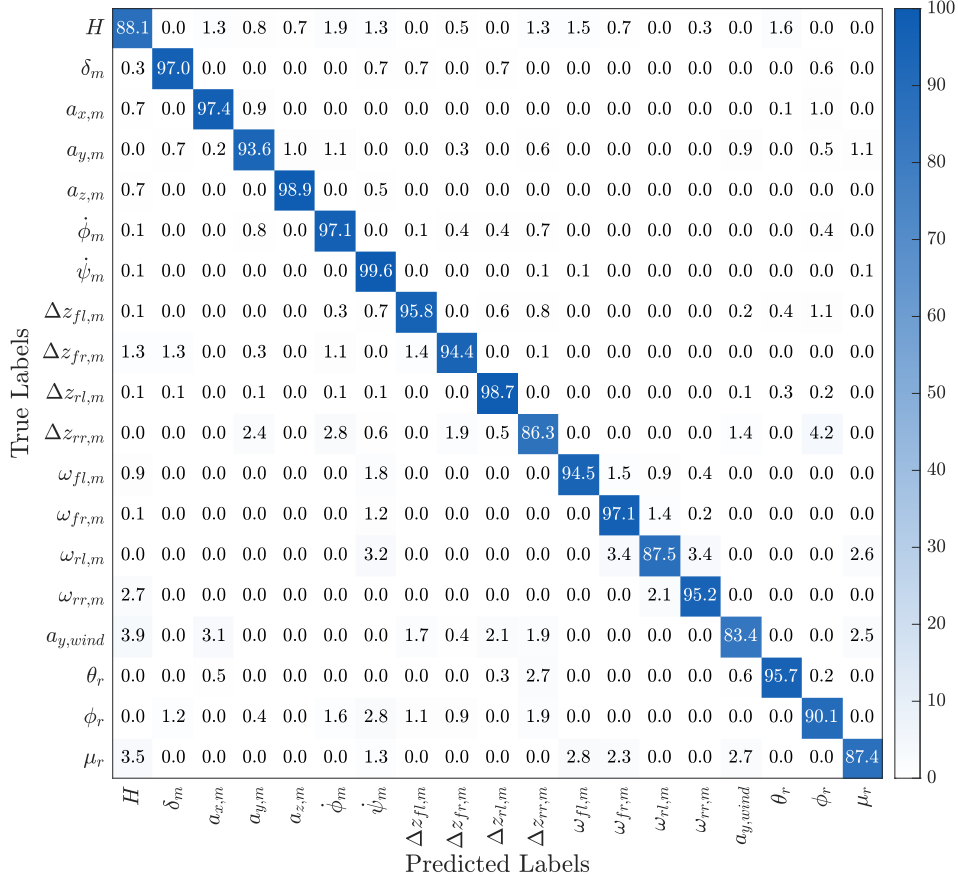


Figure 7.16: Confusion matrix for fault isolation with all nodes labeled.

the entire network. The strong performance in this fully labeled scenario confirms the reliability and robustness of the approach when supervision is available at all nodes.

Figure 7.17 presents the confusion matrix for fault isolation when all nodes in the FIG are unlabeled—i.e., each node is equipped with an unsupervised autoencoder trained only on healthy data. As shown, the overall isolation performance is significantly degraded, with widespread misclassification across fault types. This result is expected given the absence of supervision and the use of a consensus mechanism that aggregates unsupervised anomaly scores from neighboring nodes. Since faults can indirectly affect the features of nearby nodes, these nodes may exhibit elevated anomaly scores even if they are not the true fault source. As a result, the consensus-based decision can incorrectly attribute the fault to

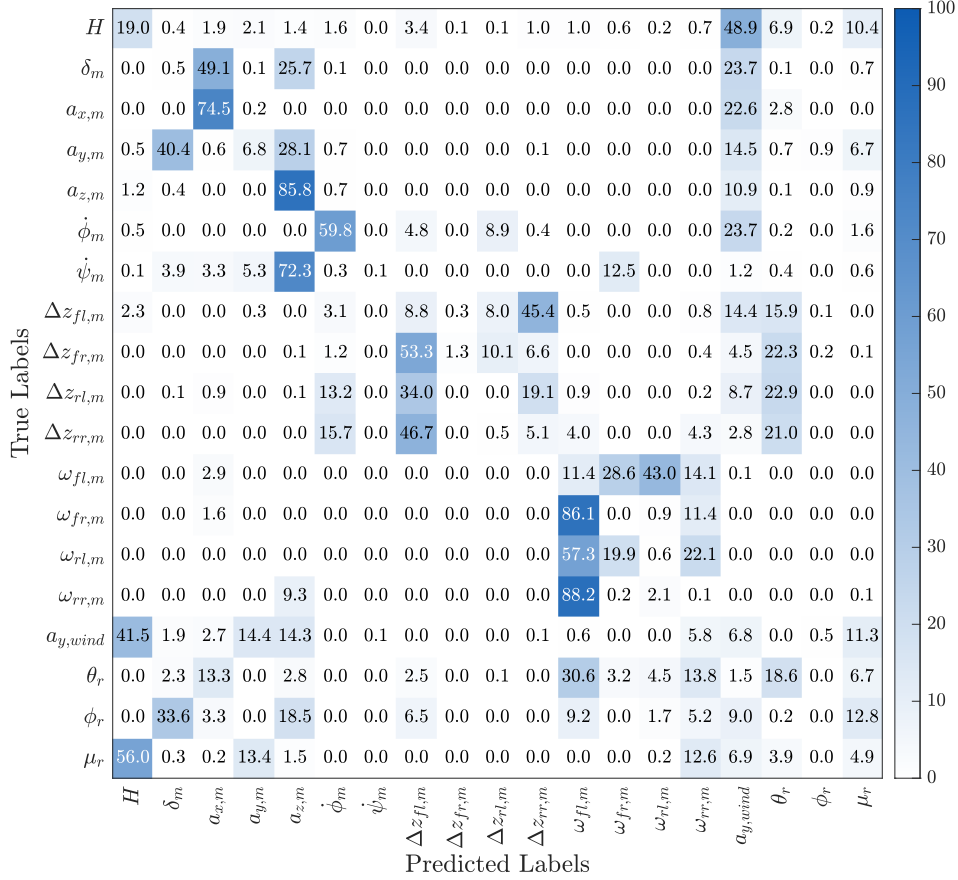


Figure 7.17: Confusion matrix for fault isolation with all nodes unlabeled.

multiple candidates, especially in regions with strong topological coupling.

This behavior is explained by the consensus strategy defined in (5.13), which aggregates the raw anomaly scores from neighboring nodes when a node is unsupervised. Without access to labeled fault data, each node cannot distinguish between a true local anomaly and one indirectly induced by a neighbor's fault. Therefore, the consensus propagation mechanism amplifies uncertainty and leads to confusion between structurally adjacent faults. While this highlights the limitation of relying solely on unsupervised local detectors, it also underscores the value of structural reasoning in fault inference. Despite the degraded performance in this extreme case, the unified propagation framework still enables some level of inference under label-scarce conditions.

Impact of Labeled Neighbors: Synthetic Example

To provide intuition about the impact of labeled neighbors on global fault localization in the proposed consensus-based framework, a synthetic example is considered, as illustrated in Figure 7.18. A fault is assumed to occur at node1, which is treated as an *unlabeled* node. Two scenarios are analyzed: (a) all neighbors of node1 are also unlabeled, and (b) all neighbors of node 1 are labeled. For each case, two sub-scenarios are examined depending on whether the neighboring nodes produce correct or incorrect outputs.

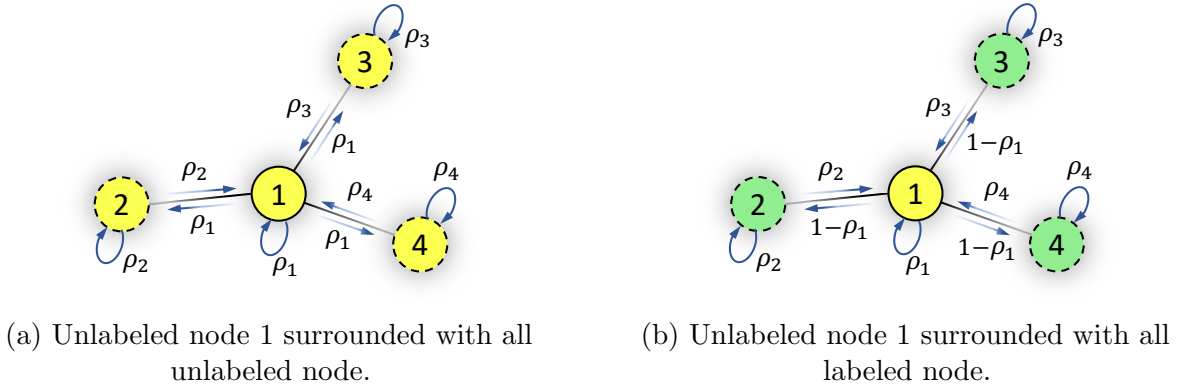


Figure 7.18: Illustration of two cases of consensus for global fault localization.

Case 1: All nodes are unlabeled.

Node 1 is assumed to be faulty and is assigned a high anomaly score of $\rho_1 = 0.9$. The neighboring unlabeled nodes may either:

- Scenario 1: correctly remain unaffected and output low scores: $\rho_2 = \rho_3 = \rho_4 = 0.1$, or
- Scenario 2: falsely respond to node 1's fault and output high scores: $\rho_2 = \rho_3 = \rho_4 = 0.9$.

The consensus scores, calculated using (5.13), are summarized in Table 7.6. In both scenarios, the neighboring nodes exhibit consensus scores that are equal to or greater than that of node 1, despite the fault being localized only at node 1. This mis-classification results from the fact that all nodes are unsupervised and directly propagate potentially misleading anomaly scores.

Table 7.6: Consensus scores when all nodes are unlabeled.

Scenario	Consensus score of node 1	Consensus score of neighbors
Scenario 1: $\rho_{2,3,4} = 0.1$	$0.3 = \frac{1}{4}(0.9 + 0.1 + 0.1 + 0.1)$	$0.5 = \frac{1}{2}(0.9 + 0.1)$
Scenario 2: $\rho_{2,3,4} = 0.9$	$0.9 = \frac{1}{4}(0.9 + 0.9 + 0.9 + 0.9)$	$0.9 = \frac{1}{2}(0.9 + 0.9)$

Case 2: Node 1 is unlabeled and neighbors are labeled.

Once again, node 1 is assumed to be faulty and assigned an output of $\rho_1 = 0.9$. The labeled neighboring nodes may either:

- Scenario 1: correctly output low fault scores: $\rho_2 = \rho_3 = \rho_4 = 0.1$, or
- Scenario 2: incorrectly produce high scores due to indirect influence: $\rho_2 = \rho_3 = \rho_4 = 0.9$.

The consensus scores are calculated in Table 7.7. Recall that for labeled neighbors, the transformation $\phi_i(\rho_j) = 1 - \rho_j$ is applied before contributing to the consensus of the unlabeled node. In contrast to case 1, the consensus scores for node 1 remain higher than those of its labeled neighbors in both scenarios. This results in correct fault localization, even when the labeled nodes are indirectly influenced and produce high local scores.

Table 7.7: Consensus scores when node 1 is unlabeled and neighbors are labeled.

Scenario	Consensus score of node 1	Consensus score of neighbors
Scenario 1: $\rho_{2,3,4} = 0.1$	$0.3 = \frac{1}{4}(0.9 + 0.1 + 0.1 + 0.1)$	$0.1 = \frac{1}{2}(0.1 + (1 - 0.9))$
Scenario 2: $\rho_{2,3,4} = 0.9$	$0.9 = \frac{1}{4}(0.9 + 0.9 + 0.9 + 0.9)$	$0.5 = \frac{1}{2}(0.9 + (1 - 0.9))$

This example illustrates that the inclusion of labeled nodes in the neighborhood of an unlabeled node can enhance the reliability of consensus-based decision-making. In case 1, where all nodes are unsupervised, the resulting consensus scores are ambiguous and may lead to incorrect fault localization, as the influence of neighboring anomalies cannot be effectively mitigated. In contrast, case 2 demonstrates that labeled neighbors—through the transformation $\phi_i(\cdot)$ —serve as informative references that suppress false consensus in non-faulty regions and reinforce the correct identification of the true fault source. Even when labeled nodes are indirectly affected and produce elevated scores, the consensus mechanism preserves a clear distinction between the faulty node and its neighbors. This behavior underscores a key strength of the proposed consensus algorithm: by incorporating supervision through the transformation function, more accurate global decisions can be achieved under local uncertainty.

This observation motivates the next evaluation, which aims to quantify how both the number and topological arrangement of labeled nodes affect global fault localization accuracy within the FIG. Intuitively, if labeled nodes are positioned such that they cover or influence a larger portion of the unlabeled nodes, the overall accuracy of the consensus-based fault isolation is expected to surpass that of arbitrary or sparsely connected labeling configurations with the same number of labeled nodes. To formalize this notion of coverage and influence, the concept of the k -Tuple dominating set (k -TDS), defined in Definition 10, is employed. This graph-theoretic tool ensures that each unlabeled node is connected to at least k labeled nodes, thereby promoting redundancy and improving the accuracy of the consensus mechanism in partially labeled scenarios.

For the fault interaction graph (FIG) shown in Figure 7.14, different configurations of labeled and unlabeled nodes based on the k -tuple dominating sets (k -TDS) of labeled nodes are illustrated in Figure 7.19. For each value of k , there may exist multiple k -TDS configurations; however, only one example of a minimum-size k -TDS is selected for analysis. For instance, a minimum-size 1-TDS of labeled nodes may consist of nodes $\{6, 17\}$, ensuring that every unlabeled node (highlighted in yellow) is connected to at least one labeled node.

Figure 7.20 presents the average number of labeled and unlabeled neighbors per unlabeled node as topological metrics for each configuration. As expected, the average number of *labeled neighbors per unlabeled node* increases with increasing k , reflecting greater influence from the labeled set. Conversely, the average number of *unlabeled neighbors per unlabeled node* generally decreases with higher k , although minor increases can occur (e.g., from $k = 5$ to $k = 7$). This overall trend aligns with the intuition that when unlabeled nodes are less interconnected and more exposed to labeled nodes, classification performance improves—reducing the chance of misclassification among tightly clustered unlabeled nodes.

To jointly capture this effect, the *coverage ratio* of unlabeled nodes is defined as:

$$\text{CR} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathcal{N}_L(u)|}{|\mathcal{N}_L(u)| + |\mathcal{N}_U(u)|} \quad (7.2)$$

where $\mathcal{U}$ denotes the set of unlabeled nodes, and $\mathcal{N}_L(u)$ and $\mathcal{N}_U(u)$ represent the labeled and unlabeled neighbors of node u , respectively. The coverage ratio increases when unlabeled nodes are more connected to labeled nodes and less to one another, and serves as a useful topological indicator of the potential for accurate fault isolation.

Figure 7.21 illustrates the relationship between classification performance and the coverage ratio under different labeling configurations in the FIG. The bar plots show the fault isolation accuracy and precision for each configuration, while the orange line represents the corresponding coverage ratio of unlabeled nodes, as defined in (7.2).

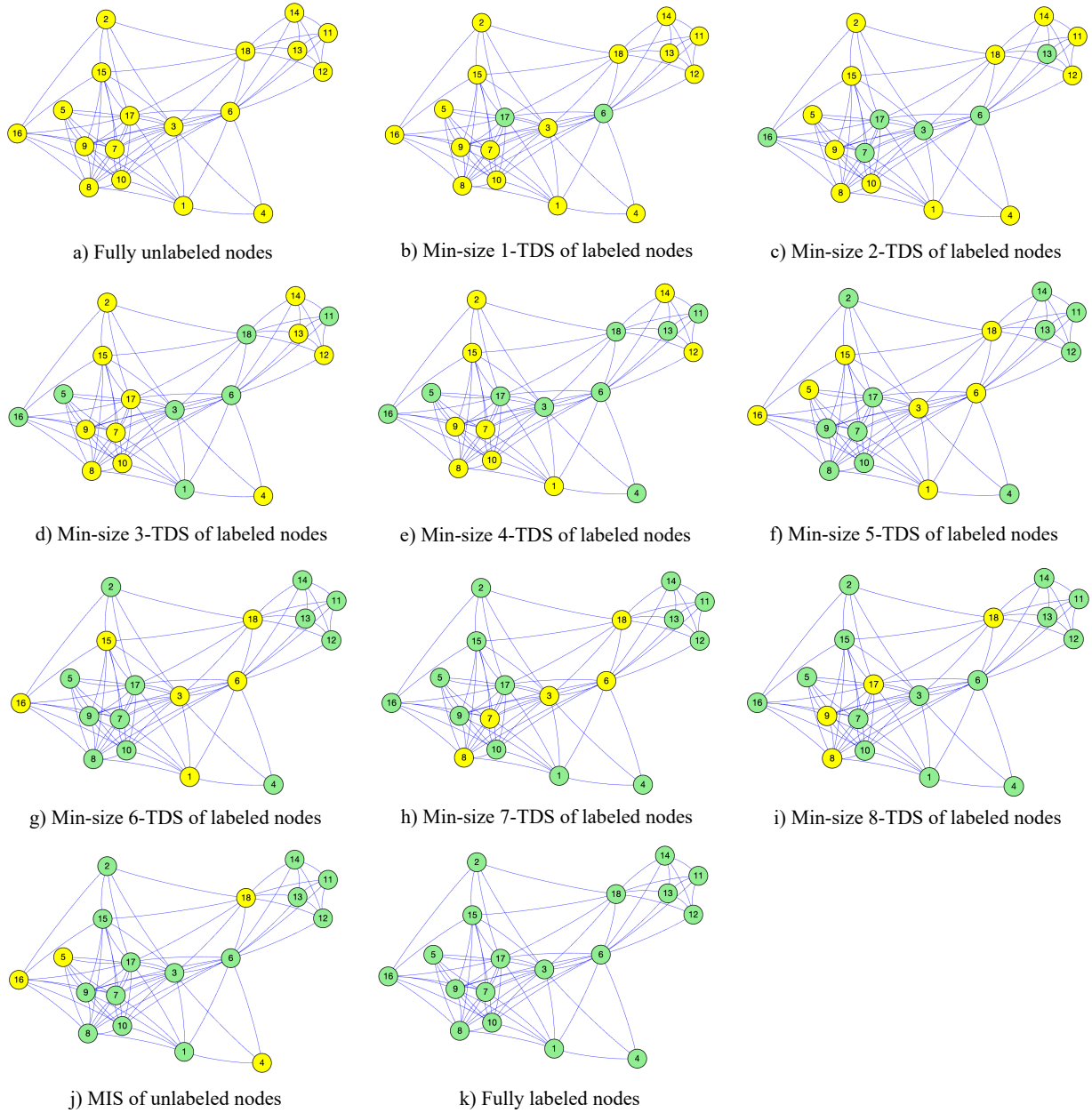


Figure 7.19: Different topologies of the Fault Interaction Graph (FIG), each exhibiting distinct fault isolation performance based on connectivity patterns.

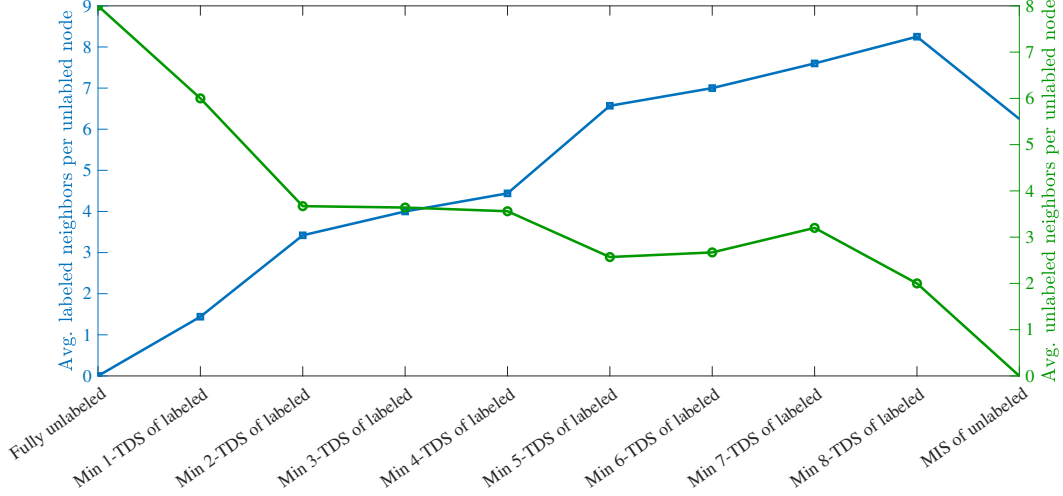


Figure 7.20: Average number of labeled and unlabeled neighbors per unlabeled node under different labeling configurations. As k increases, unlabeled nodes are more exposed to labeled ones ($L \rightarrow U$) and less connected to each other ($U \rightarrow U$).

As seen in the figure, there is a clear and consistent trend: configurations with higher coverage ratios exhibit better classification performance. When the labeled nodes are selected based on minimum-size k -TDS (for increasing values of k), the coverage ratio steadily increases, indicating greater exposure of unlabeled nodes to labeled information. This increase in coverage corresponds closely with improved accuracy and precision across the configurations.

Notably, when the coverage ratio is low—such as in the fully unlabeled case or small k -TDS configurations—both accuracy and precision are significantly reduced. As the coverage ratio increases (e.g., from 1-TDS to 6-TDS of labeled nodes), a substantial improvement in classification accuracy is observed. This trend supports the interpretation of the coverage ratio as a meaningful topological metric for predicting classification performance. Moreover, sharp improvements in accuracy are aligned with steep increases in the coverage ratio. For example, noticeable jumps in accuracy occur when transitioning from 1-TDS to 2-TDS and again from 7-TDS to 8-TDS of labeled nodes. These transitions mark structural shifts where unlabeled nodes become more strongly connected to labeled nodes, enhancing their isolation and reducing potential misclassification. The highest performance is observed in the fully labeled configuration, where all nodes are labeled.

An interesting observation arises when comparing the 8-TDS of labeled nodes and the

maximum independent set (MIS) of unlabeled nodes. Despite the MIS configuration having four unlabeled nodes, its performance is remarkably close to the fully labeled case. This highlights the power of leveraging structural knowledge inherent in the system model: even when some labeled data are unavailable, carefully selecting unlabeled nodes based on their topological roles—such as ensuring they are disconnected from one another—can still yield high fault isolation accuracy.

As shown in Figure 7.19, both the 8-TDS and MIS configurations include exactly four unlabeled nodes. At first glance, one might expect the 8-TDS to outperform MIS, as the average number of labeled neighbors per unlabeled node is higher (8.25 for 8-TDS vs. 6.25 for MIS). However, this metric alone is insufficient. The more informative metric is the coverage ratio, which also accounts for how unlabeled nodes are connected among themselves. In the MIS configuration, the unlabeled nodes are fully disconnected, eliminating the possibility of fault propagation or false positives between them. This structural advantage is reflected in the results: the MIS configuration achieves approximately 6% higher accuracy than the 8-TDS case, consistent with its higher coverage ratio.

If labeled data is available for all faults, the structural information inherent in the FIG can be effectively exploited using a graph convolutional network (GCN), as discussed in the following section.

7.4.4 Graph Convolutional Network for Fault Isolation

Fault isolation is performed at two levels by leveraging the structural knowledge encoded in the FIG. This distributed architecture first employs lightweight binary local detectors, each trained in a supervised manner for a specific fault source node, as detailed in Section 5.2.5. The predicted fault probabilities from these local detectors are then used as node features in a GCN-based network, described in Section 5.2.6, which serves as a global classifier by capturing spatial dependencies via GCN layer and temporal patterns via RNN layer to isolate the fault source; see Figure 5.13.

To better demonstrate the performance improvement of the proposed distributed method, it is compared against a centralized classifier that performs fault isolation directly using all available residuals (features). The architecture of this centralized approach is shown in Figure 5.5. Unlike the distributed method, it assumes full access to all features and does not explicitly leverage the structural connectivity among fault sources.

To compare the proposed distributed GCN-based architecture and the centralized architecture, both accuracy and training efficiency—in terms of the number of epochs to convergence—are evaluated. The number of epochs required for each model is determined

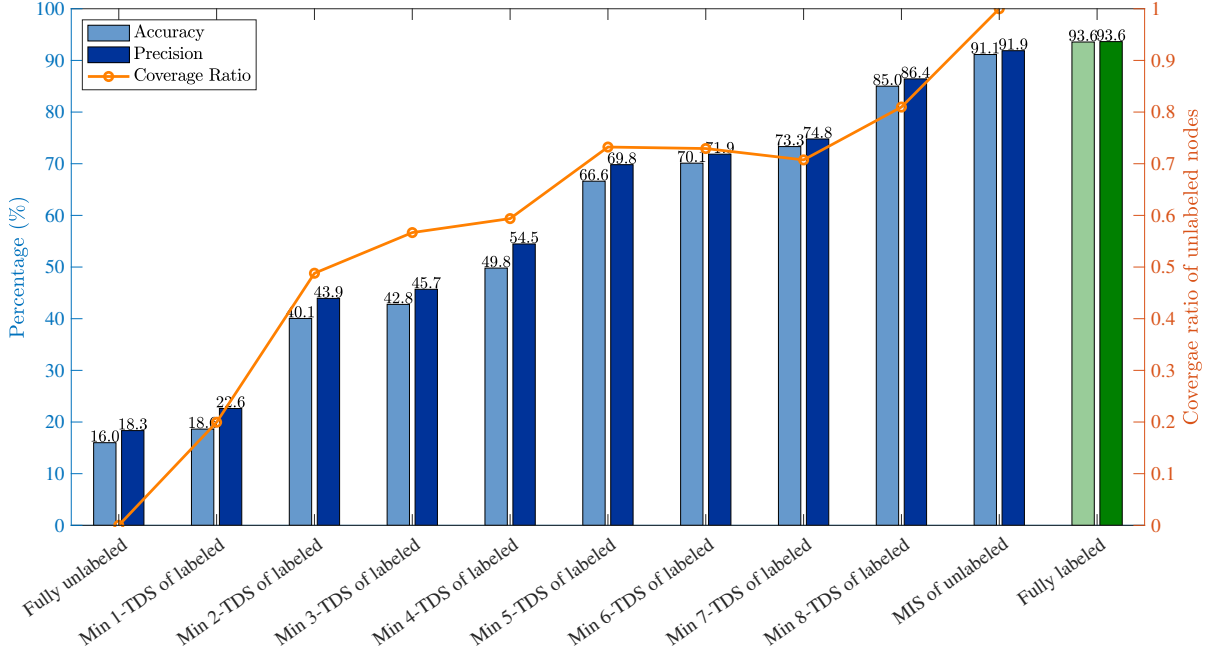


Figure 7.21: Classification accuracy, precision, and coverage ratio of unlabeled nodes under different FIG topologies. Bar plots show the performance metrics (left y-axis), while the orange line shows the coverage ratio (right y-axis). Increased coverage corresponds to improved performance across configurations.

by monitoring the validation accuracy across training epochs. The point at which no further improvement is observed is identified, and the epoch immediately prior to this plateau is selected as the effective convergence point.

Figure 7.22 investigates the effect of model complexity, represented by the number of trainable parameters. As expected, the accuracy of both models improves with increasing model capacity; however, the distributed architecture consistently outperforms the centralized one, particularly at lower parameter budgets. Notably, the distributed method achieves near-maximum accuracy with fewer than 1200 parameters—thanks to its modular design consisting of lightweight local detectors (with a total of only 648 parameters) and a compact global classifier—whereas the centralized model requires over 3000 parameters to reach similar performance. This highlights the efficiency gained by leveraging structural information, where the prior knowledge encoded in the adjacency matrix can effectively guide the training process, reducing the need for excessive model complexity to achieve

high isolation performance. Furthermore, the number of training epochs required for the centralized model increases with model size, consistent with the general expectation that more complex models require longer training. In contrast, the distributed model exhibits a decreasing trend in required epochs as its capacity grows. This counterintuitive behavior suggests that the structural guidance provided by the adjacency matrix in the GCN accelerates convergence: as more trainable parameters become available, the expressiveness of the GCN increases, enabling the structural prior (i.e., fault connectivity) to guide learning more effectively and reduce the number of training iterations needed.

Figure 7.23 evaluates the impact of training data size. It shows that while both architectures experience reduced accuracy with less data, the distributed method maintains high performance even at lower data sizes, whereas the centralized classifier suffers a sharp drop. This highlights the advantage of incorporating structural knowledge, which helps compensate for limited data availability and improves generalization by embedding system-level dependencies into the learning process. More importantly, the centralized model requires a constant number of epochs (320) regardless of the data size, indicating limited adaptability to the volume of data. In contrast, the distributed model benefits from increased data with significantly fewer training epochs needed. This again highlights the advantage of structural knowledge incorporated through the GCN layer. As more labeled data becomes available, the information encoded in the adjacency matrix becomes increasingly effective, helping the model converge faster with fewer iterations.

These results demonstrate that the proposed distributed hybrid approach, which combines structural knowledge from the fault interaction graph (FIG) with data via a GCN layer, significantly improves both fault isolation accuracy and training efficiency. By embedding system-level dependencies, the method achieves higher accuracy with fewer parameters and training epochs, especially under limited data. Its modular architecture enables scalable deployment—when new fault sources are introduced, only the corresponding local detector and global classifier (with an updated adjacency matrix) need retraining, avoiding the costly full-model retraining required in centralized methods. This modular training strategy is particularly advantageous in large-scale systems with hundreds of fault sources, where full retraining would be computationally prohibitive.

Table 7.8 compares the performance and complexity of various fault isolation methods. The Fault Signature Matrix (FSM), which uses a threshold-based residual analysis guided by a predefined fault signature matrix and relies on 25 threshold parameters—one for each residual—yields the lowest accuracy (74.83%). This is expected, as residuals may deviate from ideal zero-mean behavior due to model imperfections or noise, resulting in false positives. The centralized CNN+RNN model achieves high accuracy (97.40%) by effectively learning spatial-temporal dependencies; however, it requires a large number of

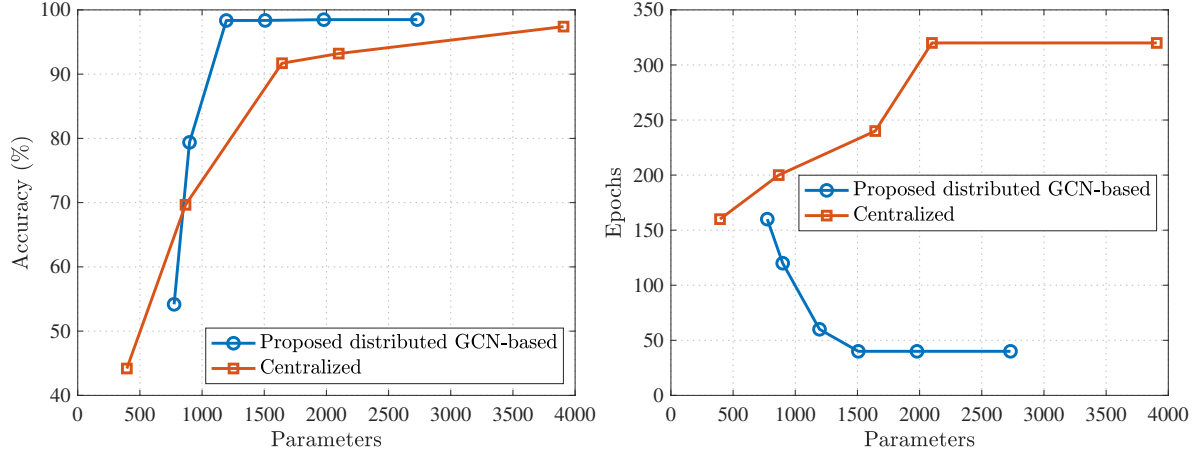


Figure 7.22: Comparison of the proposed distributed GCN-based architecture and the centralized architecture in terms of fault isolation accuracy and training efficiency (number of epochs to convergence) versus the number of model parameters.

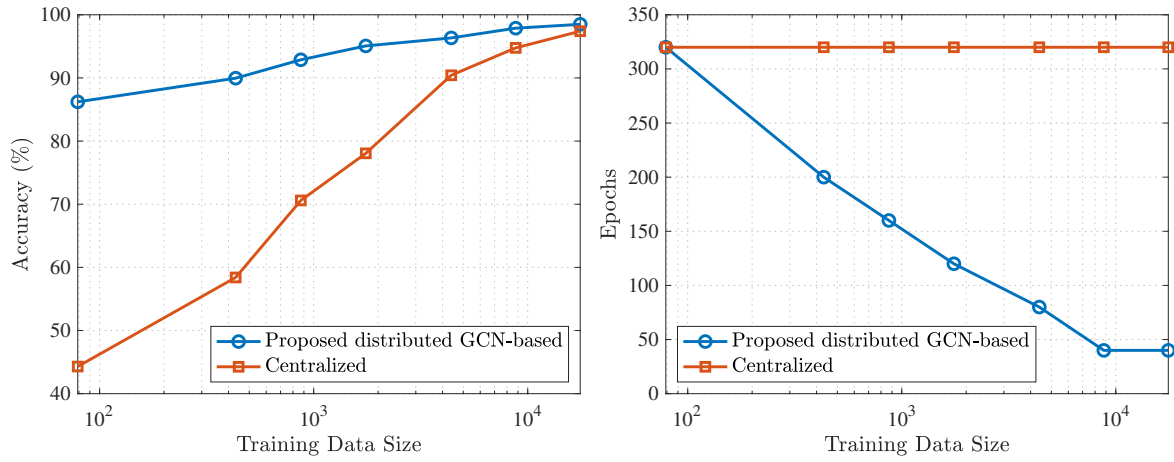


Figure 7.23: Comparison of the proposed distributed GCN-based architecture and the centralized architecture in terms of fault isolation accuracy and training efficiency (number of epochs to convergence) versus the training data size.

trainable parameters (3907) and sufficient labeled data across all fault classes. In contrast, the proposed hybrid distributed method—comprising lightweight local detectors and a GCN-based global classifier—achieves the highest accuracy (98.35%) using only 1515 parameters, which is less than half the parameter count of the centralized model. This highlights the effectiveness of leveraging structural knowledge from the Fault Interaction Graph (FIG) to guide learning and reduce model complexity. The consensus-based distributed method achieves competitive accuracy (93.60%) and operates without additional parameters in the global stage, relying solely on the 648 trainable parameters of the local detectors. Importantly, it supports partial supervision and, as discussed in Section 7.4.3, remains effective even when some fault nodes lack labeled data, provided their corresponding nodes are not structurally adjacent in the FIG. These results collectively highlight the advantages of the proposed distributed and structure-informed method, which offers a scalable, interpretable, and data-efficient solution for fault isolation in large-scale systems, outperforming centralized approaches in both accuracy and practicality.

Table 7.8: Comparison of fault isolation methods in terms of classification accuracy and model complexity.

Method	Accuracy	Trainable Parameters
Fault Signature Matrix (FSM)	74.83 %	25
Centralized (CNN + RNN)	97.40 %	3907
Distributed (Local Detectors + GCN-based Global Classifier)	98.35 %	1515
Distributed (Consensus-based)	93.60 %	648

7.5 Summary

This chapter presents the results of the multi-stage hybrid FDI framework developed in Chapter 5, applied to a vehicle system modeled as a networked dynamical system using an estimation graph. The proposed framework integrates model-based techniques with data-driven neural network methods to enhance accuracy, robustness, and efficiency. In the detection stage, a hybrid approach combining the residual-based χ^2 -test with a neural network classifier through voting logic achieves a fault detection accuracy of 99.6%, significantly reducing false positives compared to standalone methods. For fault isolation, model-based likelihood information—quantified analytically through the graph structure—is fused with historical data and processed by an RNN-based classifier, resulting in improved

isolation performance. The hybrid method achieves 99.5% isolation accuracy, outperforming both model-only and data-only approaches, especially in data-limited scenarios or cases with similar residual signature patterns. Finally, model-based fault magnitude identification is performed by analyzing fault-residual relationships within the estimation graph, enabling accurate estimation even for challenging cases such as the $a_{z,m}$ bias, which becomes identifiable through its effect on the residuals of the descendant node (E_5) in the estimation graph.

In the second part of the chapter, two distributed fault isolation methods—consensus-based and GCN-based—are evaluated within the proposed hybrid framework. Unlike traditional model-based techniques, this hybrid distributed approach relies solely on structural information extracted from the system model, without requiring detailed model parameters. This makes it particularly valuable for large-scale systems, where only the structural connectivity can be readily obtained via structural analysis. A fault signature matrix (FSM) is first constructed using structural analysis and then refined through correlation and sensitivity analysis between residuals and fault sources, resulting in a sparser and more informative representation. A novel application of one-mode projection is then applied to the bipartite fault signature graph (BFSG), yielding the fault interaction graph (FIG) that captures the underlying interdependencies among fault sources. This FIG serves as the basis for the two distributed isolation strategies. The consensus-based method demonstrates competitive performance with minimal trainable parameters and supports partial supervision. Meanwhile, the GCN-based method integrates the structural information encoded in the FIG with learned representations, outperforming the centralized model in both accuracy and training efficiency. Importantly, the distributed strategies are modular and scalable: when new fault sources are added, only the corresponding local detector and the consensus algorithm or the global classifier (with an updated adjacency matrix) require retraining—unlike centralized methods that demand full retraining. This structure-informed modularity makes the proposed approach highly practical and efficient for fault isolation in complex, large-scale systems.

Chapter 8

Optimal Topology in Estimation Graph

To ensure accurate and reliable state estimation, a 3-stage framework as shown in Figure 8.1 is proposed:

1. **Fault detection and isolation:** In the first stage, the fault is detected and isolated using the residual-based FDI methods presented in Sections 5.1.1 and 5.2.1.
2. **Real-time uncertainty quantification:** For the isolated fault source, the analytical or surrogate-based models introduced in Chapter 4 are used to compute the real-time covariance matrix of the estimated states, quantifying the uncertainty along each estimation path.
3. **Reconfiguration to the optimal path:** The uncertainty measure is evaluated across the available paths, and the path with the highest reliability index is selected as the most reliable estimation path.

To evaluate the effectiveness of the proposed systematic framework for real-time reconfiguration of state estimation toward the most reliable path under varying operating conditions, two distinct estimation paths are considered for estimating the vehicle's lateral velocity, as shown in Figure 8.2. A cornering maneuver is performed at a constant longitudinal speed of 70 km/h on a dry road surface with a friction coefficient of $\mu_r = 1$. Figure 6.13 illustrates the time histories of the steering angle, roll angle, yaw rate, and lateral acceleration—key variables characterizing the vehicle's lateral dynamics. Three sensor faults—biases in $a_{y,m}$,

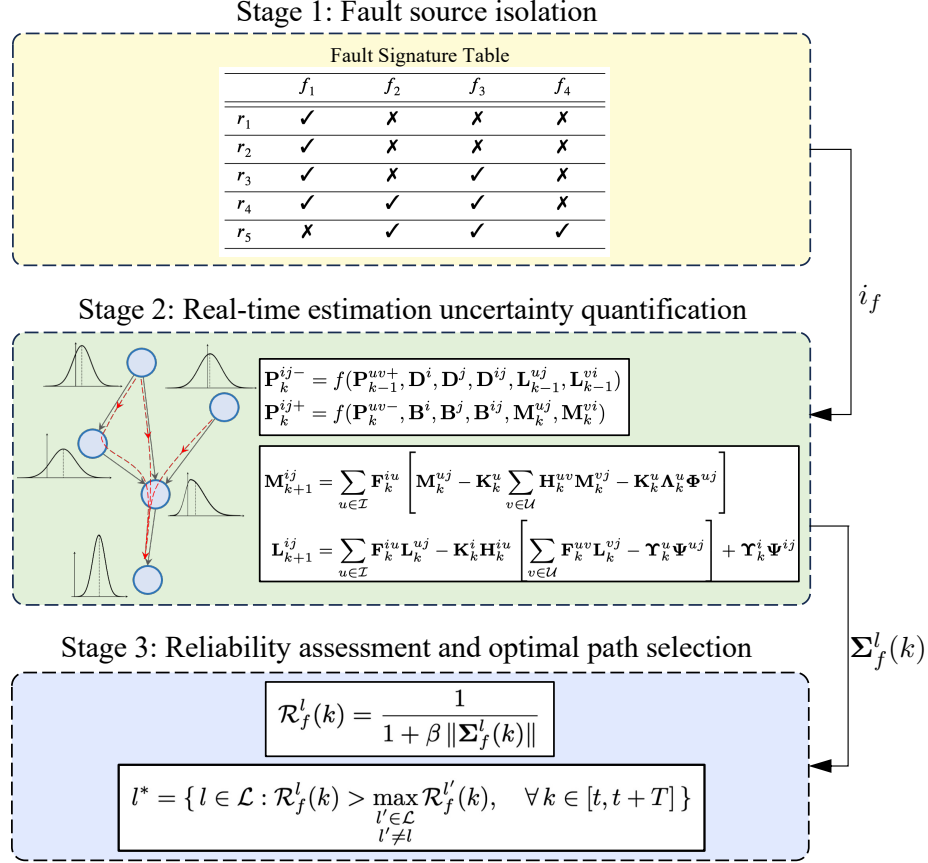


Figure 8.1: Structured framework for fault isolation and reliability assessment of multiple estimation paths under diverse fault scenarios to enable optimal path selection.

$a_{z,m}$, and $\dot{\psi}_m$ —along with an external disturbance from lateral wind gusts, denoted as $a_{y,\text{wind}}$, are considered in this analysis. These fault sources are $a_{y,m}$, $\dot{\psi}_m$, $a_{z,m}$, and $a_{y,\text{wind}}$, respectively. The evaluation is structured into three stages: (1) fault source isolation, (2) real-time uncertainty quantification, and (3) reconfiguration to the optimal estimation path. Each stage of the framework is detailed in the following sections.

8.1 Fault Detection and Isolation (Stage 1)

Residual-based methods are employed in this stage to perform FDI. The residual sequences $\mathbf{r}_k$ are generated during the state estimation process and serve as the primary indicators

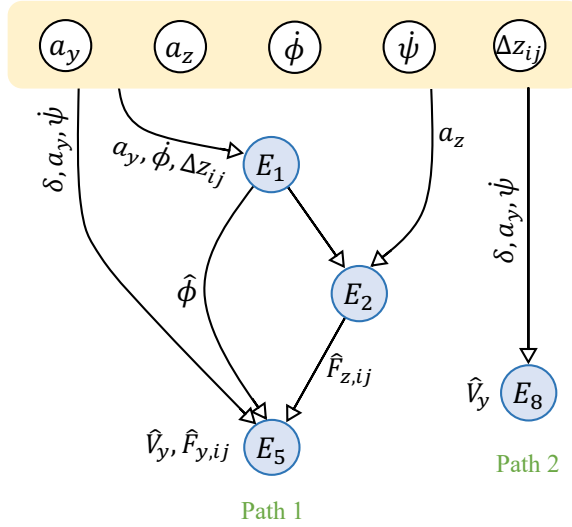


Figure 8.2: Illustration of two distinct estimation paths for the lateral velocity V_y within the vehicle estimation graph.

of system abnormalities. To facilitate isolation, a fault signature matrix is constructed, capturing the expected sensitivity of each residual to different fault sources. For fault isolation, the generated residuals are compared against this matrix. A fault f_i is considered isolated if the pattern of residuals that exceed their respective thresholds matches the corresponding column of the fault signature matrix.

By carefully analyzing the flow of information between nodes and redundant paths within the estimation graph, the precise and effective localization of fault sources is feasible. Four estimation residuals are used here to isolate the exact fault source: three measurement residuals

$$r_{ay}^{(E_1)} = a_{y,m} - \hat{a}_y^{(E_1)}, \quad r_{ay}^{(E_5)} = a_{y,m} - \hat{a}_y^{(E_5)}, \quad r_{ay}^{(E_8)} = a_{y,m} - \hat{a}_y^{(E_8)},$$

and one state residual

$$\tilde{V}_y = \hat{V}_y^{(E_5)} - \hat{V}_y^{(E_8)},$$

where the superscript E_i indicates that the variable is estimated by estimator E_i . Figure 8.3 shows the four residuals under various fault conditions. In the absence of faults (for times less than 2 seconds), all residuals are near zero mean, as expected, and stay within fault detection thresholds. In faulty scenarios—such as biases in $a_{y,m}$, $a_{z,m}$, $\dot{\psi}_m$, or constant

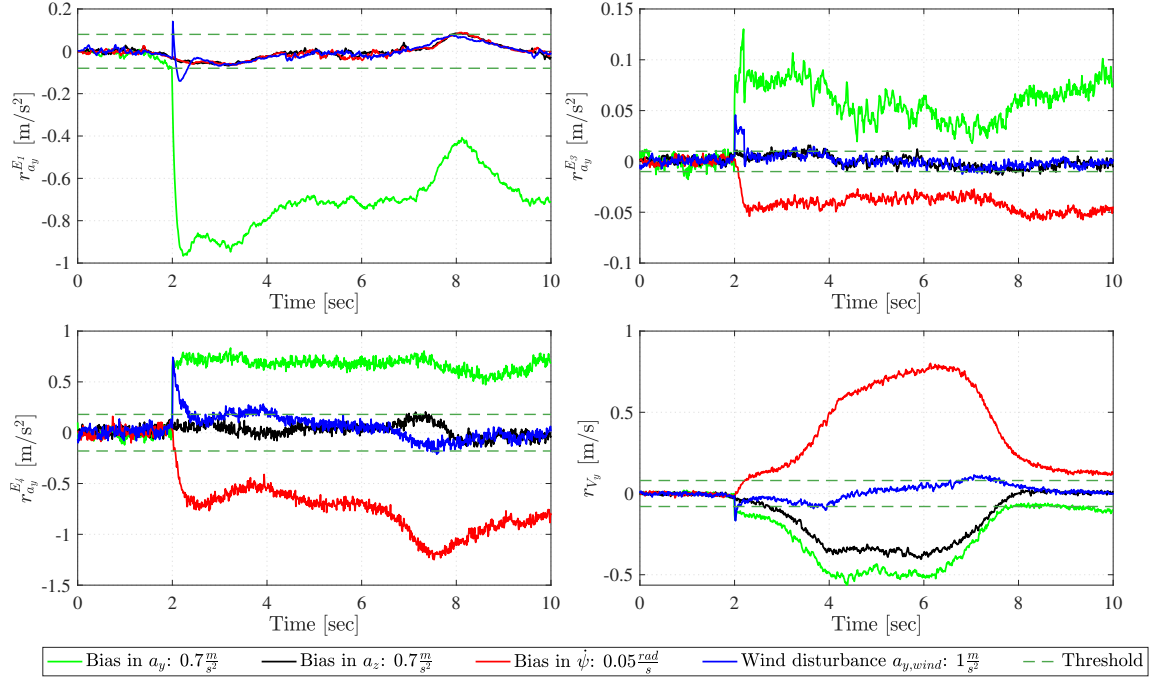


Figure 8.3: Residual patterns under different fault sources and lateral disturbance.

lateral disturbance—some residuals deviate from zero mean. To pinpoint the exact fault source, further examination is required once nonzero-mean residuals are observed.

Table 8.1 summarizes the residual signatures in the estimation graph under various fault scenarios. For instance, nonzero entries across all residuals indicate a fault in the a_y measurement, whereas a fault in the ψ measurement affects only the last three residuals, as highlighted by the red lines. During straight-line driving ($\delta = \dot{\psi} = 0$) in the time intervals 0–2 and 8–10 seconds, a fault in a_z does not satisfy the detectability condition and has negligible impact on lateral velocity estimation accuracy. However, during cornering maneuvers, it becomes easily isolable through the state residual $\tilde{V}_y$. In fact, the random-walk structure of estimator E_2 makes the a_z fault undetectable at that node. Nevertheless, structural analysis reveals that its effect propagates to the downstream estimator E_5 , where the redundant residual $\tilde{V}_y$ allows the fault to be isolated. The lateral wind gust disturbance is not detectable by the residuals alone, as its corresponding column in the fault signature matrix is entirely zero. Although the disturbance induces a sharp but short transient response in some residuals, this behavior is not persistent enough to enable reliable detection or isolation. However, since the disturbance satisfies the conditions for

Table 8.1: Fault signature matrix indicating which residuals are impacted by each fault. A residual is marked as impacted if its magnitude exceeds a predefined threshold (ϵ).

Residual	Fault source			
	$a_{y,m}$	$a_{z,m}$	$\dot{\psi}_m$	$a_{y,\text{wind}}$
$r_{a_y}^{(E_1)}$	1	0	0	0
$r_{a_y}^{(E_5)}$	1	0	1	0
$r_{a_y}^{(E_8)}$	1	0	1	0
$\tilde{V}_y \mid \delta \neq 0$	1	1	1	0

the unknown input observer (UIO), it can still be effectively isolated and identified using a UIO-based approach.

8.2 Estimation Uncertainty Quantification (Stage 2)

This stage addresses real-time UQ for the estimation path consisting of local Kalman filters, assuming fault $\mathbf{f}^i$ is identified in Stage 1. Without loss of generality, all process and output faults are assumed additive. Linearization with respect to parameters, θ^i , and estimated states, $\hat{\mathbf{x}}^{\mathcal{I}}$, can be utilized if they are introduced nonlinearly to estimator in (4.2).

The process fault $\mathbf{f}_b^i \in \mathbb{R}^p$ and output fault $\mathbf{f}_d^i \in \mathbb{R}^q$ is assumed to have the following known properties:

$$\begin{aligned} \mathbb{E}[\mathbf{f}_d^i] &= \mathbf{0} \ , \quad \mathbb{E}[\mathbf{f}_d^i \mathbf{f}_d^{j\top}] = \mathbf{D}^i \\ \mathbb{E}[\mathbf{f}_b^i] &= \mathbf{0} \ , \quad \mathbb{E}[\mathbf{f}_b^i \mathbf{f}_b^{j\top}] = \mathbf{B}^i \end{aligned} \tag{8.1}$$

Note that the cross-covariances $\mathbf{D}^{ij} = \mathbb{E}[\mathbf{f}_d^i \mathbf{f}_d^{j\top}]$ and $\mathbf{B}^{ij} = \mathbb{E}[\mathbf{f}_b^i \mathbf{f}_b^{j\top}]$ are non-zero if nodes i and j share same outputs or process faults, and $\mathbb{E}[\mathbf{f}_b^i \mathbf{f}_d^{j\top}] = \mathbf{0}$. Both excessive noise faults and bias faults are addressed similarly. For bias faults, $\mathbf{f}_b^i \in \mathbb{R}^p$ and $\mathbf{f}_d^i \in \mathbb{R}^q$ are assumed as constant random values, as documented in [54]; for excessive noise faults, they are treated as zero-mean Gaussian processes where the time-related subscript k is neglected from $\mathbf{f}_d$ and $\mathbf{f}_b$ for simplicity. However, the properties outlined in (8.1) hold true for both cases.

The proposed analytical UQ framework is not limited to white noise assumptions. The formulation remains valid for a broader class of fault models such as Gauss-Markov models—so long as the faults are zero-mean with bounded second-order moments. For instance, faults modeled by a first-order Gauss-Markov process—often used to capture

sensor bias instability in inertial measurement units (IMUs) [124]—are fully compatible with our formulation. This generality allows the proposed uncertainty propagation approach to capture a range of realistic fault behaviors encountered in practical systems, beyond idealized white noise or constant biases.

To quantify the propagation of uncertainty due to process and output faults along an estimation path, the auto-covariances and cross-covariances of all nodes within that path must be computed. In a Kalman-based estimation framework, the propagation of the *a priori* and *a posteriori* error covariances between nodes i and j caused by faults is described by (4.17) and (4.18). These covariances are computed recursively using the compact expressions:

$$\begin{aligned}\mathbf{P}_k^{ij-} &= f(\mathbf{P}_{k-1}^{uv+}, \mathbf{D}^i, \mathbf{D}^j, \mathbf{D}^{ij}, \mathbf{L}_{k-1}^{uj}, \mathbf{L}_{k-1}^{vi}), \\ \mathbf{P}_k^{ij+} &= f(\mathbf{P}_k^{uv-}, \mathbf{B}^i, \mathbf{B}^j, \mathbf{B}^{ij}, \mathbf{M}_k^{uj}, \mathbf{M}_k^{vi}),\end{aligned}\tag{8.2}$$

where $u \in \mathcal{N}^{\text{in}}(i)$ and $v \in \mathcal{N}^{\text{in}}(j)$. The recursive updates for the influence matrices $\mathbf{L}_k^{ij}$ and $\mathbf{M}_k^{ij}$ are provided by (4.19) and (4.20), respectively.

Figure 8.4 illustrates the lateral velocity estimation and its analytically quantified uncertainty for paths 1 and 2 under a bias fault in $a_{y,m}$. As the lateral velocity increases, the estimation uncertainty also grows. Between 4–7 seconds, the average standard deviation—used as the uncertainty measure defined in (8.3)—is 0.41 m/s for path 1, nearly ten times higher than that of path 2 at 0.046 m/s. This comparison shows that path 2 is more resilient to an a_y bias than path 1. Similarly, Figure 8.5 compares the impact of a bias fault in the yaw rate measurement $\dot{\psi}$ on lateral velocity estimation for the two paths. In contrast to the $a_{y,m}$ fault scenario, path 1 demonstrates significantly better resilience to a $\dot{\psi}$ bias than path 2. Specifically, during the 4–7 second interval, the average standard deviation in lateral velocity for path 1 is 0.024 m/s, which is substantially lower than path 2 at 0.43 m/s. Two key observations emerge: (1) path 2 yields more accurate lateral velocity estimates under an $a_{y,m}$ bias, while path 1 outperforms under a $\dot{\psi}$ bias; and (2) estimation accuracy aligns with the quantified uncertainty. These results demonstrate that real-time uncertainty quantification enables the selection of the optimal estimation path based on the fault isolated in Stage 1.

8.3 Optimal Estimation Path Selection (Stage 3)

Let the vector of target states to be reliably estimated in path l be denoted by $\mathbf{x}^l = (x_{l_1}, x_{l_2}, \dots, x_{l_n})$. The *uncertainty measure* for path l under fault f over the time window

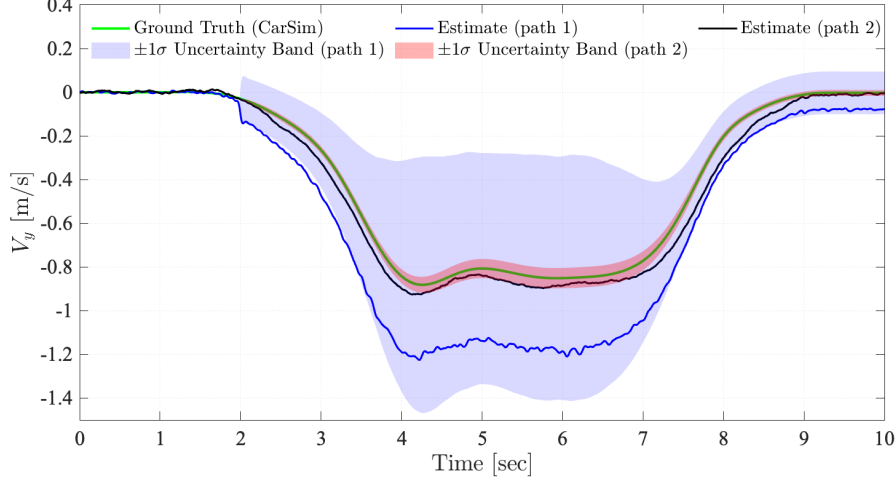


Figure 8.4: Comparing lateral velocity estimation and its quantified uncertainty for paths 1 and 2 following a bias fault onset in a_y at 2 seconds (bias = 0.5 m/s^2).

of length T is defined as

$$\|\Sigma_f^l(t)\| = \frac{1}{T} \sum_{k=t-T}^t \sqrt{w_1 p_k^{l_1 l_1 +} + w_2 p_k^{l_2 l_2 +} + \dots + w_n p_k^{l_n l_n +}}, \quad (8.3)$$

where w_i are user-defined weights, and $p_k^{l_i l_i +}$ represents the posterior variance of the estimated state $\hat{x}_{l_i}$ at time k , corresponding to the diagonal entry of the covariance matrix defined in (4.18).

Different estimation paths exhibit varying reliability under different fault scenarios. For instance, both estimation paths 1 and 2 rely on lateral acceleration a_y and yaw rate $\dot{\psi}$ measurements to estimate lateral velocity. However, they produce different levels of estimation uncertainty when faults occur in these measurements, as shown in Figures 8.4 and 8.5. The results demonstrate that quantified estimation uncertainty aligns with estimation accuracy and can effectively guide the selection of the optimal estimation path under different faults.

The uncertainty measure in (8.3) simplifies to the time-averaged standard deviation $\bar{\sigma}[\hat{V}_y]$, since lateral velocity is the sole state of interest here. The optimal path is the one with minimal sensitivity to the isolated fault. In Figure 8.6, the uncertainty measure (averaged over a 10-second cornering maneuver) for lateral velocity estimation is presented

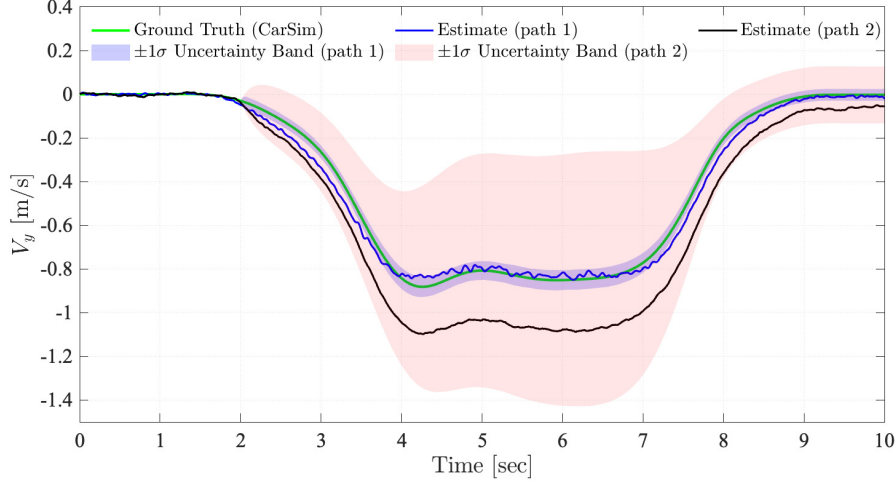


Figure 8.5: Comparing lateral velocity estimation and its quantified uncertainty for paths 1 and 2 following a bias fault onset in $\dot{\psi}$ at 2 seconds (bias = 0.02 rad/s).

for various fault sources and types. There are several key insights regarding this graph: (1) For both estimation paths, the uncertainty measure for excessive noise faults is significantly lower than for bias faults of the same magnitude. (2) In the case of a bias fault, path 2 provides more reliable (lower uncertainty measure) estimates when faults occur in $a_{y,m}$ and $a_{z,m}$, whereas path 1 is far more reliable and accurate for faults in $\dot{\psi}_m$. For bias-type lateral disturbances, both paths show the same level of uncertainty, suggesting that additional methods, such as UIO, could be used to mitigate its effects. (3) For noise-type faults, path 2 offers better reliability across all fault types and external disturbances. In contrast, path 1 has a higher uncertainty measure for faults in a_y and for external disturbances. (4) Finally, path 2 is notably less sensitive to noise than bias-type disturbances.

In conclusion, to ensure reliable estimation under faults, the framework dynamically reconfigures estimation paths using the reliability index in (8.4)—e.g., selecting path 2 for a lateral acceleration fault and path 1 for a yaw rate fault. This analysis can be extended to evaluate path sensitivity to various faults and disturbances. For instance, simulations show that path 1 is largely unaffected by biases in a_x . While both paths 1 and 2 are less sensitive to noise than to bias of the same magnitude, certain paths—especially those involving derivatives—may be more susceptible to noise. In some cases, a simplified model may also yield more accurate results, particularly when the fault affects only specific components.

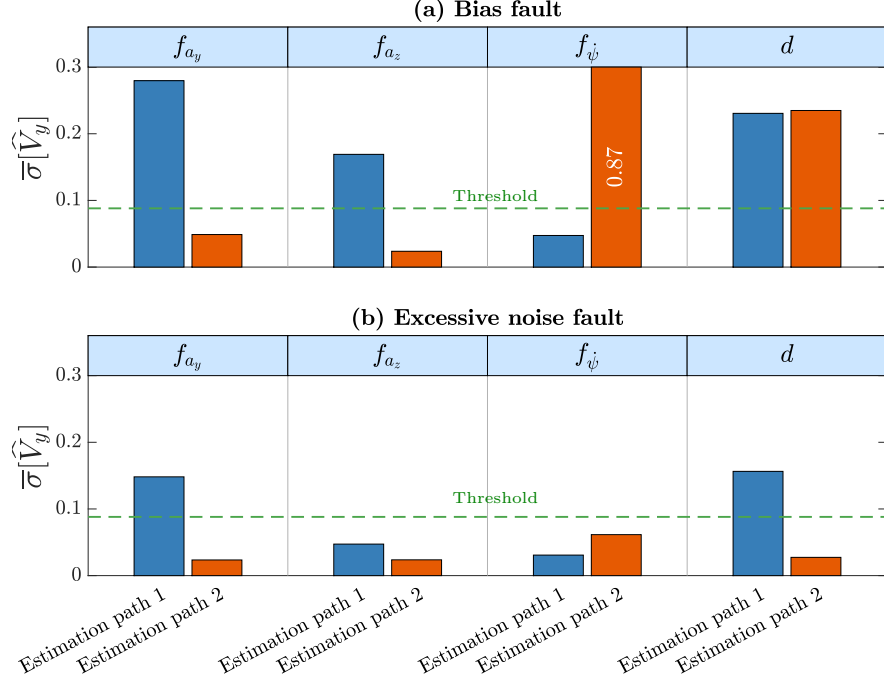


Figure 8.6: Comparison of uncertainty measure ($\bar{\sigma}[\hat{V}_y]$) in paths 1 and 2 for different fault sources and fault types. The threshold is set at $0.1 \max(V_y)$.

8.4 Experimental Validation and Results

This section presents the experimental results obtained from tests conducted on an electric Chevrolet Equinox vehicle. A cornering maneuver is performed at a nearly constant longitudinal velocity of 30 km/h.

To validate the effectiveness of the proposed methodology, a bias of 1 m/s^2 is injected into the measured lateral acceleration a_y at $t = 6 \text{ s}$ during the cornering maneuver. According to the fault signature matrix in Table 8.1, the residuals r_{ay}^{E3} and r_{ay}^{E4} are sufficient to detect a fault in the lateral acceleration measurement. Figure 8.7 illustrates these two residuals. As shown, both signals exceed their respective thresholds following the fault injection, thereby triggering fault detection. A fault is declared when both residuals remain above the thresholds for at least 20 consecutive time steps.

In the last stage of the framework, the *reliability index* of each estimation path over a time window T is evaluated under the isolated fault f , and the path with the highest index is selected as the most reliable for estimating the target states under that fault.

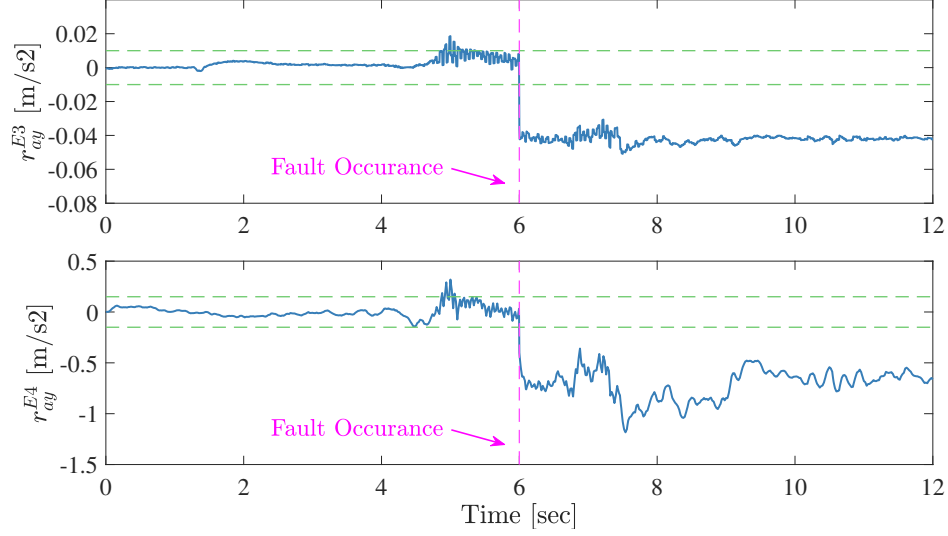


Figure 8.7: Residuals of lateral acceleration a_y in estimation paths 1 and 2. A bias fault of 1 m/s^2 is injected into a_y at $t = 6 \text{ s}$.

While classical reliability analysis typically defines reliability as the probability that a system or component performs its intended function without failure over a specified period, the statistical definition of reliability in measurement models refers to the degree of confidence in an observed variable. Specifically, the reliability ρ of a measured quantity $x = X_t + \epsilon$ is defined as the proportion of total observed variance attributable to the true value X_t , given by $\rho = \frac{\sigma_t^2}{\sigma_t^2 + \sigma_\epsilon^2}$; see [125]. This expression can also be written in the form

$$\rho = \frac{1}{1 + \beta \cdot \sigma_\epsilon^2}$$

where $\beta = \frac{1}{\sigma_t^2}$ is a scaling factor highlighting the inverse relationship between reliability and the variance of the measurement noise.

To assess the reliability of estimation under fault f , the same statistical perspective for physical measurements is adopted. The estimation uncertainty provides a natural surrogate for the error variance in the reliability equation. In soft fault scenarios—such as sensor bias or additive noise that degrade estimation without causing total failure—reliability reflects the confidence in the estimated values. Accordingly, a *reliability index* for an estimation path l under fault f at time step k is defined as:

$$\mathcal{R}_f^l(k) = \frac{1}{1 + \beta \|\Sigma_f^l(k)\|} \quad (8.4)$$

where $\|\Sigma_f^l\|$ is the uncertainty measure due to fault f defined in (8.3), and $\beta > 0$ is a tunable scaling parameter (set to $\beta = 1$ in our experiments). With this mapping, a path with zero uncertainty yields maximum reliability index $\mathcal{R}_f^l = 1$, while higher levels of uncertainty reduce the index toward zero. Finally, the optimal estimation path, l^* , under fault f is selected as the one whose reliability index $\mathcal{R}_f^l$ remains greater than that of all other paths for T consecutive time steps:

$$\mathcal{R}_f^{l^*}(k) > \max_{\substack{l \in \mathcal{L} \\ l \neq l^*}} \mathcal{R}_f^l(k) \quad \forall k \in [t - T + 1, t] \quad (8.5)$$

where $\mathcal{L}$ denotes the set of all available estimation paths.

Following fault detection via residual analysis (Figure 8.7), UQ is performed for each estimation path with respect to the identified fault—a bias in a_y in this case. The path with the highest reliability index over a 20-step window is selected as the most reliable. Figure 8.8 presents the lateral velocity estimates (top) and corresponding reliability indices (bottom) for the two paths under the fault scenario. Prior to the fault injection at $t = 6$ s, path 1 exhibits slightly better estimation accuracy and consistently higher reliability, and is therefore selected as the primary estimation path.

At $t = 6$ s, a bias fault of 1 m/s^2 is introduced into the lateral acceleration measurement a_y , and is detected shortly after at $t = 6.1$ s, based on residual thresholding. Following detection, the reliability indices of both estimation paths are evaluated. As expected from Figure 8.6, estimation path 2—which is less sensitive to faults in a_y —maintains a higher reliability index post-fault, while the reliability of path 1 degrades significantly.

Based on this assessment, the proposed framework initiates a reconfiguration at the switching time ($t = 6.2$ s) indicated in the figure, replacing path 1 with path 2 for subsequent estimation. This decision aligns with the reliability assessment and leads to improved estimation accuracy. The post-fault estimation results confirm that path 2 provides a more accurate estimate of lateral velocity under the given fault condition, thereby validating the effectiveness of the reliability-driven reconfiguration strategy.

8.5 Summary

The proposed framework enhances fault isolability by exploiting the structural properties of the estimation graph. Unlike traditional single-path estimators, the estimation graph comprises multiple interconnected estimation paths, allowing for the generation of additional

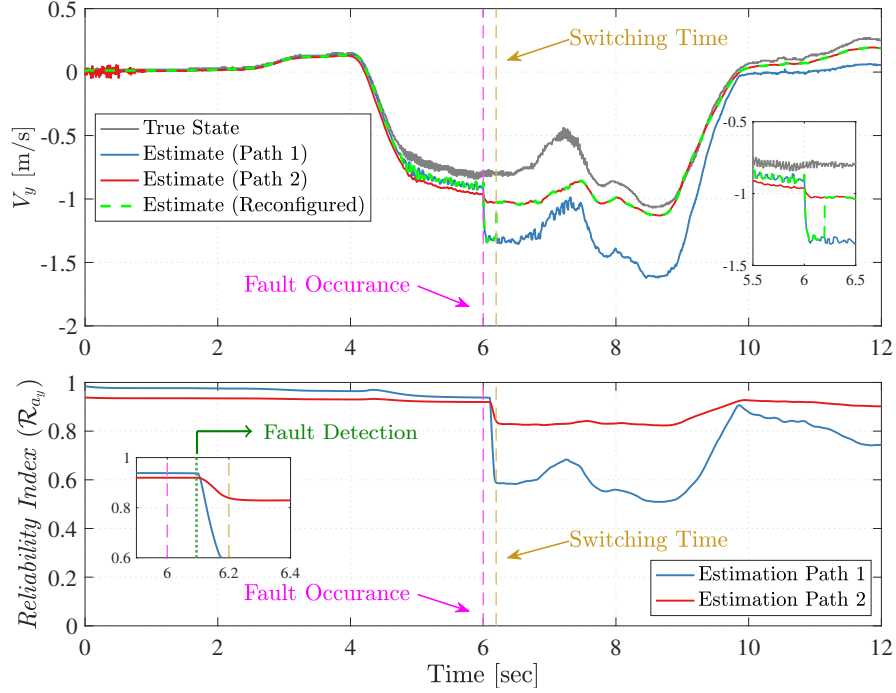


Figure 8.8: Lateral velocity estimation (top) and reliability index (bottom) for estimation paths 1 and 2. A bias fault of 1 m/s^2 is injected into a_y at $t = 6 \text{ s}$. Fault detection occurs at $t = 6.1 \text{ s}$, after which the reliability indices are evaluated, triggering a reconfiguration from path 1 to path 2 at $t = 6.2 \text{ s}$.

diagnostic signals—referred to as state residuals—derived from redundant estimates of the same state. These state residuals augment conventional measurement residuals and add extra rows to the fault signature table, enhancing isolability—especially in cases where faults are not isolable using a single estimator. This capability is demonstrated by the isolation of a fault in the vertical acceleration, which is undetectable using its respective estimator (E_2), but is successfully isolated using the lateral velocity residual derived from the estimation graph in Figure 8.2. Thus, the estimation graph enables richer residual generation and provides a principled and scalable foundation for FDI in complex systems.

Furthermore, by quantifying estimation uncertainty in real time, the framework enables dynamic reconfiguration to the most reliable estimation path under isolated faults, thereby improving robustness against both sensor faults and external disturbances.

Chapter 9

Conclusion and Future Directions

Modern vehicle systems increasingly rely on interconnected, modular estimators that form redundant and cascaded pathways to ensure robust state estimation under varying operational and fault conditions. This thesis has demonstrated that while such architectures provide fault tolerance through redundancy, they also introduce complex dependencies that may lead to cascading fault propagation across estimation modules. A central insight is that reliability in such interconnected estimation systems must be actively managed by systematically leveraging prior knowledge—such as system dynamics model, fault distributions, and structural interdependencies.

Real-time fault diagnosis and real-time-capable uncertainty quantification (UQ) are essential for managing estimation reliability. The analytical and surrogate-based UQ techniques developed in this work enabled accurate real-time reliability assessment without relying on expensive Monte Carlo simulations. The analytical method developed for linear/linearized estimators, while the gPC-based surrogate extended applicability to nonlinear and black-box models. These methods proved effective for embedded implementation, enabling timely estimation reconfiguration under fault conditions.

A further critical realization was the value of structural modeling in fault diagnosis. By introducing the *estimation graph*, this thesis provided a structured method for analyzing how faults manifest across estimation modules. This abstraction facilitated the derivation of a fault signature matrix based on structural dependencies, offering both interpretability and scalability. Notably, it was shown that structural analysis alone can reveal otherwise undetectable faults when viewed from downstream nodes in the graph, highlighting the importance of topological reasoning in fault diagnosis.

Another key insight was that hybrid methods— combining model-based and data-driven

reasoning—consistently outperform individual approaches. Fault diagnosis was improved by leveraging system models at both quantitative and structural levels to guide learning:

- Quantitatively, system dynamics model and fault distributions were used to derive residual uncertainty and fault likelihoods in cascaded estimators, yielding superior isolation accuracy compared to purely model-based or data-driven methods, even in data-limited scenarios. Moreover, the fault magnitude was identified using mathematically-quantified impact of faults on residuals.
- Structurally, the introduced fault interaction graph (FIG) captured dependencies among fault sources. This insight enabled two distributed isolation methods—a consensus-based algorithm and a GCN-based classifier—that achieved high accuracy with significantly lower complexity and data requirements than centralized architectures.

To enhance scalability, distributed isolation strategies were developed based on structural analysis. The extracted fault signature matrix was first refined through correlation and sensitivity analysis to reduce weak dependencies—for improving interpretability and scalability in large-scale systems. A novel one-mode projection was then applied to the refined matrix to construct the FIG, effectively encoding the underlying dependencies among fault sources.

The first strategy, a consensus-based method, offered modularity and supported partial supervision with minimal trainable parameters. The graph-theoretic tools such as coverage ratio and maximum independent sets revealed how fault isolation performance depends on the structural location of labeled nodes. These insights support efficient labeling strategies by prioritizing topological hubs and avoiding structurally isolated faults, especially in large-scale systems.

The second strategy employed a GCN-based global classifier that incorporated topological information directly into the learning process. This approach achieved higher accuracy and faster convergence than centralized models with reduced data. Guided by prior structural information, the GCN-based network maintained strong performance even in data-scarce scenarios and reduced model complexity.

Unlike centralized methods, the proposed distributed approaches supported modular retraining: only the affected local detectors and global classifier (with an updated adjacency matrix) required retraining when new faults were added. This made the hybrid distributed framework scalable, data-efficient, and well-suited for large-scale systems with evolving fault

configurations, offering significant computational and operational advantages for real-world deployment.

In conclusion, this work introduced a unified framework for fault diagnosis and reliability-based topology selection in vehicle state estimation. By integrating system model knowledge, real-time uncertainty quantification, structural analysis, and data-driven learning, the proposed methods enabled accurate and efficient fault diagnosis and fault-tolerant estimation. Abstractions such as the estimation graph and FIG, along with graph-theoretic tools like coverage ratio and MIS, improved interpretability and supported scalable, distributed diagnostic architectures. The modular design enabled reduced computational overhead, addressing key challenges in deploying reliable estimation in modern complex automated vehicles.

To further broaden the scope and practical impact of this framework, future work can be pursued along two directions: (i) expanding its application to new domains and scenarios, and (ii) advancing its methodological capabilities to enhance diagnostic performance.

Future Application-Oriented Directions:

- **Extension to Large-Scale Networked Systems:** The proposed distributed fault isolation methods are inherently scalable and can be applied to a wide range of large-scale systems beyond automotive applications. These include power grids, industrial process networks, transportation systems, and complex cyber-physical infrastructures. By leveraging solely structural connectivity instead of detailed models, the framework enables scalable, real-time fault isolation in complex, high-dimensional systems where full modeling is impractical.
- **Expansion to Diverse Fault Types and System Layers:** This work focuses on soft faults such as excessive noise and bias, common in physical sensors and actuators. Future extensions could address other fault types—e.g., data dropouts, stuck-at faults, and transient spikes—as well as timing delays, and communication errors in networked systems. Broadening the scope to these multi-layered scenarios would enhance the framework’s applicability to a wider range of cyber-physical and autonomous systems.
- **Virtualization and Digital Twin Applications:** The developed uncertainty quantification methods can be leveraged to simulate fault and uncertainty propagation within virtual environments or digital twins of complex dynamical systems. These platforms offer a cost-effective, risk-free alternative to physical testing, especially for hazardous, rare, or expensive scenarios. Integrating the proposed analytical techniques

enables more realistic and robust insights into system behavior under diverse fault conditions, supporting design validation, fault injection, and predictive maintenance.

- **Extension to Advanced Sensing in Autonomous Vehicles:** As autonomous vehicles increasingly depend on sensors like cameras, radars, LiDARs, and GNSS, future work could extend the proposed framework to handle faults in these perception systems. This involves modeling their failure modes and developing hybrid learning-based methods for real-time fault detection, isolation, and correction—enhancing reliability in complex, uncertain environments. Such developments would significantly enhance the reliability and safety.
- **Integration with Fault-Tolerant Control:** Building upon the FDI framework, a natural extension is its integration with fault-tolerant control (FTC). The identified faults and their magnitudes can be used to reconfigure controllers in real time, closing the loop for safe operation under faulty conditions.

Future Methodological Directions:

- **Hierarchical and Zone-Based Fault Diagnosis:** For large-scale systems, a hierarchical distributed FDI architecture can be adopted. Fault localization begins by identifying a faulty zone or subsystem using high-level indicators or rule-based symptom monitoring. A zone-specific isolation algorithm (e.g., a local version of the proposed hybrid method) is then applied to pinpoint the fault source. This modular design supports scalable expansion and simplifies diagnosis in systems with many potential fault points.
- **Integration of Causal Graphs for Root Cause Analysis:** The current framework employs the estimation graph and FIG, based on structural connectivity, for diagnostic reasoning. A promising extension is to construct causal graphs from the estimation structure to capture the directionality of fault propagation. Unlike data-driven causal discovery, the estimation graph provides model-derived causal relationships, enabling transparent and data-efficient inference. Tracing faults along causal paths helps distinguish root causes from secondary effects, enhancing interpretability, reducing false positives due to indirect correlations, and supporting explainable fault propagation and root cause localization in complex systems.
- **Integration of Probabilistic Graphical Models:** A promising future direction involves leveraging quantified uncertainty due to faults as probabilistic input to graphical models such as Bayesian or conditional graph networks. Combined with causal

structures from the estimation graph, this enables modeling both fault propagation directionality and associated uncertainties. The resulting framework supports interpretable, probabilistically grounded diagnosis, improves root cause disambiguation, reduces false positives, and enhances reasoning under uncertainty in dynamic or partially observable systems.

- **Data-Driven Graph Refinement and Topology Adaptation:** Future work could focus on enhancing both the BFSG and the FIG through data-driven adaptations. For the BFSG, assigning edge weights based on residual sensitivity metrics can quantify the strength of fault-residual relationships, yielding more informative and sparse projections into the FIG. Simultaneously, the FIG itself can be refined or dynamically adapted using residual correlations, causal inference, or attention-based mechanisms to capture evolving system behavior or modeling uncertainties. These data-informed graph structures offer greater flexibility, robustness, and accuracy for fault isolation in real-world systems operating under dynamic conditions.
- **Data-Driven Fault Magnitude Identification:** While this thesis employed a model-based technique to estimate fault magnitudes in Kalman-based cascaded estimators, future work could integrate data-driven methods in black-box or nonlinear estimators—especially in transient conditions where model assumptions may break down. Learning-based regressors trained on historical residuals can also enhance robustness in complex or nonlinear scenarios.

These future directions aim to enhance the scalability, interpretability, and adaptability of the proposed methods by advancing both its methodological foundations and practical applicability. By integrating causal inference, probabilistic reasoning, and data-driven graph adaptations, the framework can better address the challenges of fault diagnosis in dynamic, uncertain environments. Simultaneously, expanding its deployment to complex, safety-critical systems—such as connected autonomous vehicles, smart grids, and distributed cyber-physical infrastructures—ensures its continued relevance and impact in real-world applications.

References

- [1] Dieter Schramm, Manfred Hiller, Roberto Bardini, et al. *Modellbildung und simulation der dynamik von Kraftfahrzeugen*, volume 124. Springer, 2010.
- [2] Kanwar Bharat Singh, Mustafa Ali Arat, and Saied Taheri. Literature review and fundamental approaches for vehicle and tire state estimation. *Vehicle system dynamics*, 57(11):1643–1665, 2019.
- [3] Xianjian Jin, Guodong Yin, and Nan Chen. Advanced estimation techniques for vehicle system dynamic state: A survey. *Sensors*, 19(19):4289, 2019.
- [4] Hongyan Guo, Zhenyu Yin, Dongpu Cao, Hong Chen, and Chen Lv. A review of estimation for vehicle tire-road interactions toward automated driving. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(1):14–30, 2018.
- [5] Jianfeng Chen, Shulin Hu, Yicai Ye, Haoqian Huang, Reza Langari, and Chuanye Tang. A cascaded scheme for high-performance estimation of vehicle states. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 235(8):2101–2113, 2021.
- [6] Ayyoub Rezaeian, Reza Zarringhalam, Saber Fallah, William Melek, Amir Khajepour, Shih-Ken Chen, and Baktiarr Litkouhi. Cascaded dual extended kalman filter for combined vehicle state estimation and parameter identification. Technical report, 2013.
- [7] David M Bevly and Bradford Parkinson. Cascaded kalman filters for accurate estimation of multiple biases, dead-reckoning navigation, and full state feedback control of ground vehicles. *IEEE Transactions on Control Systems Technology*, 15(2): 199–208, 2007.

- [8] Pengyu Han, Zeyi Liu, Xiao He, Steven X Ding, and Donghua Zhou. Multi-condition fault diagnosis of dynamic systems: A survey, insights, and prospects. *IEEE Transactions on Automation Science and Engineering*, 2025.
- [9] Magdi S Mahmoud and Haris M Khalid. Distributed kalman filtering: a bibliographic review. *IET Control Theory & Applications*, 7(4):483–501, 2013.
- [10] Dragoslav D Siljak. *Decentralized control of complex systems*. Courier Corporation, 2011.
- [11] Mehrdad Saif and Yuping Guan. Decentralized state estimation in large-scale interconnected dynamical systems. *Automatica*, 28(1):215–219, 1992.
- [12] Dapeng Wang, Hai Zhang, and Baoshuang Ge. Adaptive unscented kalman filter for target tacking with time-varying noise covariance based on multi-sensor information fusion. *Sensors*, 21(17):5808, 2021.
- [13] Takayuki Ishizaki, Yukihiro Sakai, Kenji Kashima, and Jun-ichi Imura. Hierarchical decentralized observer design for linearly coupled network systems. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7831–7836. IEEE, 2011.
- [14] Shesheng Gao, Yongmin Zhong, Xueyuan Zhang, and Bijan Shirinzadeh. Multi-sensor optimal data fusion for ins/gps/sar integrated navigation system. *Aerospace Science and Technology*, 13(4-5):232–237, 2009.
- [15] Martin Liggins II, David Hall, and James Llinas. *Handbook of multisensor data fusion: theory and practice*. CRC press, 2017.
- [16] Arthur GO Mutambara. *Decentralized estimation and control for multisensor systems*. CRC press, 1998.
- [17] Xu Li and Weigong Zhang. Reliable integrated navigation system based on adaptive fuzzy federated kalman filter for automated vehicles. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 224(3):327–346, 2010.
- [18] Xu Li, Ching-Yao Chan, and Yu Wang. A reliable fusion methodology for simultaneous estimation of vehicle sideslip and yaw angles. *IEEE Transactions on Vehicular Technology*, 65(6):4440–4458, 2015.

- [19] Neal A Carlson. Federated square root filter for decentralized parallel processors. *IEEE Transactions on Aerospace and Electronic Systems*, 26(3):517–525, 1990.
- [20] Xu Li and Qimin Xu. A reliable fusion positioning strategy for land vehicles in gps-denied environments based on low-cost sensors. *IEEE Transactions on Industrial Electronics*, 64(4):3205–3215, 2016.
- [21] Igor Mezić and Thordur Runolfsson. Uncertainty propagation in dynamical systems. *Automatica*, 44(12):3003–3013, 2008.
- [22] Jie Liu, Xianghua Meng, Can Xu, Dequan Zhang, and Chao Jiang. Forward and inverse structural uncertainty propagations under stochastic variables with arbitrary probability distributions. *Computer Methods in Applied Mechanics and Engineering*, 342:287–320, 2018.
- [23] Timothy John Sullivan. *Introduction to uncertainty quantification*, volume 63. Springer, 2015.
- [24] Hannes Risken and Hannes Risken. *Fokker-planck equation*. Springer, 1996.
- [25] John L Crassidis and John L Junkins. *Optimal estimation of dynamic systems*. Chapman and Hall/CRC, 2004.
- [26] Frederick Daum. Exact finite-dimensional nonlinear filters. *IEEE Transactions on Automatic Control*, 31(7):616–622, 1986.
- [27] Fred Schweppe. *Uncertain dynamic systems*. Prentice-Hall electrical engineering series. Prentice Hall, Old Tappan, NJ, September 1973.
- [28] Min-Hee Gu, Chihyun Cho, Hahng-Yun Chu, No-Weon Kang, and Joo-Gwang Lee. Uncertainty propagation on a nonlinear measurement model based on taylor expansion. *Measurement and Control*, 54(3-4):209–215, 2021.
- [29] Hamed Haddad Khodaparast, John E Mottershead, and Michael I Friswell. Perturbation methods for the estimation of parameter variability in stochastic model updating. *Mechanical systems and signal processing*, 22(8):1751–1773, 2008.
- [30] Ryan S Park and Daniel J Scheeres. Nonlinear mapping of gaussian statistics: theory and applications to spacecraft trajectory design. *Journal of Guidance, Control, and Dynamics*, 29(6):1367–1375, 2006.

- [31] Sang Hoon Lee and Wei Chen. A comparative study of uncertainty propagation methods for black-box-type problems. *Structural and multidisciplinary optimization*, 37:239–253, 2009.
- [32] Gabriel Terejanu, Puneet Singla, Tarunraj Singh, and Peter D Scott. Uncertainty propagation for nonlinear dynamic systems using gaussian mixture models. *Journal of guidance, control, and dynamics*, 31(6):1623–1633, 2008.
- [33] Roger Ghanem, David Higdon, Houman Owhadi, et al. *Handbook of uncertainty quantification*, volume 6. Springer, 2017.
- [34] Geir Evensen. The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53:343–367, 2003.
- [35] Peter Matisko and Vladimír Havlena. Noise covariance estimation for kalman filter tuning using bayesian approach and monte carlo. *International Journal of Adaptive Control and Signal Processing*, 27(11):957–973, 2013.
- [36] Petar M Djuric, Jayesh H Kotecha, Jianqui Zhang, Yufei Huang, Tadesse Ghirmai, Mónica F Bugallo, and Joaquin Miguez. Particle filtering. *IEEE signal processing magazine*, 20(5):19–38, 2003.
- [37] Arnaud Doucet, Nando De Freitas, Neil James Gordon, et al. *Sequential Monte Carlo methods in practice*, volume 1. Springer, 2001.
- [38] Biswarup Bhattacharyya. Uncertainty quantification of dynamical systems by a pod–kriging surrogate model. *Journal of Computational Science*, 60:101602, 2022.
- [39] Dongbin Xiu and George Em Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, 24(2):619–644, 2002.
- [40] Xiaoliang Wan and George Em Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics*, 209(2):617–642, 2005.
- [41] Eric Jacquelin, Sondipon Adhikari, J-J Sinou, and Michael I Friswell. Polynomial chaos expansion and steady-state response of a class of random dynamical systems. *Journal of Engineering Mechanics*, 141(4):04014145, 2015.

- [42] Adrian Sandu, Corina Sandu, and Mehdi Ahmadian. Modeling multibody systems with uncertainties. part i: Theoretical and computational aspects. *Multibody System Dynamics*, 15:369–391, 2006.
- [43] Jinglai Wu, Zhen Luo, Nong Zhang, and Yunqing Zhang. A new uncertain analysis method and its application in vehicle dynamics. *Mechanical Systems and Signal Processing*, 50:659–675, 2015.
- [44] Corina Sandu, Adrian Sandu, and Mehdi Ahmadian. Modeling multibody systems with uncertainties. part ii: Numerical applications. *Multibody System Dynamics*, 15: 241–262, 2006.
- [45] Gaurav Kewlani, Justin Crawford, and Karl Iagnemma. A polynomial chaos approach to the analysis of vehicle dynamics under uncertainty. *Vehicle system dynamics*, 50 (5):749–774, 2012.
- [46] Jinglai Wu, Yunqing Zhang, Liping Chen, and Zhen Luo. A chebyshev interval method for nonlinear dynamic systems under uncertainty. *Applied Mathematical Modelling*, 37(6):4578–4591, 2013.
- [47] Jinglai Wu, Zhen Luo, Yunqing Zhang, Nong Zhang, and Liping Chen. Interval uncertain method for multibody mechanical systems using chebyshev inclusion functions. *International Journal for Numerical Methods in Engineering*, 95(7):608–630, 2013.
- [48] Zhe Chen et al. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.
- [49] Haigen Min, Yukun Fang, Xia Wu, Xiaoping Lei, Shixiang Chen, Rui Teixeira, Bing Zhu, Xiangmo Zhao, and Zhigang Xu. A fault diagnosis framework for autonomous vehicles with sensor self-diagnosis. *Expert Systems with Applications*, 224:120002, 2023.
- [50] Franco Van Wyk, Yiyang Wang, Anahita Khojandi, and Neda Masoud. Real-time sensor anomaly detection and identification in automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):1264–1276, 2019.
- [51] Corina Sandu, Adrian Sandu, and Mehdi Ahmadian. Modeling multibody systems with uncertainties. part ii: Numerical applications. *Multibody System Dynamics*, 15: 241–262, 2006.

- [52] Byoung-Gyu Song, Jong-Jin Bae, and Namcheol Kang. Uncertainty quantification of ride comfort based on gpc framework for a fully coupled human–vehicle model. *Applied Sciences*, 13(11):6785, 2023.
- [53] Yijun Xu, Lamine Mili, and Junbo Zhao. A novel polynomial-chaos-based kalman filter. *IEEE Signal Processing Letters*, 26(1):9–13, 2018.
- [54] Renato Zanetti and Robert H Bishop. Kalman filters with uncompensated biases. *Journal of Guidance, Control, and Dynamics*, 35(1):327–335, 2012.
- [55] Awais Khan, Wei Xie, Bo Zhang, and Long-Wen Liu. A survey of interval observers design methods and implementation for uncertain systems. *Journal of the Franklin institute*, 358(6):3077–3126, 2021.
- [56] Tenglong Huang, Huihui Pan, and Weichao Sun. A sensor fault detection, isolation, and estimation method for intelligent vehicles. *Control Engineering Practice*, 139: 105620, 2023.
- [57] Sid-Ahmed Raka and Christophe Combastel. Fault detection based on robust adaptive thresholds: A dynamic interval approach. *Annual Reviews in Control*, 37(1):119–128, 2013.
- [58] Yamei Ju, Xin Tian, Hongjian Liu, and Lifeng Ma. Fault detection of networked dynamical systems: A survey of trends and techniques. *International Journal of Systems Science*, 52(16):3390–3409, 2021.
- [59] André Teixeira, Iman Shames, Henrik Sandberg, and Karl H Johansson. Distributed fault detection and isolation resilient to network model uncertainties. *IEEE transactions on cybernetics*, 44(11):2024–2037, 2014.
- [60] Aadil Sarwar Khan, Abdul Qayyum Khan, Naeem Iqbal, Ghulam Mustafa, Muhammad Asim Abbasi, and Atif Mahmood. Design of a computationally efficient observer-based distributed fault detection and isolation scheme in second-order networked control systems. *ISA transactions*, 128:229–241, 2022.
- [61] Jun Hu, Chaoqing Jia, Hongjian Liu, Xiaojian Yi, and Yurong Liu. A survey on state estimation of complex dynamical networks. *International Journal of Systems Science*, 52(16):3351–3367, 2021.
- [62] Khaoula Tidriri, Nizar Chatti, Sylvain Verron, and Teodor Tiplica. Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring:

A review of researches and future challenges. *Annual reviews in control*, 42:63–81, 2016.

- [63] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Kewen Yin, and Surya N Kavuri. A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. *Computers & chemical engineering*, 27(3):293–311, 2003.
- [64] Haedong Jeong, Bumsoo Park, Seungtae Park, Hyungcheol Min, and Seungchul Lee. Fault detection and identification method using observer-based residuals. *Reliability Engineering & System Safety*, 184:27–40, 2019.
- [65] Jie Chen, Ron J Patton, and Hong-Yue Zhang. Design of unknown input observers and robust fault detection filters. *International Journal of control*, 63(1):85–105, 1996.
- [66] Saeed Ahmadizadeh, Jafar Zarei, and Hamid Reza Karimi. Robust unknown input observer design for linear uncertain time delay systems with application to fault detection. *Asian Journal of Control*, 16(4):1006–1019, 2014.
- [67] Shreyas Sundaram and Christoforos N Hadjicostis. Partial state observers for linear systems with unknown inputs. *Automatica*, 44(12):3126–3132, 2008.
- [68] Iman Shames, André MH Teixeira, Henrik Sandberg, and Karl H Johansson. Distributed fault detection for interconnected second-order systems. *Automatica*, 47(12):2757–2764, 2011.
- [69] Mohammadreza Doostmohammadian and Nader Meskin. Sensor fault detection and isolation via networked estimation: Full-rank dynamical systems. *IEEE Transactions on Control of Network Systems*, 8(2):987–996, 2020.
- [70] Mohammadreza Doostmohammadian, Houman Zarrabi, and Themistoklis Charalambous. Sensor fault detection and isolation via networked estimation: rank-deficient dynamical systems. *International Journal of Control*, 96(11):2853–2870, 2023.
- [71] Mohammad Pirani, Aritra Mitra, and Shreyas Sundaram. Graph-theoretic approaches for analyzing the resilience of distributed control systems: A tutorial and survey. *Automatica*, 157:111264, 2023.
- [72] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

- [73] Navid Mohammadi Foumani, Lynn Miller, Chang Wei Tan, Geoffrey I Webb, Germain Forestier, and Mahsa Salehi. Deep learning for time series classification and extrinsic regression: A current survey. *ACM Computing Surveys*, 56(9):1–45, 2024.
- [74] Jie Liu, Yang Hu, and Shunkun Yang. A svm-based framework for fault detection in high-speed trains. *Measurement*, 172:108779, 2021.
- [75] Hao Wu and Jinsong Zhao. Deep convolutional neural network model based chemical process fault diagnosis. *Computers & chemical engineering*, 115:185–197, 2018.
- [76] Yahui Zhang, Taotao Zhou, Xufeng Huang, Longchao Cao, and Qi Zhou. Fault diagnosis of rotating machinery based on recurrent neural networks. *Measurement*, 171:108774, 2021.
- [77] Kun Zhou, Yifan Tong, Xintong Li, Xiaoran Wei, Hao Huang, Kai Song, and Xu Chen. Exploring global attention mechanism on fault detection and diagnosis for complex engineering processes. *Process Safety and Environmental Protection*, 170:660–669, 2023.
- [78] Shaohua Qiu, Xiaopeng Cui, Zuowei Ping, Nanliang Shan, Zhong Li, Xianqiang Bao, and Xinghua Xu. Deep learning techniques in intelligent fault diagnosis and prognosis for industrial systems: A review. *Sensors*, 23(3):1305, 2023.
- [79] Jinchuan Qian, Zhihuan Song, Yuan Yao, Zheren Zhu, and Xinmin Zhang. A review on autoencoder based representation learning for fault detection and diagnosis in industrial processes. *Chemometrics and Intelligent Laboratory Systems*, 231:104711, 2022.
- [80] Mehdi Jabbari Zideh, Paroma Chatterjee, and Anurag K Srivastava. Physics-informed machine learning for data anomaly detection, classification, localization, and mitigation: A review, challenges, and path forward. *IEEE Access*, 12:4597–4617, 2023.
- [81] Kunjin Chen, Jun Hu, Yu Zhang, Zhanqing Yu, and Jinliang He. Fault location in power distribution systems via deep graph convolutional networks. *IEEE Journal on Selected Areas in Communications*, 38(1):119–131, 2019.
- [82] Zhiwen Chen, Jiamin Xu, Cesare Alippi, Steven X Ding, Yuri Shardt, Tao Peng, and Chunhua Yang. Graph neural network-based fault diagnosis: A review. *arXiv preprint arXiv:2111.08185*, 2021.

- [83] Dongyue Chen, Ruonan Liu, Qinghua Hu, and Steven X Ding. Interaction-aware graph neural networks for fault diagnosis of complex industrial processes. *IEEE Transactions on neural networks and learning systems*, 34(9):6015–6028, 2021.
- [84] Mingwei Jia, Junhao Hu, Yi Liu, Zengliang Gao, and Yuan Yao. Topology-guided graph learning for process fault diagnosis. *Industrial & Engineering Chemistry Research*, 62(7):3238–3248, 2023.
- [85] Bang LH Nguyen, Tuyen V Vu, Thai-Thanh Nguyen, Mayank Panwar, and Rob Hovsopian. Spatial-temporal recurrent graph neural networks for fault diagnostics in power distribution systems. *IEEE Access*, 11:46039–46050, 2023.
- [86] Hossein Darvishi, Domenico Ciuonzo, and Pierluigi Salvo Rossi. Deep recurrent graph convolutional architecture for sensor fault detection, isolation, and accommodation in digital twins. *IEEE Sensors Journal*, 23(23):29877–29891, 2023.
- [87] Tianfu Li, Zheng Zhou, Sinan Li, Chuang Sun, Ruqiang Yan, and Xuefeng Chen. The emerging graph neural networks for intelligent fault diagnostics and prognostics: A guideline and a benchmark study. *Mechanical Systems and Signal Processing*, 168: 108653, 2022.
- [88] Benjamin Nguyen and Moncef Chioua. A propagation path-based interpretable neural network model for fault detection and diagnosis in chemical process systems. *Control Engineering Practice*, 150:105988, 2024.
- [89] Yi-Jing Zhang and Li-Sheng Hu. Fault propagation inference based on a graph neural network for steam turbine systems. *Energies*, 14(2):309, 2021.
- [90] Zhiwen Chen, Jiamin Xu, Tao Peng, and Chunhua Yang. Graph convolutional network-based method for fault diagnosis using a hybrid of measurement and prior knowledge. *IEEE transactions on cybernetics*, 52(9):9157–9169, 2021.
- [91] Zhaoming Miao, Yingxiang Xia, Fengyu Zhou, and Xianfeng Yuan. Fault diagnosis of wheeled robot based on prior knowledge and spatial-temporal difference graph convolutional network. *IEEE Transactions on Industrial Informatics*, 19(5):7055–7065, 2022.
- [92] Lei Guo, Hongbo Shi, Shuai Tan, Bing Song, and Yang Tao. Sensor fault detection and diagnosis using graph convolutional network combining process knowledge and process data. *IEEE Transactions on Instrumentation and Measurement*, 72:1–10, 2023.

- [93] Wenjian Wu, Aiping Pang, and Wen Yang. Heterogeneous sensor fault detection for networked systems based on a graph transformer. *IEEE Sensors Journal*, 24(4): 5266–5278, 2023.
- [94] Daniel Jung, Kok Yew Ng, Erik Frisk, and Mattias Krysander. Combining model-based diagnosis and data-driven anomaly classifiers for fault isolation. *Control Engineering Practice*, 80:146–156, 2018.
- [95] M Amine Atoui and Achraf Cohen. Coupling data-driven and model-based methods to improve fault diagnosis. *Computers in Industry*, 128:103401, 2021.
- [96] Jianhui Luo, Madhavi Namburu, Krishna R Pattipati, Liu Qiao, and Shunsuke Chigusa. Integrated model-based and data-driven diagnosis of automotive antilock braking systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(2):321–336, 2009.
- [97] Dilek Düstegör, Erik Frisk, Vincent Cocquempot, Mattias Krysander, and Marcel Staroswiecki. Structural analysis of fault isolability in the damadics benchmark. *Control Engineering Practice*, 14(6):597–608, 2006.
- [98] Daniel Jung and Christofer Sundström. A combined data-driven and model-based residual selection algorithm for fault detection and isolation. *IEEE Transactions on Control Systems Technology*, 27(2):616–630, 2017.
- [99] Yang Zhao, Fu Xiao, and Shengwei Wang. An intelligent chiller fault detection and diagnosis methodology using bayesian belief network. *Energy and Buildings*, 57: 278–288, 2013.
- [100] Mohamed Amine Atoui, Sylvain Verron, and Abdessamad Kobi. A bayesian network dealing with measurements and residuals for system monitoring. *Transactions of the Institute of Measurement and Control*, 38(4):373–384, 2016.
- [101] Balyogi Mohan Dash, Belkacem Ould Bouamama, Mahdi Boukerdja, and Komi Midzodzi Pekpe. Bond graph-cnn based hybrid fault diagnosis with minimum labeled data. *Engineering Applications of Artificial Intelligence*, 131:107734, 2024.
- [102] Liqiao Xia, Yongshi Liang, Pai Zheng, and Xiao Huang. Residual-hypergraph convolution network: A model-based and data-driven integrated approach for fault diagnosis in complex equipment. *IEEE Transactions on Instrumentation and Measurement*, 72: 1–11, 2022.

- [103] Aleksandr Kovalenko, Vitaliy Pozdnyakov, and Ilya Makarov. Graph neural networks with trainable adjacency matrices for fault diagnosis on multivariate sensor data. *IEEE Access*, 2024.
- [104] Xiangmo Zhao, Yukun Fang, Haigen Min, Xia Wu, Wuqi Wang, and Rui Teixeira. Potential sources of sensor data anomalies for autonomous vehicles: An overview from road vehicle safety perspective. *Expert Systems with Applications*, 236:121358, 2024.
- [105] Hok K Ng, Robert H Chen, and Jason L Speyer. A vehicle health monitoring system evaluated experimentally on a passenger vehicle. *IEEE Transactions on control systems technology*, 14(5):854–870, 2006.
- [106] Wonbin Na, Changwoo Park, Seokjoo Lee, Seongo Yu, and Hyeongcheol Lee. Sensitivity-based fault detection and isolation algorithm for road vehicle chassis sensors. *Sensors*, 18(8):2720, 2018.
- [107] Shiqing Li, Michael Frey, and Frank Gauterin. Model-based condition monitoring of the sensors and actuators of an electric and automated vehicle. *Sensors*, 23(2):887, 2023.
- [108] Ehsan Hashemi, Reza Zarringhalam, Amir Khajepour, William Melek, Alireza Kasaeizadeh, and Shih-Ken Chen. Real-time estimation of the road bank and grade angles with unknown input observers. *Vehicle system dynamics*, 55(5):648–667, 2017.
- [109] Miguel Meléndez-Useros, Manuel Jiménez-Salas, Fernando Viadero-Monasterio, and María Jesús López-Boada. Novel methodology for integrated actuator and sensors fault detection and estimation in an active suspension system. *IEEE Transactions on Reliability*, 2024.
- [110] Samir Benmoussa, Belkacem Ould Bouamama, and Rochdi Merzouki. Bond graph approach for plant fault detection and isolation: Application to intelligent autonomous vehicle. *IEEE Transactions on Automation Science and Engineering*, 11(2):585–593, 2013.
- [111] Wijaya Kurniawan and Lorinc Marton. Sensor faults isolation in networked control systems: Application to mobile robot platoons. *Sensors*, 21(20):6702, 2021.
- [112] Saeid Safavi, Mohammad Amin Safavi, Hossein Hamid, and Saber Fallah. Multi-sensor fault detection, identification, isolation and health forecasting for autonomous vehicles. *Sensors*, 21(7):2547, 2021.

- [113] Jie Man, Honghui Dong, Limin Jia, and Yong Qin. Attggen model: A novel multi-sensor fault diagnosis method for high-speed train bogie. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):19511–19522, 2022.
- [114] Barry Dowdeswell, Roopak Sinha, and Stephen G MacDonell. Finding faults: A scoping study of fault diagnostics for industrial cyber–physical systems. *Journal of systems and software*, 168:110638, 2020.
- [115] Moustapha Doumiati, Alessandro Victorino, Ali Charara, Guillaume Baffet, and Daniel Lechner. An estimation process for vehicle wheel-ground contact normal forces. *IFAC Proceedings Volumes*, 41(2):7110–7115, 2008.
- [116] Moustapha Doumiati, A Victorino, Daniel Lechner, G Baffet, and A Charara. Observers for vehicle tyre/road forces estimation: experimental validation. *Vehicle System Dynamics*, 48(11):1345–1378, 2010.
- [117] Jie Chen and Ron J Patton. *Robust model-based fault diagnosis for dynamic systems*, volume 3. Springer Science & Business Media, 2012.
- [118] Bahram Shafai and Mehrdad Saif. Proportional-integral observer in robust control, fault detection, and decentralized control of dynamic systems. In *Control and Systems Engineering: A Report on Four Decades of Contributions*, pages 13–43. Springer, 2015.
- [119] Mohinder S Grewal and Angus P Andrews. *Kalman filtering: Theory and Practice with MATLAB*. John Wiley & Sons, 2014.
- [120] Steven J Leon, Åke Björck, and Walter Gander. Gram-schmidt orthogonalization: 100 years and more. *Numerical Linear Algebra with Applications*, 20(3):492–532, 2013.
- [121] Walter Gautschi. Algorithm 726: Orthpol—a package of routines for generating orthogonal polynomials and gauss-type quadrature rules. *ACM Transactions on Mathematical Software (TOMS)*, 20(1):21–62, 1994. URL <https://dl.acm.org/citation.cfm?id=174605>.
- [122] Jianbo Yu and Yue Zhang. Challenges and opportunities of deep learning-based process fault detection and diagnosis: a review. *Neural Computing and Applications*, 35(1):211–252, 2023.
- [123] Suman Banerjee, Mamata Jenamani, and Dilip Kumar Pratihari. Properties of a projected network of a bipartite network. In *2017 International Conference on Communication and Signal Processing (ICCSP)*, pages 0143–0147. IEEE, 2017.

- [124] Jay A Farrell, Felipe O Silva, Farzana Rahman, and Jan Wendel. Inertial measurement unit error modeling tutorial: Inertial navigation system state estimation with real-time sensor calibration. *IEEE Control Systems Magazine*, 42(6):40–66, 2022.
- [125] Wayne A Fuller. *Measurement error models*. John Wiley & Sons, 2009.
- [126] F. Bullo. *Lectures on Network Systems*. Kindle Direct Publishing, 1.6 edition, 2022. ISBN 978-1986425643. URL <http://motion.me.ucsb.edu/book-1ns>.
- [127] Hans B Pacejka and Egbert Bakker. The magic formula tyre model. *Vehicle system dynamics*, 21(S1):1–18, 1992. URL <http://motion.me.ucsb.edu/book-1ns>.
- [128] Ron J Patton, Paul M Frank, and Robert N Clark. *Issues of fault diagnosis for dynamic systems*. Springer Science & Business Media, 2013.
- [129] Omar García Crespillo, Steve Langel, and Mathieu Joerger. Tight bounds for uncertain time-correlated errors with gauss–markov structure in kalman filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 59(4):4347–4362, 2023.
- [130] Steven X Ding. *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Science & Business Media, 2008.
- [131] Wanki Cho, Jangyeol Yoon, Seongjin Yim, Bongyeong Koo, and Kyongsu Yi. Estimation of tire forces for application to vehicle stability control. *IEEE Transactions on Vehicular Technology*, 59(2):638–649, 2009.

APPENDICES

Appendix A

Preliminaries and System Modeling

This chapter introduces the notational conventions and key mathematical concepts used throughout the thesis.

A.1 Mathematical Preliminaries

Notation. Throughout this thesis, boldface lowercase letters (e.g., $\mathbf{x}$) denote vectors, and boldface uppercase letters (e.g., $\mathbf{A}$) denote matrices. Scalars are denoted using non-boldface symbols (e.g., x or a). The superscript $(\cdot)^\top$ denotes the transpose of a vector or matrix.

Stochastic Process. A *stochastic process* is defined as a sequence of random vectors $\{\mathbf{x}_k\}_{k \in \mathbb{Z}_{\geq 0}}$, where each $\mathbf{x}_k \in \mathbb{R}^n$ represents the *state vector* of the process at time step k .

State Vector. Each component of the state vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$ in a stochastic process corresponds to a specific state variable of the system.

Sub-vector. For any index set $i = \{i_1, i_2, \dots, i_m\} \subseteq \{1, 2, \dots, n\}$ with $m \leq n$, the vector $\mathbf{x}^i = (x_{i_1}, x_{i_2}, \dots, x_{i_m})$ is called a *sub-vector* of $\mathbf{x}$. If $\mathbf{x}$ represents the state vector of a networked system, the superscript " i " in $\mathbf{x}^i$ denotes the subset of states associated with node i in the system.

Cross-covariance. The *cross-covariance* matrix between two random vectors $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{y}_k \in \mathbb{R}^m$ is defined as

$$\mathbb{C}[\mathbf{x}_k, \mathbf{y}_k] \triangleq \mathbb{E}[(\mathbf{x}_k - \mathbb{E}[\mathbf{x}_k])(\mathbf{y}_k - \mathbb{E}[\mathbf{y}_k])^\top],$$

where $\mathbb{E}[\cdot]$ and $\mathbb{C}[\cdot]$ denotes the expectation and covariance operators, respectively.

Auto-covariance. The *auto-covariance* matrix of a random vector $\mathbf{x}_k$ is the special case of the cross-covariance where $\mathbf{x}_k = \mathbf{y}_k$, given by

$$\mathbb{C}[\mathbf{x}_k, \mathbf{x}_k] \triangleq \mathbb{E}[(\mathbf{x}_k - \mathbb{E}[\mathbf{x}_k])(\mathbf{x}_k - \mathbb{E}[\mathbf{x}_k])^\top].$$

A.2 Graph-Theoretic Preliminaries

To precisely define the concept of estimation graph and structural influence of faults on the global fault isolation performance, some definitions and concepts of graph theory is provided as follows:

A.2.1 Graph-Related Terminology

In the following, graph-related definitions are provided based on [126].

Digraph. A *directed graph* (in short, a digraph) of order n is a pair $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set with n elements called nodes and $\mathcal{E}$ is a set of ordered pairs of nodes called edges. In a digraph, a *self-loop* is an edge from a node to itself.

Walk. A *directed walk* in a digraph is an ordered sequence of nodes such that any pair of consecutive nodes in the sequence is a directed edge of the digraph. G is *strongly connected* if there exists a directed walk from any node to any other node.

Cycle. A *cycle* (or directed cycle) in a digraph is a directed walk that starts and ends at the same node. A digraph that contains no cycles is called a directed acyclic graph (DAG).

Strongly Connected Component. A subgraph H is a strongly connected component of G if H is strongly connected and any other subgraph of G strictly containing H is not strongly connected.

Condensation Digraph. The *condensation digraph* of a digraph G , denoted by $C(G)$, is defined as follows: the nodes of $C(G)$ are the strongly connected components of G , and there exists a directed edge in $C(G)$ from node H_1 to node H_2 if and only if there exists a directed edge in G from a node of H_1 to a node of H_2 , see Figure A.1.

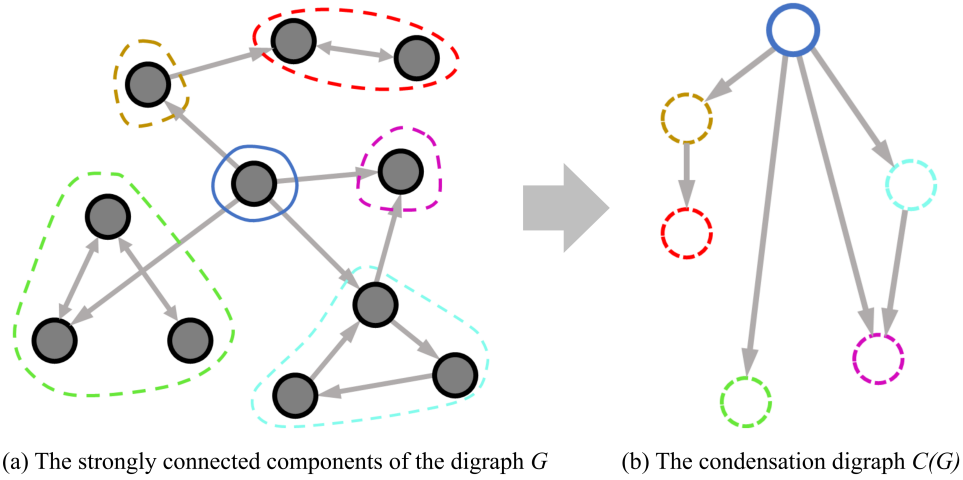


Figure A.1: An example digraph, its strongly connected components, and the corresponding condensation digraph. The condensation digraph forms an acyclic digraph.

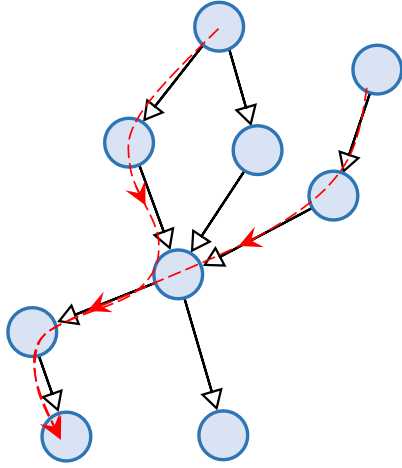
In- and Out-neighbors. In a digraph G with an edge $(u, v) \in E$, u is called an *in-neighbor* of v , and v is called an *out-neighbor* of u . In this study, $\mathcal{N}^{in}(v)$ denotes the set of in-neighbors of v , and $\mathcal{N}^{out}(v)$ denotes the set of out-neighbors of v . Note that a self-loop at a node v makes v both an in-neighbor as well as an out-neighbor of itself.

In- and Out-degree. The *in-degree* $d_{in}(v)$ and *out-degree* $d_{out}(v)$ of v are the number of in-neighbors and out-neighbors of v , respectively. Note that the degree related to the self-loops is not taken into account here.

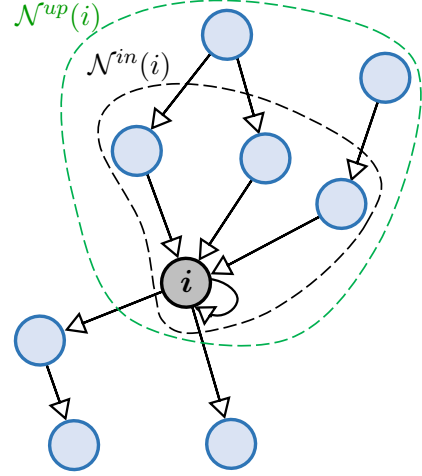
Sources and Sinks. In a digraph, every node with in-degree 0 is called a *source*, and every node with out-degree 0 is called a *sink*. Every acyclic digraph has at least one source and at least one sink.

Definition 8 (Path). *In this study, a path (or stream) is a subgraph of a directed graph that has only one sink node and contains some or all of directed walks ending in that node; see Figure A.2a.*

Definition 9 (Upstream and Downstream). *In this study, the upstream of node v , denoted by $\mathcal{N}^{up}(v)$, is the set of all nodes from which there is a directed walk to node v ; see Figure A.2b. Similarly, the downstream of node v , denoted by $\mathcal{N}^{down}(v)$, is the set of all nodes to which there is a directed walk from node v .*



(a) An example path in a DAG.



(b) Neighborhoods in an acyclic digraph.

Figure A.2: (a) A potential path (stream) in an acyclic digraph. (b) Acyclic digraph illustrating a cascaded estimation scheme. The in-neighbors of node i are represented by $\mathcal{N}^{in}(i)$, and its upstream nodes are denoted by $\mathcal{N}^{up}(i)$. For clarity, self-loops are not shown.

Adjacency Matrix. Given a directed graph $G = (\mathcal{V}, \mathcal{E})$ with $V = \{1, \dots, n\}$, the *weighted adjacency matrix* $A \in \mathbb{R}^{n \times n}$ is defined such that each entry a_{ij} represents the weight of the edge from node i to node j . If there is an edge from node i to node j , then $a_{ij} > 0$; otherwise, the corresponding matrix entry is zero. An example of a weighted digraph and its associated adjacency matrix is shown in Figure A.3.

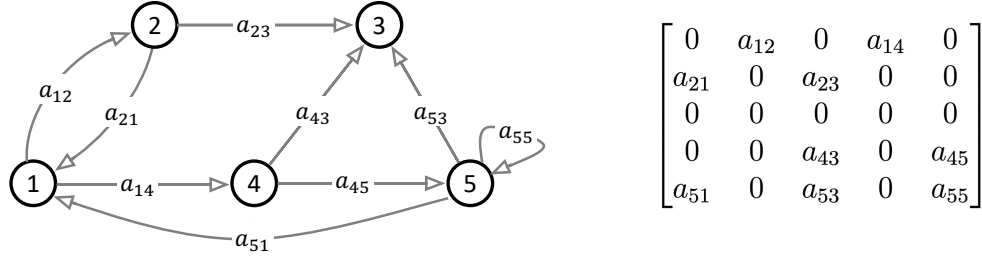


Figure A.3: A weighted digraph and its adjacency matrix.

Bipartite Graph. A *bipartite graph* is a graph whose vertex set can be partitioned into two disjoint sets $\mathcal{U}$ and $\mathcal{V}$, such that every edge connects a vertex in $\mathcal{U}$ to a vertex in $\mathcal{V}$. Formally,

$$G = (\mathcal{U}, \mathcal{V}, \mathcal{E}),$$

where $\mathcal{U}$ and $\mathcal{V}$ are disjoint sets of vertices, i.e., $\mathcal{U} \cap \mathcal{V} = \emptyset$, and $\mathcal{E} \subseteq \mathcal{U} \times \mathcal{V}$ is the set of edges, where each edge $(u, v) \in \mathcal{E}$ connects a vertex $u \in \mathcal{U}$ to a vertex $v \in \mathcal{V}$.

A weighted bipartite graph is shown in Figure A.4. In many domains networked systems, bipartite graphs are used to model interactions between two distinct types of entities.

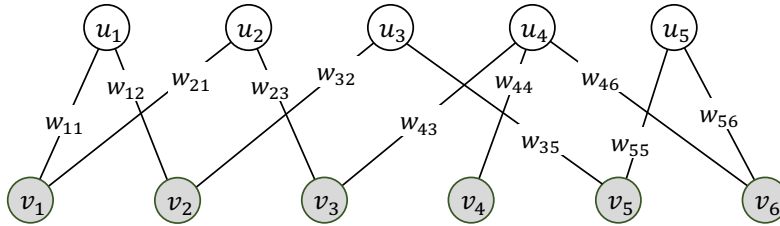


Figure A.4: A weighted bipartite graph, where w_{ij} denotes the weight of the edge between node $u_i \in \mathcal{U}$ and node $v_j \in \mathcal{V}$.

A.2.2 Topological Properties for Node Influence

In the following, two graph-theoretic concepts are defined to characterize the structural relationships between two distinct sets of nodes in a graph. These concepts reflect complementary aspects of node influence: *domination*, which captures the notion of coverage or

control over other nodes, and *independence*, which ensures non-interference by maintaining structural separation. These definitions will later be used in the performance analysis of fault isolation, particularly in contexts involving labeled and unlabeled nodes.

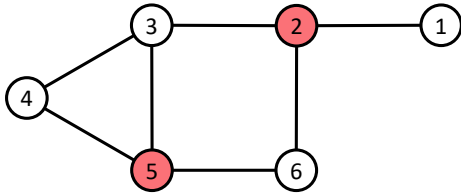
Definition 10. A k -TDS in a graph $G = (\mathcal{V}, \mathcal{E})$ is a subset $S \subseteq \mathcal{V}$ such that

$$\forall v \in \mathcal{V} \setminus S, \quad |\{u \in S \mid (u, v) \in \mathcal{E}\}| \geq k.$$

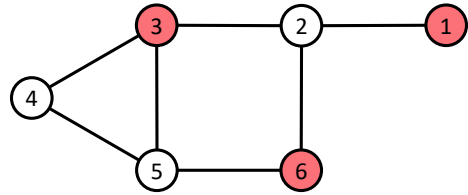
Every node not included in S must have at least k neighbors within S , ensuring redundant connectivity between the non-dominating and dominating nodes. When $k = 1$, the definition reduces to the classical *total dominating set* (TDS), in which each node has at least one neighbor in the dominating set. For a given value of k , multiple valid k -TDS configurations may exist within the graph. In this work, the interest lies in identifying a *minimum-size* k -TDS—that is, the smallest subset of nodes that satisfies the k -tuple domination condition. An illustration of such a configuration is shown in Figure A.5a.

Definition 11. A *maximum independent set* (MIS) in a graph $G = (\mathcal{V}, \mathcal{E})$ is a largest possible subset $S \subseteq \mathcal{V}$ such that no two nodes in S are adjacent; that is, $(u, v) \notin \mathcal{E}$ for all $u, v \in S$.

This structural property ensures that the nodes in the MIS are mutually non-adjacent, which can be advantageous in applications where reducing local interactions is desirable—for example, to minimize information overlap or to isolate sources of influence within the graph. An example of a MIS configuration is depicted in Figure A.5b.



(a) Nodes $\{2, 5\}$ form a 1-TDS.



(b) Nodes $\{1, 3, 6\}$ form a MIS.

Figure A.5: Illustration of node configurations: (a) a minimum-size 1-tuple dominating set and (b) a maximal independent set within a graph.

A.3 Vehicle Motion Model

This section provides a description of the vehicle's kinematic and dynamic behavior, including the governing equations and underlying physical assumptions.

A.3.1 Longitudinal Dynamics

The longitudinal motion of a four-wheel front-wheel-drive (Front-wheel drive (FWD)) vehicle, neglecting lateral tire forces and aerodynamic drag, is governed by the following equation:

$$\dot{V}_x = \frac{1}{m} \sum_{i \in \{f,r\}} \sum_{j \in \{l,r\}} F_{x,ij}, \quad (\text{A.1})$$

where V_x is the longitudinal velocity of the vehicle at the center of gravity (CG), m denotes the total vehicle mass, and $F_{x,ij}$ represents the longitudinal tire force at wheel ij . The tire force is computed using a tire model (e.g., Pacejka), which depends on the slip ratio, normal load $F_{z,ij}$, and the tire-road friction interaction.

A.3.2 Lateral Dynamics

The lateral dynamics of a four-wheel FWD vehicle are governed by the following equations of motion:

$$\begin{cases} \dot{V}_y = -V_x \dot{\psi} + \frac{1}{m} \left\{ (F_{y,fl} + F_{y,fr}) \cos \delta_f + (F_{y,rl} + F_{y,rr}) + F_{x,f} \sin \delta_f \right\} \\ \ddot{\psi} = \frac{1}{I_{zz}} \left[l_f \left\{ (F_{y,fl} + F_{y,fr}) \cos \delta_f + F_{x,f} \sin \delta_f \right\} - l_r (F_{y,rl} + F_{y,rr}) \right. \\ \left. + \frac{b_w}{2} \left\{ (F_{x,fl} - F_{x,fr}) \cos \delta_f + (F_{y,fr} - F_{y,fl}) \sin \delta_f \right\} \right] \end{cases} \quad (\text{A.2})$$

Here, V_y denotes the lateral velocity of the vehicle at the center of gravity (CG), and $\ddot{\psi}$ represent the yaw acceleration, respectively. The variables $F_{y,ij}$ and $F_{x,ij}$, as illustrated in Figure A.6c, represent the lateral and longitudinal tire forces acting at each wheel $ij \in \{fl, fr, rl, rr\}$, respectively. The total longitudinal force at the front axle is given by $F_{x,f} = F_{x,fl} + F_{x,fr}$. The symbol δ_f refers to the front axle steering angle. The parameter I_{zz} denotes the yaw moment of inertia of the vehicle. The parameter l_f and l_r represent the longitudinal distances from the vehicle frame origin to the front and rear axles, respectively, while b_w denotes the track width.

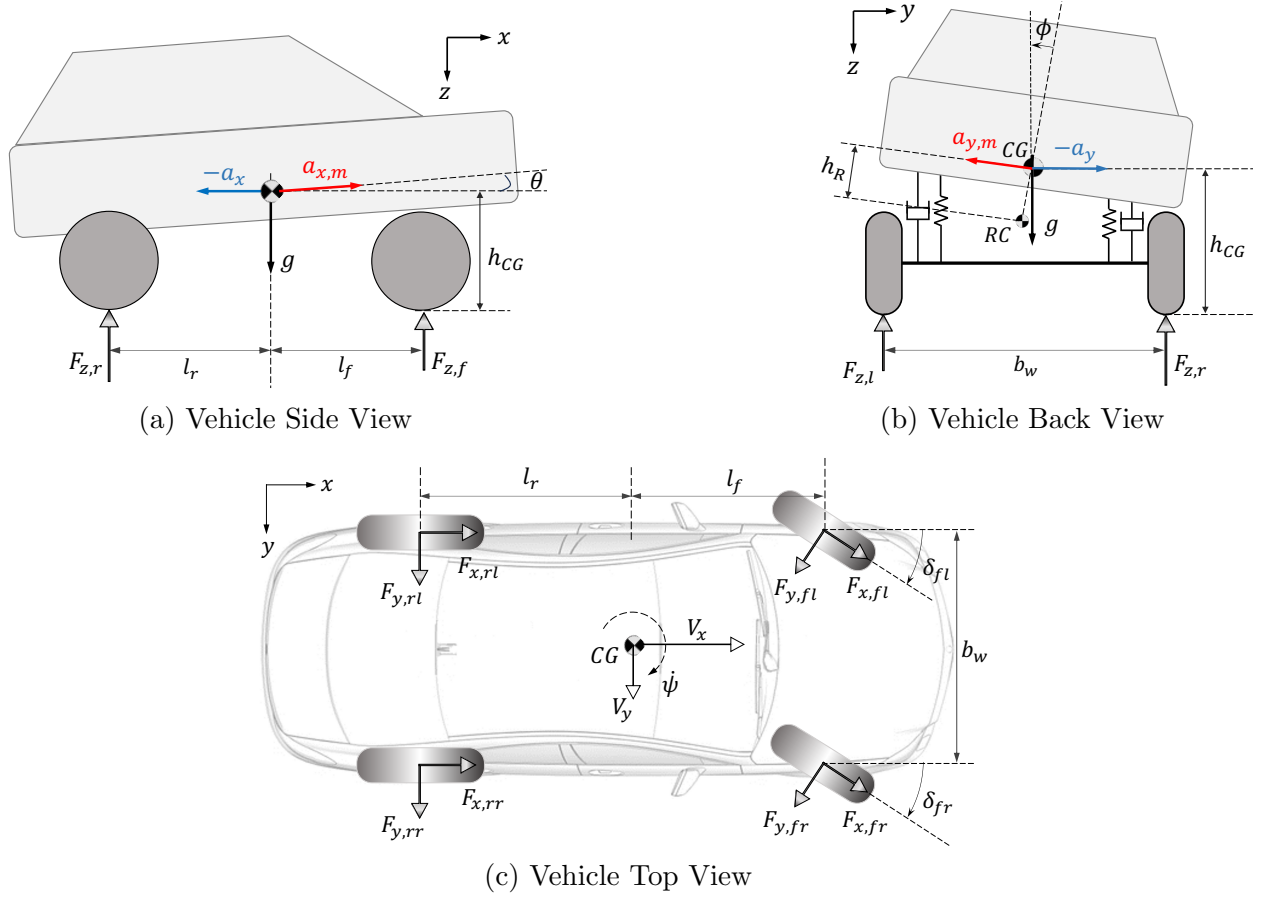


Figure A.6: Vehicle dynamics illustration and geometric definitions: (a) side view showing longitudinal acceleration and pitch dynamics, (b) rear view illustrating lateral acceleration and roll behavior, and (c) top view depicting velocity components, yaw rate, tire forces, and geometric parameters. Forces and accelerations are shown in the vehicle coordinate system, and the center of gravity (CG) is used as the reference point for motion states.

The third equation introduces a first-order tire dynamics model that accounts for transient effects in lateral force buildup, improving the fidelity of the lateral dynamics representation under aggressive maneuvers or in the presence of faults.

According to Figure A.6b, the measured lateral acceleration $a_{y,m}$ accounts for the effect of gravitational acceleration g due to the vehicle's roll angle ϕ . Note that a subscript $(\cdot)_m$ indicates that the corresponding variable represents a measured quantity. The expression for $a_{y,m}$ is given by:

$$\begin{aligned} a_{y,m} &= a_y - g \sin \phi \\ &= \frac{1}{m} \left\{ F_{x,f} \sin \delta_f + (F_{y,fl} + F_{y,fr}) \cos \delta_f + F_{y,rl} + F_{y,rr} \right\} - g \sin \phi. \end{aligned} \quad (\text{A.3})$$

A.3.3 Lateral Dynamics (Linear Bicycle Model)

The lateral dynamics of a vehicle can be approximated using a linear bicycle model, which captures the relationship between lateral motion and yaw dynamics. In this model, the left and right wheels on each axle are lumped into a single equivalent wheel, and both roll dynamics and vertical motion are neglected. The model assumes small slip angles and a constant longitudinal velocity V_x .

The state-space representation of the linear bicycle model is given by:

$$\begin{bmatrix} \dot{V}_y \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -\frac{C_f + C_r}{mV_x} & -\frac{l_f C_f - l_r C_r}{mV_x} - V_x \\ -\frac{l_f C_f - l_r C_r}{I_{zz} V_x} & -\frac{l_f^2 C_f + l_r^2 C_r}{I_{zz} V_x} \end{bmatrix} \begin{bmatrix} V_y \\ \psi \end{bmatrix} + \begin{bmatrix} \frac{C_f}{l_f m} \\ \frac{l_f C_f}{I_{zz}} \end{bmatrix} \delta \quad (\text{A.4})$$

and the corresponding measurement model is:

$$\begin{bmatrix} a_{y,m} \\ \psi_m \end{bmatrix} = \begin{bmatrix} -\frac{C_f + C_r}{mV_x} & -\frac{l_f C_f - l_r C_r}{mV_x} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_y \\ \psi \end{bmatrix} + \begin{bmatrix} \frac{C_f}{m} \\ 0 \end{bmatrix} \delta \quad (\text{A.5})$$

where C_f and C_r are the cornering stiffness coefficients.

A.3.4 Normal Load Distribution

The effective normal forces on the front and rear axles due to longitudinal acceleration are computed as:

$$F_{z,f} = m \left(\frac{l_r}{l} g - \frac{h_{CG}}{l} a_x \right), \quad F_{z,r} = m \left(\frac{l_f}{l} g + \frac{h_{CG}}{l} a_x \right), \quad (\text{A.6})$$

where a_x is the longitudinal acceleration, h_{CG} is the height of the center of gravity, and $l = l_f + l_r$ is the total wheelbase.

The axle loads are further redistributed between the left and right sides of the vehicle due to lateral acceleration. The normal tire forces at each wheel, adjusted for lateral load transfer, are given by:

$$\begin{cases} F_{z,fl} = F_{z,f} \left(\frac{1}{2} + \frac{h_{CG}}{b_w} \frac{a_y}{g} \right), \\ F_{z,fr} = F_{z,f} \left(\frac{1}{2} - \frac{h_{CG}}{b_w} \frac{a_y}{g} \right), \\ F_{z,rl} = F_{z,r} \left(\frac{1}{2} + \frac{h_{CG}}{b_w} \frac{a_y}{g} \right), \\ F_{z,rr} = F_{z,r} \left(\frac{1}{2} - \frac{h_{CG}}{b_w} \frac{a_y}{g} \right), \end{cases} \quad (\text{A.7})$$

where a_y is the lateral acceleration and b_w is the track width.

Additionally, assuming that the vertical velocity V_z is negligible, the vertical acceleration a_z of the vehicle body is governed by the equilibrium of vertical forces:

$$ma_z = \sum_{i \in \{f,r\}} \sum_{j \in \{l,r\}} F_{z,ij}, \quad (\text{A.8})$$

This relation provides a redundancy between vertical accelerometer readings and the estimated normal tire forces.

A.3.5 Pitch and Roll Dynamics

The pitch and roll dynamics of the vehicle body, illustrated in Figure A.7, are described by the following state-space representations:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_\theta}{I_{yy} + m_s h_P^2} & -\frac{c_\theta}{I_{yy} + m_s h_P^2} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{m_s h_P}{I_{yy} + m_s h_P^2} \end{bmatrix} (-a_x + g \sin(\theta + \theta_r)), \quad (\text{A.9})$$

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_\phi}{I_{xx} + m_s h_R^2} & -\frac{c_\phi}{I_{xx} + m_s h_R^2} \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{m_s h_R}{I_{xx} + m_s h_R^2} \end{bmatrix} (a_y + g \sin(\phi + \phi_r)), \quad (\text{A.10})$$

where θ and ϕ denote the pitch and roll angles of the vehicle body, while $\dot{\theta}$ and $\dot{\phi}$ represent the corresponding angular rates. The parameters I_{xx} and I_{yy} denote the moments of inertia about the roll and pitch axes, and m_s is the sprung mass. The coefficients c_θ and k_θ are the pitch damping and stiffness coefficients, respectively, while c_ϕ and k_ϕ are their roll counterparts. The geometric terms h_P and h_R represent the vertical distances from the center of gravity (CG) to the pitch center (PC) and roll center (RC), respectively. The variables θ_r and ϕ_r correspond to the road grade and road bank angles.

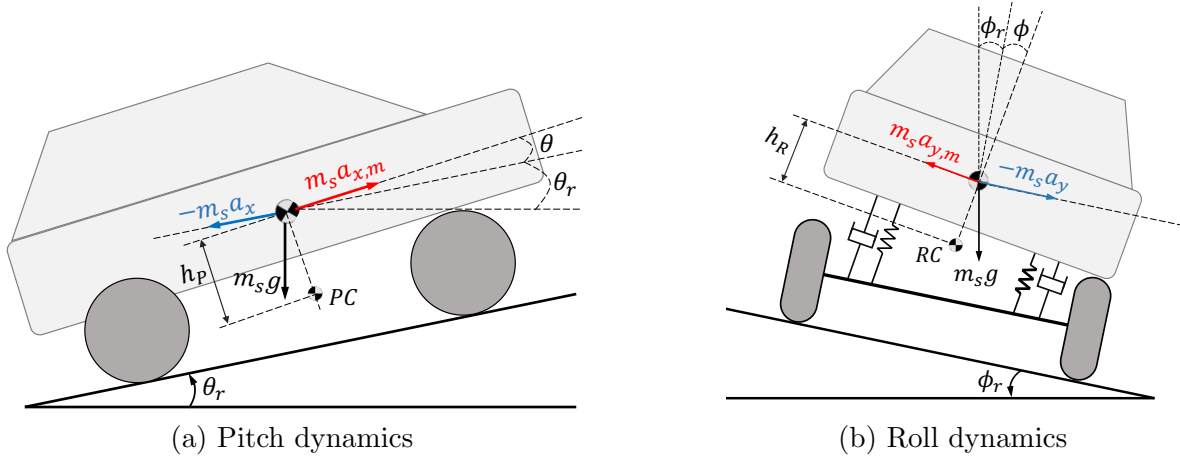


Figure A.7: Illustration of vehicle body motion due to road geometry: (a) pitch dynamics on a road with grade angle, and (b) roll dynamics on a road with bank angle.

A.3.6 Suspension Kinematics

Analytical redundancy relations for estimating suspension heights Δz_{ij} and vehicle body roll and pitch angles are presented based on the geometric configuration of suspension height sensors, as detailed in [108].

Let the vehicle frame coordinate system (x_F, y_F, z_F) be fixed to the sprung mass. The position vector of the suspension height sensor at corner $ij \in \{fl, fr, rl, rr\}$ is denoted by $\mathbf{p}_{ij} \in \mathbb{R}^3$, defined as:

$$\begin{aligned} \mathbf{p}_{fl} &= [l_f \quad b_w/2 \quad \Delta z_{fl}]^\top, & \mathbf{p}_{fr} &= [l_f \quad -b_w/2 \quad \Delta z_{fr}]^\top, \\ \mathbf{p}_{rl} &= [-l_r \quad b_w/2 \quad \Delta z_{rl}]^\top, & \mathbf{p}_{rr} &= [-l_r \quad -b_w/2 \quad \Delta z_{rr}]^\top, \end{aligned}$$

These vectors are illustrated in Figure A.8. The relative position between two corners ij

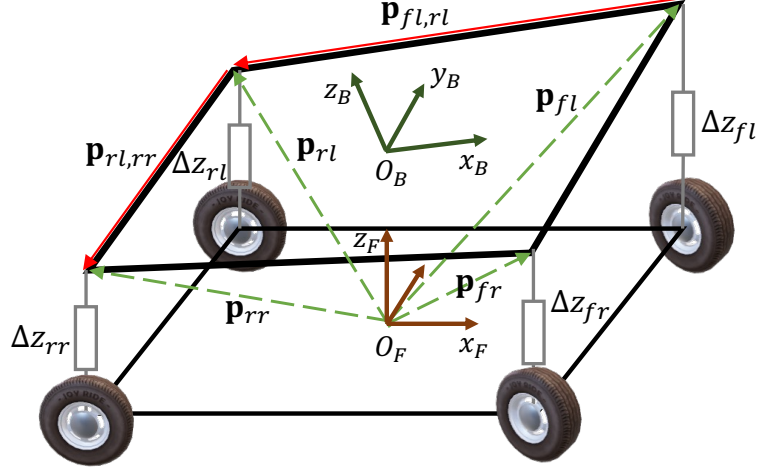


Figure A.8: Suspension kinematics and sensor positions in the vehicle frame.

and mn is given by:

$$\mathbf{p}_{ij,mn} = \mathbf{p}_{mn} - \mathbf{p}_{ij}.$$

A normal vector $\mathbf{n}_{-ij}$ to the body plane is computed using the cross product of two relative position vectors that do not involve corner ij :

$$\mathbf{n}_{-fl} = \mathbf{p}_{rl,rr} \times \mathbf{p}_{rr,fr},$$

$$\mathbf{n}_{-fr} = \mathbf{p}_{fl,rl} \times \mathbf{p}_{rl,rr},$$

$$\mathbf{n}_{-rl} = \mathbf{p}_{rr,fr} \times \mathbf{p}_{fr,fl},$$

$$\mathbf{n}_{-rr} = \mathbf{p}_{fl,rl} \times \mathbf{p}_{fl,fr},$$

and the corresponding scalar offset $\mathbf{m}_{-ij}$ is computed as:

$$\mathbf{m}_{-fl} = -\mathbf{n}_{-fl}^\top \mathbf{p}_{fr},$$

$$\mathbf{m}_{-fr} = -\mathbf{n}_{-fr}^\top \mathbf{p}_{rl},$$

$$\mathbf{m}_{-rl} = -\mathbf{n}_{-rl}^\top \mathbf{p}_{rr},$$

$$\mathbf{m}_{-rr} = -\mathbf{n}_{-rr}^\top \mathbf{p}_{fl}.$$

Using $\mathbf{n}_{-ij}$ and $\mathbf{m}_{-ij}$, the estimated suspension height at corner ij is obtained by:

$$\widehat{\Delta z}_{ij} = -\frac{\mathbf{n}_{x,-ij} p_{x,ij} + \mathbf{n}_{y,-ij} p_{y,ij} + \mathbf{m}_{-ij}}{\mathbf{n}_{z,-ij}}. \quad (\text{A.11})$$

The corresponding roll and pitch angles of the vehicle body, excluding height sensor ij , are estimated by projecting the body plane normal onto the vehicle frame axes:

$$\begin{aligned}\hat{\phi}_{-ij} &= -\arcsin\left(\frac{\mathbf{n}_{y,-ij}}{\|\mathbf{n}_{-ij}\|}\right), \\ \hat{\theta}_{-ij} &= -\arcsin\left(\frac{\mathbf{n}_{x,-ij}}{\|\mathbf{n}_{-ij}\|}\right).\end{aligned}\tag{A.12}$$

A.3.7 Tire Model

The longitudinal and lateral tire-road interaction forces, $F_{x,ij}$ and $F_{y,ij}$, can be modeled using the well-established Pacejka tire model, commonly referred to as the Magic Formula [127].

The longitudinal force is modeled as:

$$F_{x,ij} = \mu_r F_{z,ij} D_x \sin\left[C_x \arctan\left\{B_x \lambda_{ij} - E_x(B_x \lambda_{ij} - \arctan(B_x \lambda_{ij}))\right\}\right]\tag{A.13}$$

where μ denotes the road friction coefficient, $F_{z,ij}$ is the normal load on the ij -th wheel, λ_{ij} is the slip ratio, and B_x, C_x, D_x, E_x are model parameters (see Figure A.9).

The slip ratio λ_{ij} represents the relative motion between the tire's rotational and translational velocities and is given by:

$$\lambda_{ij} = \begin{cases} 0, & \text{if } \max(R_e \omega_{ij}, V_{x,ij}) = 0, \\ \frac{R_e \omega_{ij} - V_{x,ij}}{V_{x,ij}}, & \text{otherwise} \end{cases}\tag{A.14}$$

where $R_e \omega_{ij}$ is the circumferential speed of the wheel and $V_{x,ij}$ is the longitudinal velocity of the wheel center in its local frame. The transformation from vehicle CG velocity to corner velocities is expressed as:

$$\begin{cases} V_{x,fl} = (V_x + \frac{b_w}{2}\dot{\psi}) \cos \delta_{fl} + (V_y + l_f \dot{\psi}) \sin \delta_{fl} \\ V_{x,fr} = (V_x - \frac{b_w}{2}\dot{\psi}) \cos \delta_{fr} + (V_y + l_f \dot{\psi}) \sin \delta_{fr} \\ V_{x,rl} = V_x + \frac{b_w}{2}\dot{\psi} \\ V_{x,rr} = V_x - \frac{b_w}{2}\dot{\psi} \end{cases}\tag{A.15}$$

where V_x , V_y are the longitudinal and lateral velocities, $\dot{\psi}$ is the yaw rate, and δ_{ij} denotes the steering angle. The effect of V_y is negligible and assumed to be zero.

Similarly, the lateral force is given by:

$$F_{y,ij} = \mu_r F_{z,ij} D_y \sin \left[C_y \arctan \left\{ B_y \alpha_{ij} - E_y (B_y \alpha_{ij} - \arctan(B_y \alpha_{ij})) \right\} \right] \quad (\text{A.16})$$

where α_{ij} is the slip angle and B_y, C_y, D_y, E_y are the lateral force parameters. For FWD vehicles, the slip angles for the front and rear wheels are:

$$\begin{cases} \alpha_{fl,fr} = \delta_{fl,fr} - \arctan \left(\frac{V_y + l_f \dot{\psi}}{V_x} \right) \\ \alpha_{rl,rr} = - \arctan \left(\frac{V_y - l_r \dot{\psi}}{V_x} \right) \end{cases} \quad (\text{A.17})$$

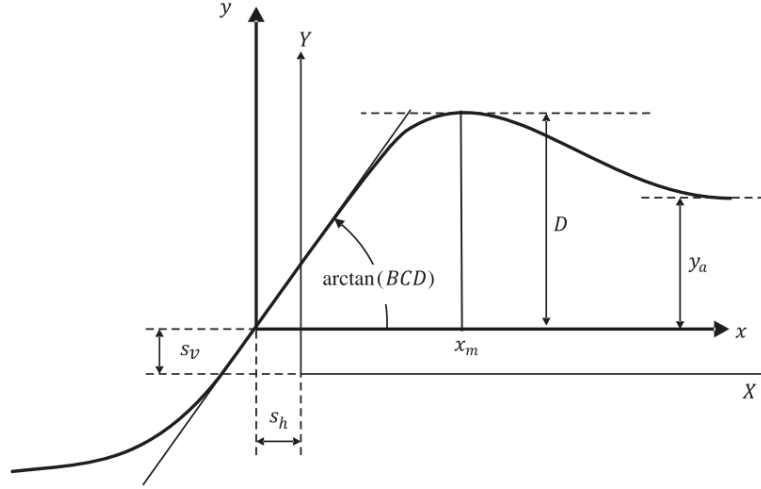


Figure A.9: Interpretation of the coefficients of Magic Formula tire model [1].

A.3.8 Wheel Dynamics

The rotational dynamics of each wheel are described by:

$$\dot{\omega}_{ij} = \frac{1}{I_w} (T_{d,ij} - T_{b,ij} - R_e F_{x,ij}), \quad (\text{A.18})$$

where ω_{ij} is the angular velocity of wheel ij , $T_{d,ij}$ and $T_{b,ij}$ denote the applied drive and braking torques, respectively, and I_w is the wheel moment of inertia. The parameter R_e represents the effective wheel radius, and $F_{x,ij}$ is the longitudinal force acting at wheel ij .

A.4 Fault Modeling

In this section, the terminology used in state estimation and fault diagnosis is clearly defined. In the literature, the definitions and statements for different aspects of state estimation and fault diagnosis are indeterminate and unclear in some cases. For example, the terms like uncertainty or error have been interpreted differently in different works. Besides, in the context of fault detection and isolation, there are overlaps and somewhat misuse between some types of faults and uncertainties. Thus, in order to avoid inconsistency and ambiguity in the terminology, it seems necessary to provide a precise and clear definition of the terms.

State Uncertainty: The uncertainty $\tilde{\varepsilon}$ of a random vector-valued variable $\mathbf{x}$ is defined as the difference between its actual and modeled values, and its probabilistic properties may be uncertain [27]. In this study, the uncertainty is modeled as:

$$\tilde{\varepsilon}(\cdot) = \varepsilon(\cdot) + \mathbf{b} \quad (\text{A.19})$$

where ε is a zero-mean noise vector and $\mathbf{b}$ is the bias error with known probabilistic properties. Similarly, the uncertainty of estimation is defined as the difference between actual and estimated states. In this study, estimation uncertainty and estimation error are used interchangeably.

Bias Error: The bias error of a variable in (A.19) is viewed as an uncertain but constant disturbance. The bias error $\mathbf{b}$ is modeled as a random vector with

$$\begin{aligned} \mathbb{E}[\mathbf{b}] &= \mathbf{0} \\ \mathbb{E}[\mathbf{b}\mathbf{b}^\top] &= \mathbf{B} \end{aligned} \quad (\text{A.20})$$

Note that a bias is a constant disturbance whose value is uncertain and is different from a nonzero mean disturbance with a known, constant mean value [27]. If the bias error is assumed to be known, then its probabilistic properties will change to:

$$\begin{aligned} \mathbb{E}[\mathbf{b}] &= \mathbf{b} \\ \mathbb{E}[\mathbf{b}\mathbf{b}^\top] &= \mathbf{0} \end{aligned} \quad (\text{A.21})$$

Excessive Noise: If the variance of a zero-mean noise term in (A.19) exceeds an acceptable threshold, it is referred to as excessive noise. This term is used because, in dynamical systems, a certain level of process or measurement noise is typically tolerable and has negligible impact on system performance. However, when the noise variance surpasses a critical level, it can significantly interfere with the system's intended operation, potentially leading to malfunction or degraded performance. If the noise process $\{\varepsilon_k\}$ is assumed to be a zero-mean white noise sequence, it satisfies the following properties:

$$\begin{aligned}\mathbb{E}[\varepsilon_k] &= \mathbf{0} \\ \mathbb{E}[\varepsilon_k \varepsilon_j] &= \mathbf{0}, \quad k \neq j \\ \mathbb{E}[\varepsilon_k \varepsilon_k^\top] &= \Sigma_k\end{aligned}\tag{A.22}$$

where Σ_k denotes the noise variance.

Model Uncertainty: Modeling uncertainty in a dynamical system can be taken into account by adding some terms in the mathematical model of the system. These additional terms are assumed to have a certain structure, such as the bias error or noise. This type of uncertainty is called structured uncertainty and is assumed generally to have known *a priori*. However, the modeling uncertainties are beyond the structured uncertainties, and for most real systems, the structure of the uncertainty is unknown. The uncertainties in the system parameters, errors due to non-linearity, and unmodeled dynamics are defined as unstructured uncertainties. To handle uncertainties introduced through nonlinear parameters or states, linearization techniques such as Taylor series expansion or perturbation methods can be employed, under the assumption that the uncertainty magnitude is small relative to the nominal signal value.

A general dynamical system with modeling uncertainty was originally presented by Patton et al. [128]; the model adopted here includes slight adjustments to the definitions and structure to better suit the current context:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{R}_1\mathbf{f} + \boldsymbol{\omega} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} + \mathbf{R}_2\mathbf{f} + \boldsymbol{\nu}\end{aligned}\tag{A.23}$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the system input vector, $\mathbf{y}$ is the measurement vector, $\boldsymbol{\omega}$ represents the process noise, $\boldsymbol{\nu}$ denotes the measurement noise, and $\mathbf{f}$ is the system fault vector. Matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are the state-space matrices. The fault entry matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ represent the effect of faults on the process and output of the system, respectively. Although the structured uncertainties $\boldsymbol{\omega}$ and $\boldsymbol{\nu}$, and the fault terms $\mathbf{R}_1\mathbf{f}$ and $\mathbf{R}_2\mathbf{f}$, affect the system in a similar mathematical form, they differ fundamentally in nature. Faults are

typically abrupt and identifiable malfunctions in system components, while uncertainties represent inherent model inaccuracies or external variations that are often persistent and less predictable.

Fault: In the formulation (A.23), the fault vector $\mathbf{f}$ encompasses sensor faults, actuator faults (treated as internal disturbances), external disturbances. For clarity, the term $\mathbf{R}_1\mathbf{f}$ in (A.23) is referred to as the process fault, denoted by $\mathbf{f}_d$, while $\mathbf{R}_2\mathbf{f}$ is referred to as the observation fault, denoted by $\mathbf{f}_b$.

According to [49, 50], anomalous sensor behavior can be categorized into six main types: 1) *spike*—an abrupt, sharp change in the recorded data; 2) *constant*—a period of unchanging measurement values regardless of actual conditions; 3) *drift*—a slow, gradual deviation in sensor output over time; 4) *bias*—a consistent offset in the measurements; 5) *noise*—a significant increase in the variance of sensor data; and 6) *miss*—intervals where data are absent or unavailable.

Generally, the bandwidth of estimators can be adjusted to mitigate intermittent high-amplitude faults, like spikes. Data miss or constant value faults are easily detectable and can be mitigated using redundant virtual or physical sensors. However, bias error and excessive noise are common soft fault often persist without rendering the sensor completely unusable; therefore, they require more investigation and mitigation strategies. Similarly, actuator faults and external disturbances can also be considered as bias and noise in the process model. Therefore, this study assumes the fault types to be either bias or excessive noise, characterized by the properties defined in (A.20) and (A.22), respectively. The process fault $\mathbf{f}_d \in \mathbb{R}^p$ and output fault $\mathbf{f}_b \in \mathbb{R}^q$ is assumed to have the following known properties:

$$\begin{aligned}\mathbb{E}[\mathbf{f}_d] &= \mathbf{0} \ , \quad \mathbb{E}[\mathbf{f}_d\mathbf{f}_d^\top] = \mathbf{D} \\ \mathbb{E}[\mathbf{f}_b] &= \mathbf{0} \ , \quad \mathbb{E}[\mathbf{f}_b\mathbf{f}_b^\top] = \mathbf{B}\end{aligned}\tag{A.24}$$

In this study, since both fault terms and structured uncertainties influence the system in the same manner as described in (A.23), and faults are modeled probabilistically with known distributions, they are treated as structured uncertainties for the purpose of evaluating model-based estimation performance. Hence, the term uncertainty quantification is used to assess the impact of faults on estimator performance.

Gauss-Markov Process for Temporally Correlated Faults. The Gauss-Markov process is widely used in the control and navigation communities to model sensor bias instability and temporally correlated errors in IMU [124]. Unlike idealized white noise or constant biases, sensor errors often exhibit slow variations over time, which are better captured by this process.

The Gauss-Markov process is a first-order autoregressive process that exhibits temporal correlation and bounded variance. In discrete-time, the process is defined by the following stochastic difference equation:

$$f_{k+1} = \alpha f_k + \sigma_d \sqrt{1 - \alpha^2} w_k, \quad (\text{A.25})$$

where $f_k \in \mathbb{R}$ is the fault process at time step k , $\alpha \in (0, 1)$ controls the memory (correlation) of the process, and $w_k \sim \mathcal{N}(0, 1)$ is standard Gaussian white noise. The parameter α is related to the correlation time τ of the process by $\alpha = e^{-\Delta t/\tau}$, where Δt is the sampling interval [129]. As $\tau \rightarrow 0$, $\alpha \rightarrow 0$, and the process converges to white noise. As $\tau \rightarrow \infty$, $\alpha \rightarrow 1$, and the process becomes a random walk. Thus, the Gauss-Markov process provides a tunable spectrum between white noise and unbounded drift.

The Gauss-Markov process has a bounded steady-state variance, with zero mean $\mathbb{E}[f_k] = 0$ and covariance $\mathbb{E}[f_k f_k^\top] = \sigma_d^2$. These properties align with the assumptions in (A.24), allowing it to be integrated into the proposed uncertainty propagation framework, which relies only on known second-order statistics without requiring independence or whiteness assumptions.

A.5 Fault Detectability and Isolability

Consider the state-space model with additive faults $\mathbf{f}$:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{R}_1\mathbf{f} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} + \mathbf{R}_2\mathbf{f} \end{aligned} \quad (\text{A.26})$$

where the fault entry matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ represent the effect of faults on the process and output of the system, respectively.

The transfer function from the fault input $\mathbf{f}(s)$ to the residual $\mathbf{r}(s)$, defined as fault transfer matrix $\mathbf{G}_{rf}(s)$, is given by:

$$\mathbf{G}_{rf}(s) = \frac{\mathbf{R}(s)}{\mathbf{F}(s)} = [\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{R}_1 + \mathbf{R}_2] \quad (\text{A.27})$$

A.5.1 Fault Detectability Condition

In order to detect the i -th fault f_i in the residual vector $\mathbf{r} := \mathbf{y} - \hat{\mathbf{y}}$, the i -th column of the fault transfer matrix $\mathbf{G}_{rf}(s)$ should be non-zero:

$$[\mathbf{G}_{rf}(s)]_i = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{R}_{1,i} + \mathbf{R}_{2,i} \neq 0 \quad (\text{A.28})$$

where $\mathbf{R}_{1,i}$ and $\mathbf{R}_{2,i}$ are the i -th column of $\mathbf{R}_1$ and $\mathbf{R}_2$, respectively.

In fault detection theory, this condition corresponds to *weak detectability*, which means that the fault has some influence on the residual — that is, the residual is not identically zero for a nonzero fault input. The condition above ensures that fault f_i affects the residual $\mathbf{r}$ at some frequency s , implying the existence of a nonzero fault signature. However, the influence of a fault on the residual may be only transient—appearing briefly and then vanishing—while its impact on the system state persists [117]. This implies that although a fault may be weakly detectable in principle, reliable fault detection cannot be guaranteed based solely on this condition. To address this limitation, the notion of *strong detectability* is introduced. A fault is said to be *strongly detectable* if it produces a sustained, non-vanishing effect on the residual signal. Mathematically, the i -th fault f_i is strongly detectable if the following condition holds:

$$[\mathbf{G}_{rf}(0)]_i = -\mathbf{CA}^{-1}\mathbf{R}_{1,i} + \mathbf{R}_{2,i} \neq 0. \quad (\text{A.29})$$

If this condition is satisfied, then the i -th fault f_i is guaranteed to have a nonzero steady-state effect on the residual $\mathbf{r}$, thereby making it strongly detectable.

A.5.2 Fault Isolability Condition

Fault isolability refers to the ability to distinguish among multiple faults once they have been detected. While a single residual signal may suffice for fault detection, isolating specific faults typically requires a structured set of residuals, each designed to be sensitive to certain faults and insensitive to others. This concept underpins the design of *structured residual sets*, where each residual is tailored to exhibit a unique sensitivity pattern across faults.

A simple isolable structure can be formed using a *dedicated residual set*, in which each residual is sensitive to only one fault. However, this design offers limited flexibility and is often impractical due to a lack of design degrees of freedom. A more practical and widely used approach is the *generalized residual set*, in which each residual is constructed to be sensitive to all faults except one. This configuration enables unique fault isolation using a Boolean decision table and simple threshold logic. Specifically, a fault f_i is declared active if

$$r_i(t) \leq T_i \quad \text{and} \quad r_j(t) > T_j \quad \forall j \neq i,$$

where T_i is the threshold associated with the i -th residual. This logic allows for reliable isolation in the presence of modeling uncertainties and multiple fault sources.

Fault isolability is a stronger requirement than fault detectability. While detectability ensures that the presence of a fault can be recognized, isolability further demands that the specific location or source of the fault within the system can be uniquely identified. For the linear system described by equations (2.3)–(2.4), all faults $f_i(t) \in \mathbf{f}(t)$, for $i = 1, \dots, \ell$, are said to be *isolable* if and only if the following rank condition is satisfied:

$$\text{rank} [\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{R}_1 + \mathbf{R}_2] = \ell, \quad (\text{A.30})$$

where ℓ is the number of faults in the fault vector $\mathbf{f}$ [130]. This condition ensures that each fault has a unique transfer function to the system output, enabling unambiguous fault isolation. In other words, the effects of individual faults must be distinguishable at the output level for complete fault isolability to be achieved.

Furthermore, the isolability condition imposes a fundamental constraint on the number of faults that can be isolated in practice. Specifically, to isolate ℓ different faults, the measurement space must contain at least an ℓ -dimensional subspace spanned by the fault signatures. Since the rank of the fault-to-output transfer matrix is bounded above by the number of sensors m , i.e.,

$$\text{rank} [\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{R}_1 + \mathbf{R}_2] \leq \min\{m, \ell\},$$

it follows that additive faults are isolable only if their total number does not exceed the number of available sensors. This condition provides a useful and easily verifiable guideline in practical fault isolation problems.

A.5.3 Fault Identifiability Condition

Given system (A.26) and the corresponding fault transfer matrix

$$\mathbf{G}_{rf}(s) = [\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{R}_1 + \mathbf{R}_2],$$

the fault vector $\mathbf{f} = [f_1 \ f_2 \ \dots \ f_\ell]^\top$ is called identifiable if $\mathbf{G}_{rf}(s)$ is invertible and its inverse is stable and causal [130].

Appendix B

Estimation Uncertainty Quantification in a Luenberger Observer

The uncertainty propagation for a general linear state observer, also referred to as the Luenberger observer, is first obtained here. Then, the uncertainty propagation for the example of longitudinal tire-road force is derived.

Consider the linear discrete-time state equation

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{u}_k + \mathbf{\Gamma}\zeta_k \\ \mathbf{y}_k &= \mathbf{H}\mathbf{x}_k + \boldsymbol{\eta}_k\end{aligned}\tag{B.1}$$

where $\mathbf{F}$, $\mathbf{G}$, $\mathbf{\Gamma}$, and $\mathbf{H}$ are given deterministic matrices; and the noise processes ζ_k and η_k are zero-mean, white, mutually uncorrelated with covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectively. The Luenberger state observer is given by:

$$\hat{\mathbf{x}}_{k+1} = (\mathbf{F} - \mathbf{L}\mathbf{H})\hat{\mathbf{x}}_k + \mathbf{G}\mathbf{u}_k + \mathbf{L}\mathbf{y}_k\tag{B.2}$$

where $\hat{\mathbf{x}}_k$ is the estimated state and $\mathbf{L}$ is the observer gain. Let us define the estimation error $\tilde{\mathbf{x}}_k$ as

$$\tilde{\mathbf{x}}_k := \mathbf{x}_k - \hat{\mathbf{x}}_k\tag{B.3}$$

then, the dynamics of estimation error will be

$$\tilde{\mathbf{x}}_{k+1} = (\mathbf{F} - \mathbf{L}\mathbf{H})\tilde{\mathbf{x}}_k + \mathbf{\Gamma}\zeta_k + \mathbf{L}\eta_k\tag{B.4}$$

while assuming that the noise processes ζ_k and η_k are zero-mean, it yields

$$\mathbb{E}[\tilde{\mathbf{x}}_{k+1}] = (\mathbf{F} - \mathbf{L}\mathbf{H})\mathbb{E}[\tilde{\mathbf{x}}_k]\tag{B.5}$$

If the pair $(\mathbf{A}, \mathbf{H})$ is observable, then the mean of error dynamics, $\mathbb{E}[\tilde{\mathbf{x}}_k]$, can be controlled by a proper choice of the observer gain $\mathbf{L}$.

The estimation error covariance $\mathbf{P}_k$, defined as $\mathbb{E}[\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^\top]$, can be propagated using the *push-forward* approach as follows:

$$\mathbf{P}_{k+1} = (\mathbf{F} - \mathbf{LH})\mathbf{P}_k(\mathbf{F} - \mathbf{LH})^\top + \mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^\top + \mathbf{L}\mathbf{R}_k\mathbf{L}^\top \quad (\text{B.6})$$

The propagation of uncertainty for the linear observers is very similar to the Kalman filter. In fact, the Kalman filter is a particular case of the linear observer where its gain, $\mathbf{L}$, is optimal in the sense that it minimizes the estimation error covariance. Hence, the same strategy as Section 4.2 can be adopted to propagate the uncertainty in any arbitrary estimation path with linear observers.

Similarly, the propagation of bias error in a linear observer can be obtained. Consider the linear discrete-time state equation

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{u}_k + \mathbf{\Upsilon}\mathbf{b}_\zeta \\ \mathbf{y}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{b}_\eta \end{aligned} \quad (\text{B.7})$$

where $\mathbf{F}$, $\mathbf{G}$, $\mathbf{\Gamma}$, and $\mathbf{H}$ are given deterministic matrices; and the random constant bias errors $\mathbf{b}_\zeta$ and $\mathbf{b}_\eta$ are assumed to have the same properties as described in (A.20), while $\mathbb{E}[\mathbf{b}_\zeta \mathbf{b}_\zeta^\top] = \mathbf{B}_\zeta$ and $\mathbb{E}[\mathbf{b}_\eta \mathbf{b}_\eta^\top] = \mathbf{B}_\eta$.

The associated estimation error covariance $\mathbf{B}_k$, defined as $\mathbb{E}[\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^\top]$, can be propagated as follows:

$$\begin{aligned} \mathbf{B}_{k+1} &= (\mathbf{F} - \mathbf{LH})\mathbf{B}_k(\mathbf{F} - \mathbf{LH})^\top + \mathbf{\Upsilon}\mathbf{B}_\zeta\mathbf{\Upsilon}^\top + \mathbf{L}\mathbf{B}_\eta\mathbf{L}^\top + \\ &\quad (\mathbf{F} - \mathbf{LH})\mathbf{S}_k\mathbf{B}_\zeta\mathbf{\Upsilon}^\top + (\cdots)^\top + (\mathbf{F} - \mathbf{LH})\mathbf{T}_k\mathbf{B}_\eta\mathbf{L}^\top + (\cdots)^\top \end{aligned} \quad (\text{B.8})$$

where $\mathbf{S}_k$ and $\mathbf{T}_k$ can be obtained using the following recursive equations:

$$\begin{aligned} \mathbf{S}_{k+1} &= (\mathbf{F} - \mathbf{LH})\mathbf{S}_k + \mathbf{\Upsilon} \\ \mathbf{T}_{k+1} &= (\mathbf{F} - \mathbf{LH})\mathbf{T}_k - \mathbf{L} \end{aligned} \quad (\text{B.9})$$

where the initial values are set as $\mathbf{S}_0 = 0$ and $\mathbf{T}_0 = 0$.

To sum up, the uncertainty propagation for a general linear observer is obtained; however, some linear observers cannot be described in the format of the well-known Luenberger observer. Thus, the uncertainty propagation for other types of linear observers needs to be addressed. Without loss of generality due to the linearity of such observers, a similar *push-forward* approach can be taken to easily propagate the uncertainty through any linear observers. In the following, the uncertainty propagation for a linear observer estimating the longitudinal tire-road force is provided as another example of the basic *push-forward* approach.

B.1 Uncertainty Propagation in Longitudinal Force Observer via push-forward approach

The uncertainty propagation for the longitudinal tire-road force observer is presented here.

Consider the wheel dynamics

$$I_w \dot{\omega} = T_d - T_b - R_e \hat{F}_x \quad (\text{B.10})$$

where ω is wheel angular speed, F_x is the longitudinal tire force, T_b is the brake torque, and T_d is the drive torque for each wheel. The parameters I_w and R_e represent wheel spin inertia and wheel effective radius, respectively. The linear recursive observer proposed in [131] to estimate the longitudinal tire force relies on the above wheel dynamics and is obtained as:

$$\begin{aligned} \hat{F}_x &= -\frac{1}{R_e} [I_w \dot{\omega}_m + \lambda(\omega_m - \hat{\omega}) + T_{b,m} - T_{d,m}] \\ \hat{\omega} &= T_{d,m} - T_{b,m} - R_e \hat{F}_x \end{aligned} \quad (\text{B.11})$$

where $\hat{F}_x$ is the estimated longitudinal tire force, $\hat{\omega}$ is the estimated wheel speed, as well as ω_m , $T_{b,m}$, and $T_{d,m}$ are the measured wheel speed, measured brake torque, and measured drive torque, respectively. The observer gain, λ , is obtained by satisfying the observer stability and performance criteria.

To propagate measurement uncertainties, this continuous-time observer is first converted to discrete-time equations. Consider a bias error in measuring ω and a non-zero mean white noise as the measurement uncertainty of $T_b - T_d$ with the following properties:

$$\begin{aligned} \mathbb{E}[\varepsilon_{T,k}] &= b_T, & \mathbb{V}[\varepsilon_{T,k}] &= R_T \\ \mathbb{E}[\varepsilon_{\omega,k}] &= b_\omega, & \mathbb{V}[\varepsilon_{\omega,k}] &= 0 \end{aligned} \quad (\text{B.12})$$

where b_T and b_ω represent the bias error in measurements $T_{b,m} - T_{d,m}$ and ω_m , respectively. After discretization over the time step Δt , the estimation error for the longitudinal tire force, defined as $\tilde{F}_x := F_x - \hat{F}_x$, can be calculated recursively by:

$$\begin{aligned} \tilde{F}_{x,k} &= -\frac{I_w}{R_e} [\varepsilon_{\dot{\omega},k} - \lambda(\tilde{\omega}_k - \varepsilon_{\omega,k})] + \frac{1}{R_e} \varepsilon_{T,k} \\ \tilde{\omega}_k &= \tilde{\omega}_{k-1} + \frac{\Delta t}{I_w} \left[\varepsilon_{T,k-1} - R_e \tilde{F}_{x,k-1} \right] \end{aligned} \quad (\text{B.13})$$

where $\tilde{\omega}$ defined as $\omega - \hat{\omega}$ is the estimation error of wheel speed. In addition, $\varepsilon_{\dot{\omega},k}$ represents the uncertainty of $\dot{\omega}_m$, calculated by discrete differentiation of ω_m .

Finally, the mean of estimation error for $\tilde{F}_x$ can be propagated by *push-forward* approach as follows:

$$\begin{aligned}\mathbb{E}[\tilde{F}_{x,k}] &= \frac{\lambda I_w}{R_e} (\mathbb{E}[\tilde{\omega}_k] - b_\omega) + \frac{1}{R_e} b_T \\ \mathbb{E}[\tilde{\omega}_k] &= \mathbb{E}[\tilde{\omega}_{k-1}] + \frac{\Delta t}{I_w} \left(b_T - R_e \mathbb{E}[\tilde{F}_{x,k-1}] \right).\end{aligned}\tag{B.14}$$

Similarly, the variance of estimation error $\tilde{F}_x$ due to the noise in measuring $T_b - T_d$ can be propagated by *push-forward* approach as follows:

$$\begin{aligned}\mathbb{V}[\tilde{F}_{x,k}] &= \frac{\lambda^2 I_w^2}{R_e^2} \mathbb{V}[\tilde{\omega}_k] + \frac{1}{R_e^2} R_T \\ \mathbb{V}[\tilde{\omega}_k] &= (1 - 2\lambda\Delta t) \mathbb{V}[\tilde{\omega}_{k-1}] - \frac{\Delta t^2}{I_w^2} R_T + \frac{R_e^2 \Delta t^2}{I_w^2} \mathbb{V}[\tilde{F}_{x,k-1}]\end{aligned}\tag{B.15}$$

Appendix C

First- and Second-Order Moments under PCE for Vector-Valued Functions

Let $\mathbf{X} \in \mathbb{R}^d$ be a random vector with joint probability density function $p(\mathbf{x})$, and let $\mathbf{f}(\mathbf{x}) \in L^2(p(\mathbf{x}); \mathbb{R}^m)$ be a square-integrable vector-valued function. Suppose $\mathbf{f}(\mathbf{x})$ admits a generalized Polynomial Chaos Expansion (gPC) of the form:

$$\mathbf{f}(\mathbf{x}) = \sum_{i=0}^{\infty} \mathbf{c}_i \varphi_i(\mathbf{x}),$$

where $\{\varphi_i(\mathbf{x})\}$ is a set of multivariate orthonormal polynomials with respect to $p(\mathbf{x})$, and $\mathbf{c}_i \in \mathbb{R}^m$ are the vector-valued coefficients.

Expectation. The expectation of $\mathbf{f}(\mathbf{x})$ is defined as:

$$\mathbb{E}_p[\mathbf{f}(\mathbf{x})] = \int_{\Omega} \mathbf{f}(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}.$$

Substituting the gPC expansion:

$$\mathbb{E}_p[\mathbf{f}(\mathbf{x})] = \int_{\Omega} \left(\sum_{i=0}^{\infty} \mathbf{c}_i \varphi_i(\mathbf{x}) \right) p(\mathbf{x}) d\mathbf{x} = \sum_{i=0}^{\infty} \mathbf{c}_i \int_{\Omega} \varphi_i(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}.$$

By orthonormality:

$$\int_{\Omega} \varphi_i(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \begin{cases} 1 & \text{if } i = 0, \\ 0 & \text{if } i \geq 1, \end{cases}$$

which gives:

$$\mathbb{E}_p[\mathbf{f}(\mathbf{x})] = \mathbf{c}_0.$$

Covariance Matrix. The covariance matrix of $\mathbf{f}(\mathbf{x})$ is defined as:

$$\mathbb{C}_p[\mathbf{f}(\mathbf{x})] = \mathbb{E}_p [(\mathbf{f}(\mathbf{x}) - \mathbb{E}_p[\mathbf{f}(\mathbf{x})])(\mathbf{f}(\mathbf{x}) - \mathbb{E}_p[\mathbf{f}(\mathbf{x})])^\top].$$

Let $\tilde{\mathbf{f}}(\mathbf{x}) := \mathbf{f}(\mathbf{x}) - \mathbf{c}_0 = \sum_{i=1}^{\infty} \mathbf{c}_i \varphi_i(\mathbf{x})$. Then, the covariance of $\mathbf{f}(\mathbf{x})$ can be written as:

$$\mathbb{C}_p[\mathbf{f}(\mathbf{x})] = \mathbb{E}_p [\tilde{\mathbf{f}}(\mathbf{x}) \tilde{\mathbf{f}}(\mathbf{x})^\top] = \mathbb{E}_p \left[\left(\sum_{i=1}^{\infty} \mathbf{c}_i \varphi_i(\mathbf{x}) \right) \left(\sum_{j=1}^{\infty} \mathbf{c}_j \varphi_j(\mathbf{x}) \right)^\top \right].$$

Expanding the inner product gives:

$$\mathbb{C}_p[\mathbf{f}(\mathbf{x})] = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \mathbf{c}_i \mathbf{c}_j^\top \mathbb{E}_p[\varphi_i(\mathbf{x}) \varphi_j(\mathbf{x})].$$

Using orthonormality of the basis $\mathbb{E}_p[\varphi_i \varphi_j] = \delta_{ij}$, we get:

$$\mathbb{C}_p[\mathbf{f}(\mathbf{x})] = \sum_{i=1}^{\infty} \mathbf{c}_i \mathbf{c}_i^\top.$$

This completes the derivation.

Appendix D

Uncertainty Propagation in Stochastic ODEs using Galerkin Projection

To propagate uncertainty in dynamical systems, polynomial chaos expansion (PCE) can be applied to time-dependent systems in which the state or output $y(t, x)$ depends on a random variable x and evolves over time t . The solution $y(t, x)$ can be expressed using a PCE, thereby allowing the uncertainty in the system parameters to propagate through time [23].

Consider the linear first-order ordinary differential equation

$$\frac{dy(t)}{dt} = g(y(t), X, t), \quad y(0) = y_0(X), \quad (\text{D.1})$$

where g governs the system dynamics, X denotes the uncertain model parameters, and the initial condition $y_0(X)$ is also assumed to be uncertain.

Assuming the solution $y(t, x)$ is square integrable, it can be expanded at any time t using a generalized polynomial chaos (gPC) expansion:

$$y(t; x) = \sum_{i=0}^{\infty} c_i(t) \varphi_i(x), \quad (\text{D.2})$$

where $\varphi_i(x)$ are orthonormal basis polynomials, and $c_i(t)$ are the corresponding time-varying coefficients.

Taking the time derivative of the expansion in (D.2) gives:

$$\frac{dy(t)}{dt} = \sum_{i=0}^{\infty} \frac{dc_i(t)}{dt} \varphi_i(x). \quad (\text{D.3})$$

Since $g(y(t), x)$ depends on $y(t)$, substituting the gPC expansion into g and projecting onto $\varphi_i(x)$ leads to:

$$\frac{dc_i(t)}{dt} = \left\langle \varphi_i(x), g \left(\sum_{j=0}^{\infty} c_j(t) \varphi_j(x), x, t \right) \right\rangle, \quad \forall i = 0, 1, 2, \dots \quad (\text{D.4})$$

In practice, the expansion is truncated to a finite order p .

The initial condition can be written as

$$y(0, x) = \sum_{i=0}^{\infty} c_i(0) \varphi_i(x) = y_0(x), \quad (\text{D.5})$$

and projecting both sides onto $\varphi_i(x)$ yields:

$$c_i(0) = \langle \varphi_i(x), y_0(x) \rangle. \quad (\text{D.6})$$

Example: Consider a dynamical system influenced by uncertain parameters modeled as random variables. Let $X = (X_1, X_2) \in L^2(\mathbb{R}^2, p)$, where each component $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ is an independent Gaussian random variable. The system dynamics are governed by:

$$\dot{y}(t) = \frac{dy(t)}{dt} = -X_1 y(t), \quad \text{with} \quad \dot{y}(t) \equiv g(y(t), X), \quad (\text{D.7})$$

and the initial condition is given by:

$$y(0) = X_2 \equiv y_0(X). \quad (\text{D.8})$$

Example: Consider a dynamical system influenced by uncertain parameters modeled as random variables. Let $X = (X_1, X_2) \in L^2(\mathbb{R}^2, p)$, where each component $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ is an independent Gaussian random variable. The evolution of the system state $y(t)$ is governed by the following ordinary differential equation:

$$\dot{y}(t) = \frac{dy(t)}{dt} = -X_1 y(t), \quad \text{with} \quad \dot{y}(t) \equiv g(y(t), X), \quad (\text{D.9})$$

subject to the uncertain initial condition:

$$y(0) = X_2 \equiv y_0(X). \quad (\text{D.10})$$

Using the gPC expansion in (D.2), the evolution of the gPC coefficients, based on (D.4), becomes:

$$\begin{aligned}\frac{dc_i(t)}{dt} &= \left\langle \varphi_i(x), -x_1 \sum_{j=0}^{\infty} c_j(t) \varphi_j(x) \right\rangle \\ &= - \sum_{j=0}^{\infty} c_j(t) \langle \varphi_i(x), x_1 \varphi_j(x) \rangle.\end{aligned}\tag{D.11}$$

The inner product in the summation is defined as:

$$\begin{aligned}A_{ij} &= \langle \varphi_i(x), x_1 \varphi_j(x) \rangle \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \varphi_i(z_1, z_2) \cdot x_1 \cdot \varphi_j(z_1, z_2) \cdot p(z_1) p(z_2) dz_1 dz_2 \\ &= \frac{1}{4} \int_{-1}^{+1} \int_{-1}^{+1} \varphi_i \left(F^{-1} \left(\frac{x_1+1}{2} \right), F^{-1} \left(\frac{x_2+1}{2} \right) \right) \cdot \left(\mu_1 + \sigma_1 F^{-1} \left(\frac{x_1+1}{2} \right) \right) \\ &\quad \cdot \varphi_j \left(F^{-1} \left(\frac{x_1+1}{2} \right), F^{-1} \left(\frac{x_2+1}{2} \right) \right) dx_1 dx_2.\end{aligned}\tag{D.12}$$

Here, F denotes the cumulative distribution function (CDF) of the standard normal distribution, and F^{-1} is its inverse. The variables z_1 and z_2 are independent standard normal random variables, i.e., $z_i \sim \mathcal{N}(0, 1)$. The Gaussian variables are expressed as $x_1 = \mu_1 + \sigma_1 z_1$ and $x_2 = \mu_2 + \sigma_2 z_2$. To evaluate the integral over the unit square $[-1, 1]^2$, a change of variables is used:

$$z_1 = F^{-1} \left(\frac{x_1+1}{2} \right), \quad z_2 = F^{-1} \left(\frac{x_2+1}{2} \right),$$

mapping uniform variables $x_1, x_2 \in [-1, 1]$ to standard normal space via inverse transform sampling.

The initial coefficients $c_i(0)$ from (D.6) become:

$$\begin{aligned}c_i(0) &= \langle \varphi_i(x), y_0(x) \rangle = \langle \varphi_i(x), x_2 \rangle \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \varphi_i(z_1, z_2) \cdot x_2 \cdot p(z_1) p(z_2) dz_1 dz_2 \\ &= \frac{1}{4} \int_{-1}^{+1} \int_{-1}^{+1} \varphi_i \left(F^{-1} \left(\frac{x_1+1}{2} \right), F^{-1} \left(\frac{x_2+1}{2} \right) \right) \cdot \left(\mu_2 + \sigma_2 F^{-1} \left(\frac{x_2+1}{2} \right) \right) dx_1 dx_2,\end{aligned}\tag{D.13}$$

Finally, the time evolution of the gPC coefficients follows:

$$\frac{dc_i(t)}{dt} = - \sum_{j=0}^{\infty} c_j(t) A_{ij},\tag{D.14}$$

with transition matrix A_{ij} and initial conditions $c_i(0)$ given by (D.12) and (D.13), respectively. If the expansion is truncated to order p , the resulting system becomes:

$$\begin{cases} \dot{c}_0 = -(c_0 A_{00} + c_1 A_{01} + \cdots + c_p A_{0p}) \\ \dot{c}_1 = -(c_0 A_{10} + c_1 A_{11} + \cdots + c_p A_{1p}) \\ \vdots \\ \dot{c}_p = -(c_0 A_{p0} + c_1 A_{p1} + \cdots + c_p A_{pp}), \end{cases} \quad (\text{D.15})$$

which describes the reduced-order dynamical system for the evolution of the polynomial chaos coefficients.

The mean and variance of $y(t, x)$, according to Proposition , are given by:

$$\begin{aligned} \mathbb{E}[y(t, x)] &= c_0(t), \\ \mathbb{V}[y(t, x)] &= \sum_{i=1}^{\infty} c_i^2(t), \end{aligned} \quad (\text{D.16})$$

where the coefficients $c_i(t)$ are obtained from the system defined in (D.15).

D.1 Integration using Quadrature Rule

To calculate the integrals in (D.12) and (D.13), the Gauss–Legendre quadrature rule is employed. Although the variables x_1 and x_2 follow Gaussian distributions, the integrals are transformed into the standard domain $[-1, 1]^2$ via inverse transform sampling.

First, N_q Gauss–Legendre quadrature points and weights are generated over the interval $[-1, 1]$. Each quadrature point $x_i \in [-1, 1]$ is mapped to the corresponding standard normal variable using:

$$x'_i = F^{-1}\left(\frac{x_i + 1}{2}\right),$$

where $F^{-1}(\cdot)$ is the inverse cumulative distribution function of the standard normal distribution. The corresponding weights are scaled as $w'_i = \frac{1}{2}w_i$ to account for the change of domain.

The resulting approximations for the integrals in (D.12) and (D.13) are given by:

$$A_{ij} = \sum_{k=1}^{N_q} \sum_{\ell=1}^{N_q} w'_{1k} w'_{2\ell} \cdot \varphi_i(x'_{1k}, x'_{2\ell}) \cdot (\mu_1 + \sigma_1 x'_{1k}) \cdot \varphi_j(x'_{1k}, x'_{2\ell}). \quad (\text{D.17})$$

$$c_i(0) = \sum_{k=1}^{N_q} \sum_{\ell=1}^{N_q} w'_{1k} w'_{2\ell} \cdot \varphi_i(x'_{1k}, x'_{2\ell}) \cdot (\mu_2 + \sigma_2 x'_{2\ell}), \quad (\text{D.18})$$

Note: Since X_1 and X_2 are independent, the 2-dimensional quadrature rule can be constructed as the tensor product of one-dimensional Gauss–Legendre quadrature rules. This allows direct computation of the 2D weights $w'_{1k} w'_{2\ell}$ using 1D rules.

Figure D.1 presents a comparison between the statistical moments obtained via Monte Carlo (MC) simulations and those computed using the Galerkin-based gPC method for the example given in (D.9)–(D.10). A total of $p = 6$ expansion terms were used, corresponding to second-order multivariate polynomials in the two independent Gaussian inputs X_1 and X_2 . The uncertain parameters are modeled as independent Gaussian random variables, with $X_1 \sim \mathcal{N}(0.05, 0.01^2)$ and $X_2 \sim \mathcal{N}(8, 0.01^2)$. The process noise $\omega(t)$ is modeled as zero-mean white noise with variance 0.01^2 . The mean and variance of the system response $y(t)$ obtained from gPC show excellent agreement with the benchmark MC results, which were computed using $n = 10,000$ samples. This confirms the accuracy and efficiency of the gPC method for capturing the effect of parametric uncertainty in the system dynamics.

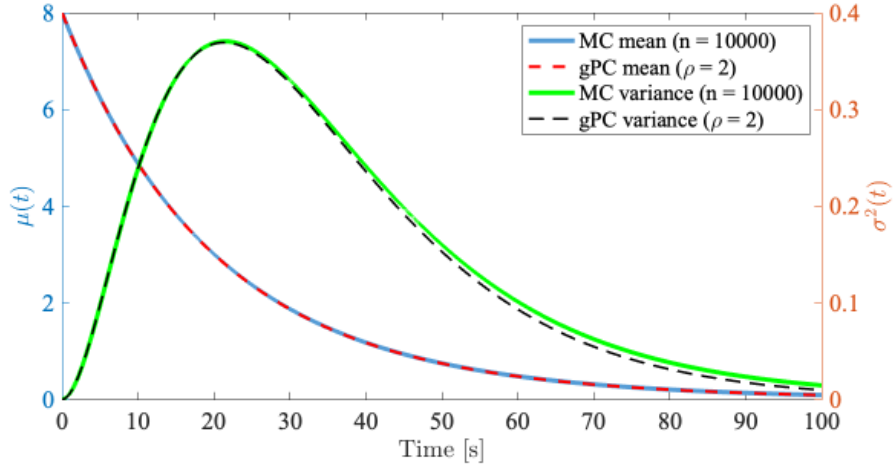


Figure D.1: Comparison between Monte Carlo (MC) simulation and Galerkin-based generalized polynomial chaos (gPC) expansion results. Second-order polynomials are used for both input variables X_1 and X_2 , resulting in $p = 6$ expansion terms.

The time evolution of the gPC expansion coefficients $c_i(t)$, for $i = 0, 1, \dots, 5$, is illustrated in Figure D.2. The first coefficient $c_0(t)$, representing the mean of the response, exhibits

exponential decay due to the system dynamics governed by $\dot{y}(t) = -X_1 y(t)$. The higher-order coefficients reflect the contributions of the stochastic modes and quantify the time-varying uncertainty in the solution. These coefficients are used to reconstruct the full stochastic response and compute its statistical moments via Equation (D.16).

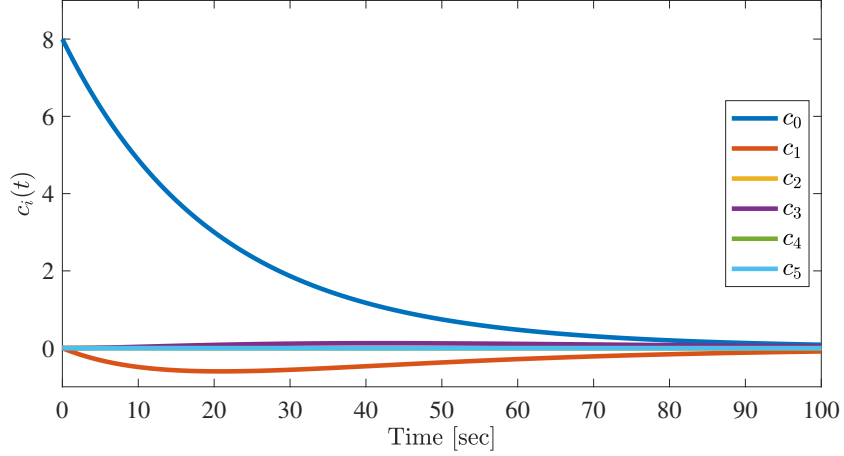


Figure D.2: Time evolution of the (gPC) expansion coefficients.

D.2 Uncertainty Propagation in Dynamical Systems with Parametric and Process Noise

To quantify uncertainty propagation in multi-dimensional dynamical systems, the generalized polynomial chaos (gPC) framework can be extended to incorporate both parametric uncertainty and additive stochastic process noise. Consider the following continuous-time stochastic dynamical model:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{g}(\mathbf{x}(t), \boldsymbol{\Theta}, t) + \mathbf{R}\mathbf{f}(t), \quad \mathbf{x}(0) = \mathbf{x}_0(\boldsymbol{\Theta}) \quad (\text{D.19})$$

where:

- $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector,
- $\boldsymbol{\Theta} \in \mathbb{R}^d$ is a vector of uncertain parameters modeled as random variables with known joint distribution,

- $\mathbf{f}(t) \sim \mathcal{N}(0, \mathbf{Q})$ is a zero-mean white Gaussian process noise vector with covariance $\mathbf{Q} \in \mathbb{R}^{m \times m}$,
- $\mathbf{R} \in \mathbb{R}^{n \times m}$ is a deterministic noise injection matrix.

Assuming the solution $\mathbf{x}(t, \boldsymbol{\Theta}) \in L^2(\Omega; \mathbb{R}^n)$, the state is approximated by a truncated generalized polynomial chaos expansion:

$$\mathbf{x}(t, \boldsymbol{\Theta}) \approx \sum_{i=0}^p \mathbf{c}_i(t) \varphi_i(\boldsymbol{\Theta}) \quad (\text{D.20})$$

where:

- $\mathbf{c}_i(t) \in \mathbb{R}^n$ are the time-dependent gPC coefficients,
- $\varphi_i(\boldsymbol{\Theta})$ are multivariate orthogonal polynomials associated with the distribution of $\boldsymbol{\Theta}$.

Stochastic Galerkin Formulation. Substituting the PCE into the stochastic model (D.19), and applying Galerkin projection yields the following system of differential equations governing the evolution of the gPC coefficients:

$$\frac{d\mathbf{c}_i(t)}{dt} = \left\langle \varphi_i(\boldsymbol{\Theta}), \mathbf{g} \left(\sum_{j=0}^p \mathbf{c}_j(t) \varphi_j(\boldsymbol{\Theta}), \boldsymbol{\Theta}, t \right) \right\rangle + \delta_{i0} \mathbf{R} \mathbf{f}(t), \quad i = 0, \dots, p \quad (\text{D.21})$$

where δ_{i0} is the Kronecker delta, implying that the process noise affects only the first coefficient due to orthogonality (assuming the basis is normalized such that $\varphi_0 = 1$).

Mean and Covariance Propagation. Define $\mathbf{c}(t) = [\mathbf{c}_0^\top(t), \dots, \mathbf{c}_p^\top(t)]^\top \in \mathbb{R}^{n(p+1)}$, the full coefficient vector. Under the assumption of linearized dynamics around a nominal trajectory or in the case of linear systems, the evolution of the mean and covariance of $\mathbf{c}(t)$ can be described by the following moment propagation equations:

$$\begin{cases} \frac{d\widehat{\mathbf{c}}(t)}{dt} = \mathbf{A}_p(t) \widehat{\mathbf{c}}(t) \\ \frac{d\mathbf{P}_c(t)}{dt} = \mathbf{A}_p(t) \mathbf{P}_c(t) \mathbf{A}_p^\top(t) + \mathbf{Q}_p \end{cases} \quad (\text{D.22})$$

where:

- $\hat{\mathbf{c}}(t)$ is the mean of the coefficient vector,
- $\mathbf{P}_c(t) \in \mathbb{R}^{n(p+1) \times n(p+1)}$ is the covariance matrix of the coefficient vector,
- $\mathbf{A}_p(t)$ is the Jacobian or equivalent linearization of the deterministic part of the Galerkin system,
- $\mathbf{Q}_p \in \mathbb{R}^{n(p+1) \times n(p+1)}$ is the noise covariance matrix defined as:

$$\mathbf{Q}_p = \begin{bmatrix} \mathbf{R}\mathbf{Q}\mathbf{R}^\top & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n(p+1) \times n(p+1)}$$

The first two moments of the original state $\mathbf{x}(t, \Theta)$ can then be reconstructed from the statistics of the coefficient vector:

Mean:

$$\mathbb{E}[\mathbf{x}(t)] = \hat{\mathbf{c}}_0(t) \quad (\text{D.23})$$

Covariance:

$$\mathbb{C}[\mathbf{x}(t)] = \sum_{i=1}^p \hat{\mathbf{c}}_i(t) \hat{\mathbf{c}}_i^\top(t) + \sum_{i=0}^p \text{diag}_n(\mathbf{P}_c(t), i) \quad (\text{D.24})$$

where $\text{diag}_n(\mathbf{P}_c, i) \in \mathbb{R}^{n \times n}$ denotes the i -th block diagonal element (i.e., covariance of $\mathbf{c}_i$) of size $n \times n$.

Example: Uncertainty Quantification in a Stochastic ODE with Process Noise

Consider the dynamical system defined in (D.9)–(D.10), with additive process noise $\omega(t)$:

$$\begin{cases} \frac{dy(t)}{dt} = -X_1 y(t) + \omega(t), \\ y(0) = X_2, \end{cases} \quad (\text{D.25})$$

where X_1 and X_2 are independent Gaussian random variables, $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. The process noise $\omega(t)$ is modeled as continuous-time white noise, i.e., $\omega(t) \sim \mathcal{N}(0, Q_c)$, with $Q_c = \frac{Q}{\Delta t}$, where Q is the covariance of the equivalent discrete-time white noise and Δt is the sampling interval.

The objective is to quantify the propagation of uncertainty in system (D.25), including both parametric and process noise contributions.

Assuming the solution $y(t, x)$ is square integrable, it can be expanded using gPC:

$$y(t; x) = \sum_{i=0}^{\infty} c_i(t) \varphi_i(x), \quad (\text{D.26})$$

where $\varphi_i(x)$ are orthonormal polynomial basis functions and $c_i(t)$ are the corresponding time-dependent coefficients.

Substituting the expansion into the stochastic ODE and projecting onto the basis yields the following system governing the evolution of the coefficients:

$$\dot{c}_i(t) = - \sum_{j=0}^p c_j(t) \langle \varphi_i(x), x_1 \varphi_j(x) \rangle + \langle \omega(t), \varphi_i(x) \rangle. \quad (\text{D.27})$$

It follows that the projection of the noise term satisfies:

$$\langle \omega(t), \varphi_i(x) \rangle = \begin{cases} \omega(t), & i = 0 \quad (\varphi_0 = 1), \\ 0, & i = 1, 2, \dots \end{cases}$$

The system of equations for the gPC coefficients can be written in the form:

$$\begin{cases} \dot{c}_0 = -(c_0 A_{00} + c_1 A_{01} + \dots + c_p A_{0p}) + \omega(t), \\ \dot{c}_1 = -(c_0 A_{10} + c_1 A_{11} + \dots + c_p A_{1p}), \\ \vdots \\ \dot{c}_p = -(c_0 A_{p0} + c_1 A_{p1} + \dots + c_p A_{pp}), \end{cases} \quad (\text{D.28})$$

Equivalently, the system can be expressed in compact vector form as:

$$\dot{\mathbf{c}} = -\mathbf{A}\mathbf{c} + \mathbf{w}, \quad (\text{D.29})$$

where $\mathbf{w} = [\omega(t) \ 0 \ 0 \ \dots \ 0]^\top$ captures the effect of process noise on the coefficient dynamics.

The discrete-time form of (D.29) is given by:

$$\mathbf{c}_{k+1} = \mathbf{A}_p \mathbf{c}_k + \mathbf{w}_{p,k}, \quad (\text{D.30})$$

where $\mathbf{A}_p = \mathbf{I}_{(p+1) \times (p+1)} - \mathbf{A}\Delta t$, and $\mathbf{w}_{p,k} \sim \mathcal{N}(0, \mathbf{Q}_p)$ denotes the discrete-time process noise. The corresponding noise covariance matrix is:

$$\mathbf{Q}_p = \mathbb{E}[\mathbf{w}_{p,k} \mathbf{w}_{p,k}^\top] = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

Note that since the system described in (D.30) is linear and driven by Gaussian noise, the gPC coefficient vector $\mathbf{c}_k$ remains Gaussian at all time steps.

The first two statistical moments of the gPC coefficients $\mathbf{c}_k$ can be propagated using the standard state mean and covariance update equations for the linear model:

$$\begin{cases} \hat{\mathbf{c}}_{k+1} = \mathbf{A}_p \hat{\mathbf{c}}_k, \\ \mathbf{P}_{c,k+1} = \mathbf{A}_p \mathbf{P}_{c,k} \mathbf{A}_p^\top + \mathbf{Q}_p, \end{cases} \quad (\text{D.31})$$

where $\hat{\mathbf{c}}$ denotes the mean and $\mathbf{P}_c$ the covariance matrix of $\mathbf{c}$.

The first two moments of the original state y at any time step k can be estimated using the gPC expansion:

$$y_k = \sum_{r=0}^p c_r(k; \omega) \varphi_r(\boldsymbol{\zeta}), \quad (\text{D.32})$$

where $\boldsymbol{\zeta}$ is a random vector used to generate the uncertain parameters X_1 and X_2 , and ω represents the stochastic process noise. It is assumed that ω and $\boldsymbol{\zeta}$ are statistically independent.

The mean of y_k is then obtained as:

$$\begin{aligned} \mathbb{E}[y_k] &= \mathbb{E} \left[\sum_{r=0}^p c_{r,k}(\omega) \varphi_r(\boldsymbol{\zeta}) \right] = \sum_{r=0}^p \mathbb{E} [c_{r,k}(\omega) \varphi_r(\boldsymbol{\zeta})] \\ &= \sum_{r=0}^p \mathbb{E} [c_{r,k}(\omega)] \mathbb{E} [\varphi_r(\boldsymbol{\zeta})] = \sum_{r=0}^p \mathbb{E} [c_{r,k}(\omega)] \int \varphi_r(\boldsymbol{\zeta}) p(\boldsymbol{\zeta}) d\boldsymbol{\zeta} \\ &= \mathbb{E} [c_{0,k}(\omega)] = \hat{c}_{0,k}. \end{aligned} \quad (\text{D.33})$$

The covariance of y_k is derived as:

$$\begin{aligned}
\mathbb{V}[y_k] &= \mathbb{E}[y_k^2] - (\mathbb{E}[y_k])^2 \\
&= \mathbb{E} \left[\left(\sum_{r=0}^p c_{r,k}(\omega) \varphi_r(\boldsymbol{\zeta}) \right)^2 \right] - \hat{c}_{0,k}^2 \\
&= \mathbb{E} \left[\sum_{l=0}^p \sum_{m=0}^p c_{l,k}(\omega) c_{m,k}(\omega) \varphi_l(\boldsymbol{\zeta}) \varphi_m(\boldsymbol{\zeta}) \right] - \hat{c}_{0,k}^2 \\
&= \sum_{l=0}^p \sum_{m=0}^p \mathbb{E} [c_{l,k}(\omega) c_{m,k}(\omega)] \mathbb{E} [\varphi_l(\boldsymbol{\zeta}) \varphi_m(\boldsymbol{\zeta})] - \hat{c}_{0,k}^2 \\
&= \sum_{l=0}^p \mathbb{E} [c_{l,k}^2(\omega)] - \hat{c}_{0,k}^2 \\
&= \sum_{l=0}^p ((\mathbf{P}_{c,k})_{l,l} + \hat{c}_{l,k}^2) - \hat{c}_{0,k}^2 \\
&= \sum_{l=1}^p \hat{c}_{l,k}^2 + \sum_{l=0}^p (\mathbf{P}_{c,k})_{l,l}
\end{aligned} \tag{D.34}$$

where $(\mathbf{P}_{c,k})_{l,l}$ denotes the (l, l) diagonal entry of the covariance matrix $\mathbf{P}_{c,k}$ at time step k , as obtained from the time update equation in (D.31).

Figure D.3 illustrates the uncertainty quantification results for the stochastic ODE described in (D.25), which includes both parametric uncertainty and additive process noise. The figure compares the time evolution of the mean and variance of the system response $y(t)$ obtained using two approaches: Monte Carlo (MC) simulation with $n = 10,000$ samples, and the Galerkin-based gPC expansion with polynomial order $\rho = 2$. The uncertain parameters are modeled as independent Gaussian random variables, with $X_1 \sim \mathcal{N}(0.05, 0.01^2)$ and $X_2 \sim \mathcal{N}(8, 0.01^2)$. The process noise $\omega(t)$ is modeled as zero-mean white noise with variance 0.01^2 . The results demonstrate excellent agreement between the gPC-based predictions and the MC benchmark in both the mean $\mu(t)$ and variance $\sigma^2(t)$ over time. This validates the accuracy and effectiveness of the developed gPC-Galerkin approach in capturing the combined influence of parametric and process uncertainties in the stochastic system.

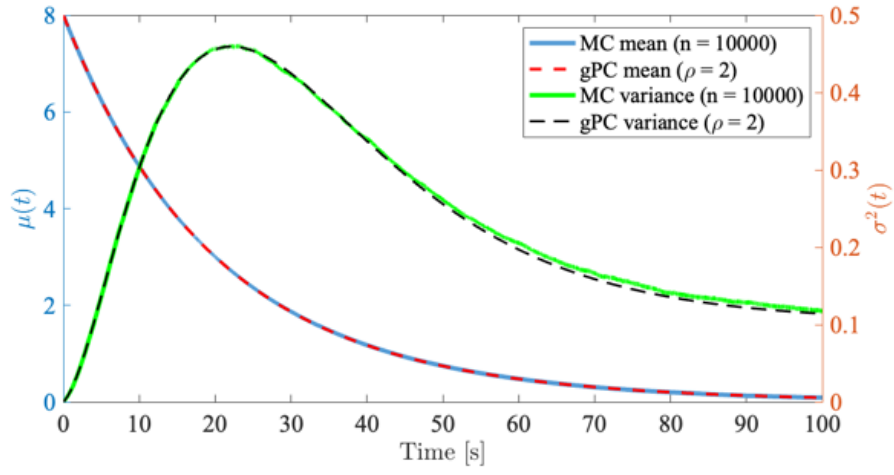


Figure D.3: Comparison between Monte Carlo (MC) simulation and Galerkin-based generalized polynomial chaos (gPC) expansion results for the stochastic ODE with both parametric uncertainty and additive process noise. Second-order polynomials ($\rho = 2$) are used for the two independent input variables X_1 and X_2 . The random variables are distributed as follows: $X_1 \sim \mathcal{N}(0.05, 0.01^2)$, $X_2 \sim \mathcal{N}(8, 0.01^2)$, and the process noise $\omega(t)$ is zero-mean white noise with variance 0.01^2 .

Appendix E

One-Mode Projection

To analyze the structural relationships among nodes of a single type in a bipartite graph, a transformation known as *one-mode projection* is often applied. This process projects the bipartite structure onto one of its node sets, creating a unipartite graph where edges represent indirect connections inferred through shared neighbors in the original bipartite graph. One-mode projection facilitates the use of conventional graph-theoretic tools, enhances interpretability, and enables learning algorithms to operate on homogeneous networks.

Given a bipartite graph $G = (\mathcal{U}, \mathcal{V}, \mathcal{E})$, where $\mathcal{U}$ and $\mathcal{V}$ are disjoint node sets and $\mathcal{E} \subseteq \mathcal{U} \times \mathcal{V}$ is the set of edges, a one-mode projection onto $\mathcal{V}$ produces a unipartite graph that captures pairwise relationships among nodes in $\mathcal{V}$ based on shared connections in $\mathcal{U}$.

Given a bipartite graph $G = (\mathcal{U}, \mathcal{V}, \mathcal{E})$, where $\mathcal{U}$ and $\mathcal{V}$ are disjoint node sets and $\mathcal{E} \subseteq \mathcal{U} \times \mathcal{V}$ is the set of edges, a one-mode projection onto $\mathcal{V}$ produces a unipartite graph that captures pairwise relationships among nodes in $\mathcal{V}$ based on shared connections in $\mathcal{U}$. An illustration of this projection is shown in Figure E.1.

In the weighted case, let w_{vu} denote the weight of an edge between node $v \in \mathcal{V}$ and node $u \in \mathcal{U}$. Two nodes v_i and v_j in $\mathcal{V}$ are connected in the projected graph if they share at least one common neighbor in $\mathcal{U}$. The weight of the edge between v_i and v_j is computed as the average of their connection weights to all shared neighbors:

$$a_{ij} = \sum_{u \in \mathcal{N}(v_i) \cap \mathcal{N}(v_j)} \frac{w_{v_i u} + w_{v_j u}}{2},$$

where $\mathcal{N}(v)$ denotes the set of neighbors of node v in the bipartite graph.

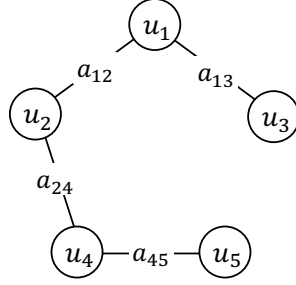


Figure E.1: One-mode projection of the bipartite graph in Figure A.4, illustrating direct relationships between nodes $u \in \mathcal{U}$ based on their shared neighbors $v \in \mathcal{V}$.

The resulting one-mode projection is represented by a weighted adjacency matrix $\mathbf{A}_{\text{proj}} \in \mathbb{R}^{n_v \times n_v}$, where $n_v = |\mathcal{V}|$, and each entry (i, j) corresponds to the computed weight a_{ij} . For example, the adjacency matrix of the fault interaction graph (FIG), derived from the one-mode projection of the fault signature matrix in Table 7.4, is given by:

$$\mathbf{A}_{\text{proj}}(\text{FIG}) = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 2 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 2 & 0 & 2 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 0 & 0 & 2 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 4 & 4 & 4 & 0 & 0 & 0 & 0 & 1 & 1 & 3 \\ 1 & 0 & 1 & 0 & 1 & 1 & 4 & 0 & 4 & 4 & 0 & 0 & 0 & 0 & 1 & 1 & 3 \\ 1 & 0 & 1 & 0 & 1 & 1 & 4 & 4 & 0 & 4 & 0 & 0 & 0 & 0 & 1 & 1 & 3 \\ 1 & 0 & 1 & 0 & 1 & 1 & 4 & 4 & 4 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 4 & 0 & 4 & 4 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 4 & 4 & 0 & 4 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 0 & 0 & 0 & 4 \\ 0 & 3 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 2 & 0 & 1 & 2 & 3 & 3 & 3 & 3 & 0 & 0 & 0 & 0 & 2 & 1 & 0 \\ 0 & 1 & 2 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 4 & 1 & 0 & 1 \end{bmatrix} \quad (\text{E.1})$$

To improve numerical stability in subsequent graph processing, self-loops are added to the matrix by summing it with the identity matrix:

$$\mathbf{A}_{\text{proj}} \leftarrow \mathbf{A}_{\text{proj}} + \mathbf{I}.$$

To account for variations in node degree and to facilitate stable training in graph-based learning models, symmetric normalization is applied:

$$\mathbf{A}_{\text{proj}} \leftarrow \mathbf{D}^{-1/2} \mathbf{A}_{\text{proj}} \mathbf{D}^{-1/2},$$

where $\mathbf{D}$ is the diagonal degree matrix with entries $D_{ii} = \sum_j A_{\text{proj},ij}$, and $\mathbf{D}^{-1/2}$ denotes the elementwise inverse square root of $\mathbf{D}$.

This one-mode projection process preserves structural dependencies encoded in the original bipartite graph and enables efficient representation of indirect relationships among homogeneous nodes for downstream tasks such as clustering, classification, or graph-based inference.

Glossary

- detection** The task of recognizing the occurrence of a fault in a system, typically by analyzing deviations in system behavior or residual signals 74
- estimation graph** An estimation graph is defined as a directed acyclic graph (DAG) whose nodes are local estimators and contains all the possible estimation paths. 4, 6, 8, 25
- estimation path** An estimation path is a directed subgraph of an estimation graph with a single sink node. 8, 25
- fault** A general term used throughout this thesis to denote any root cause of deviation from nominal system behavior, including sensor faults (e.g., noise, bias, drift), actuator faults, unknown external disturbances (e.g., wind, road grade), and model-related uncertainties (e.g., parameter variation, unmodeled dynamics). 5, 51, 204
- identification** The estimation of the magnitude or severity of an identified fault, often used for compensation or reconfiguration purposes 74
- isolation** The process of determining the type and location of a detected fault within the system 74
- path** A *path* (or *stream*) is a subgraph of a directed graph that has only one sink node and contains some or all of directed walks ending in that node. 192
- residual** A signal representing the difference between the actual output of a system and the corresponding model-predicted output. Residuals are used for fault detection and isolation, as they remain near zero under nominal conditions and deviate significantly in the presence of faults. 47, 75