Development of a fuzzy knowledge-based system for local traffic control for incident management

Trinh, Dinh Toan

2008

Trinh, D. T. (2008). Development of a fuzzy knowledge-based system for local traffic control for incident management. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/12083>

<https://doi.org/10.32657/10356/12083>

Nanyang Technological University

Downloaded on 26 Apr 2023 17:26:49 SGT

**DEVELOPMENT OF A FUZZY KNOWLEDGE-
BASED SYSTEM FOR LOCAL TRAFFIC CONTROL
FOR INCIDENT MANAGEMENT**

TRINH DINH TOAN

**SCHOOL OF CIVIL & ENVIRONMENTAL ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY**

2008

DEVELOPMENT OF A FUZZY KNOWLEDGE- BASED SYSTEM FOR LOCAL TRAFFIC CONTROL FOR INCIDENT MANAGEMENT

Trinh Dinh Toan

School of Civil & Environmental Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2008

ACKNOWLEDGEMENTS

The research work presented in this thesis report was carried out at the Nanyang Technological University (NTU), in the School of Civil and Environmental Engineering (CEE). I would highly appreciate the NTU for providing me the scholarship, the CEE for administrative and financial assistances, without which I would not have the opportunity to pursue this research.

I would like to express my deep gratitude to my supervisor, A/P Michael Lam Soi Hoi for his invaluable guidance, supports and encouragements, for his considerable amount of time teaching me on transportation backgrounds, and for his persistence and advices when I lost myself in the research.

I highly appreciate the Transportation Laboratory for offering me opportunities to work at the Lab in favourable conditions. Special appreciation goes to Professor Henry Fan, A/P Piotr Olszewski, A/P Wong Yiik Diew, A/P Lum Kit Meng, and A/P Tan Yan Weng for giving me lectures in transportation and traffic engineering disciplines, for their valuable advices and criticisms on my research report.

I highly appreciate the Transportation Lab's technical staff members: Mr Choi Siew Pheng, Liew Kai Liew, Mrs Ng-Ho Choo Hiang, and Tan-Mai Ju An, for their enthusiastic and non-conditional assistances, without which my research work would have faced many difficulties. Sincere thanks go to my colleagues, especially Dr. Mak and Xie Litian for their very kindness, friendship, and encouragement.

A special gratitude is reserved to my parents, who have been working extremely hard in their lives for their posterities' growth and education, and providing me philosophy of life. Finally, a half of my heart is dedicated to my wife, Nguyen Que Lam, and two of my daughters, Trinh Que Hoa and Trinh Khanh Linh, who continually offered their loves, sacrifices, and encouragements so that I were able to overcome stressful days of a Ph.D. candidate's life, being far away from home, doing my challenging research.

TABLE OF CONTENTS

	PAGE
ACKNOWLEDGEMENTS	ii
TABLE OF CONTENTS	iii
ABSTRACT	vii
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS	xv
 CHAPTER 1 INTRODUCTION.....	 1
1.1 BACKGROUND.....	1
1.1.1 Critical incident management	1
1.1.2 Knowledge-based systems	2
1.1.3 Knowledge Based-Decision Support Systems	3
1.1.4 Concept of uncertainty	5
1.2 PROBLEM STATEMENT	7
1.2.1 Approaches in traffic control	7
1.2.2 Rationale of fuzzy logic for traffic control	9
1.3 RESEARCH OBJECTIVES.....	12
1.4 RESEARCH SIGNIFICANCE	13
1.5 OUTLINE OF THE THESIS REPORT	15
 CHAPTER 2 LITERATURE REVIEW.....	 16
2.1 INCIDENT PROBLEMS	16
2.1.1 Overview	16
2.1.2 Driver behaviour during incidents.....	18
2.2 TRAFFIC CONTROL ON EXPRESSWAYS	19
2.2.1 Overview of Intelligent Transportation Systems	19
2.2.2 Access control	20
2.3 FUZZY LOGIC SYSTEMS	25
2.3.1 Introduction.....	25
2.3.2 Fundamental concepts of FLSs	27
2.3.3 Membership functions	33
2.3.4 Fuzzy logic system design	35
2.3.5 Fuzzy logic control.....	42
2.3.6 Fuzzy logic systems for traffic control	44
2.4 SUPPORT VECTOR MACHINES.....	48
2.4.1 Linear classification problem of SVM.....	48
2.4.2 Non-linear classification problem of SVM.....	51
2.4.3 Support vector regression	52
2.5 SUMMARY	54

CHAPTER 3	STRUCTURE OF THE PROPOSED INCIDENT MANAGEMENT SYSTEM.....	58
3.1	OVERVIEW	58
3.2	CONCEPTUAL MODEL	59
3.2.1	<i>General concept</i>	<i>59</i>
3.2.2	<i>Traffic control procedure in incident events</i>	<i>61</i>
3.3	MULTI-STAGE FLC.....	62
3.3.1	<i>Overview</i>	<i>62</i>
3.3.2	<i>Overall rule base architecture</i>	<i>63</i>
3.3.3	<i>Rule block architectures.....</i>	<i>67</i>
3.4	PROPOSED MS-FLC	68
3.4.1	<i>Conceptual model of the MS-FLC.....</i>	<i>68</i>
3.4.2	<i>Rule base structure of the proposed MS-FLC.....</i>	<i>71</i>
CHAPTER 4	DATA COLLECTION AND ANALYSIS	75
4.1	INTRODUCTION.....	75
4.1.1	<i>The expressway network in Singapore.....</i>	<i>75</i>
4.1.2	<i>ITS systems in Singapore</i>	<i>79</i>
4.1.3	<i>Issues in the LTA's data</i>	<i>80</i>
4.2	DATABASE DEVELOPMENT	83
4.2.1	<i>Overview</i>	<i>83</i>
4.2.2	<i>Relationship between database objects.....</i>	<i>84</i>
4.3	DATA ANALYSIS	85
4.3.1	<i>Traffic patterns</i>	<i>86</i>
4.3.2	<i>Incident problems.....</i>	<i>90</i>
4.4	LEARNING PARAMETERS OF MEMBERSHIP FUNCTIONS	93
4.4.1	<i>Universe of discourse.....</i>	<i>93</i>
4.4.2	<i>Fuzzy partitioning</i>	<i>96</i>
4.4.3	<i>Fuzzy rule generation.....</i>	<i>107</i>
4.5	SUMMARY	112
CHAPTER 5	QUANTIFICATION OF INCIDENT	113
TRAFFIC CONGESTION.....	113	
5.1	INTRODUCTION.....	113
5.2	EVALUATION OF CONGESTION LEVEL	114
5.2.1	<i>Choice of state and control variables</i>	<i>114</i>
5.2.2	<i>Membership functions</i>	<i>116</i>
5.2.3	<i>Fuzzy relations</i>	<i>118</i>
5.3	CONGESTION MOBILITY	121
5.3.1	<i>Membership functions</i>	<i>121</i>
5.3.2	<i>Fuzzy relations</i>	<i>122</i>
5.4	CONGESTION STATUS	126
5.5	FORMATION OF RULES.....	128
5.5.1	<i>Preliminary rule set.....</i>	<i>128</i>
5.5.2	<i>Rule tuning</i>	<i>131</i>
5.5.3	<i>Evaluation of congestion level using actual data</i>	<i>136</i>
5.6	SUMMARY	141

CHAPTER 6 ANTICIPATION OF INCIDENT TRAFFIC CONDITIONS 142

6.1	SCHEMATIC RULE FLOW	142
6.1.1	Overview	142
6.1.2	Risk factor	143
6.2	SHORT-TERM PREDICTION OF TRAFFIC VOLUME.....	150
6.2.1	Introduction.....	150
6.2.2	Basic notations.....	151
6.2.3	Data and baseline predictors.....	152
6.2.4	Overall predictive performance	156
6.2.5	Temporal behaviour.....	158
6.2.6	Effect of rolling horizon	160
6.2.7	Nearest Neighbour method for SVM training	162
6.2.8	Conclusion.....	170
6.3	SHORT-TERM PREDICTION OF TRAVEL TIME.....	171
6.3.1	Introduction.....	171
6.3.2	The study section.....	172
6.3.3	Day-to-day travel time variation in the studied section.....	174
6.3.4	Baseline predictors.....	177
6.3.5	Overall predictive performance	179
6.3.6	Prediction behaviour in congested conditions.....	183
6.3.7	Nearest Neighbour method for SVM training	187
6.3.8	Conclusion.....	190
6.4	ANTICIPATION OF TREND OF TRAFFIC CONDITIONS	191
6.5	SUMMARY	198

CHAPTER 7 CONTROL ACTIONS 201

7.1	INTRODUCTION.....	201
7.2	DECISION-MAKING LOGIC	203
7.2.1	Overview	203
7.2.2	Intervention level.....	205
7.2.3	Control strategy	207
7.3	FORMATION OF RULES.....	211
7.3.1	Input variables	211
7.3.2	Output variable	215
7.4	SUMMARY	221

CHAPTER 8 MODEL VALIDATION..... 223

AND EVALUATION 223

8.1	OVERVIEW	223
8.2	DEVELOPMENT OF THE TSC.....	224
8.2.1	Conceptual model.....	224
8.2.2	Implementation model.....	225
8.3	MODEL VALIDATION.....	230
8.3.1	Calibration of the CFM.....	230
8.3.2	Validation of the TSC model	235
8.4	MODEL EVALUATION.....	247
8.4.1	General issues	247
8.4.2	Control methods.....	256

8.4.3	<i>Results and analysis</i>	259
8.4.4	<i>Uncertainty analysis</i>	277
8.5	EXTENSION: PROPOSED KB-DSS ARCHITECTURE.....	283
8.5.1	<i>Discussion</i>	283
8.5.2	<i>Components of a KB-DSS for traffic control</i>	285
8.6	SUMMARY AND COMMENTS	290
CHAPTER 9 CONCLUSIONS AND RECOMMENDATIONS.....		293
9.1	SUMMARY AND CONCLUSIONS	293
9.2	LIMITATIONS	301
9.3	RECOMMENDATIONS FOR FUTURE RESEARCH	303
REFERENCES.....		305
APPENDIX A: HDB DATABASE		A-1
APPENDIX B1: RTDB – FTP		B1-1
APPENDIX B2: RTDB –XML.....		B2-1
APPENDIX C: CAR-FOLLOWING MODEL		C-1
APPENDIX D: TRAFFIC CONTROLLER.....		D-1

ABSTRACT

Traffic congestion is a critical and pervasive problem confronting many metropolitan areas worldwide. Congestion can be broadly classified into two types: recurring congestion and non-recurring congestion. Due to the complex, critical and uncertain nature, a traffic control for non-recurring congestion management is characterized by time-critical constraints, by the presence of various types of imprecise data and information, and by the uncertainty in evaluating the state of traffic. Characterised by time-critical constraints, the management of non-recurring congestion on expressways should be remedied by implementing effective control measures to ameliorate traffic conditions on expressways, at the same time to avoid imposing excessive congestion on sub-networks. With the above characteristics, an effective traffic control scheme for incident management often requires techniques that deal efficiently with problems of uncertainty and imprecision.

Fuzzy logic is an advanced technique of approximate reasoning in knowledge-based systems. A key motivation in following the fuzzy logic is that the approach is robust in handling problem of uncertainty and imprecision. Fuzzy logic has been widely applied in control systems to manage complex and ill-structured problems with missing and imprecise information. For this capability, fuzzy logic should be a feasible solution for traffic control in incident management. Nevertheless, while there are a wide variety of applications in traffic control for urban intersections, little is known regarding research effort in applying the fuzzy logic systems for traffic control for incident management.

The goal of this research is the development of a fuzzy knowledge-base system for traffic control in incident management on expressways. The system is designed as an engine for a decision support system to assist traffic operators in making decisions to ameliorate incident congestion in a systematic and structured way. The primary objectives of this research include the development of a multi-stage fuzzy logic controller (MS-FLC) for local traffic control during incidents on expressways, and the development of a simulation model for the evaluation of the MS-FLC.

The essence of the MS-FLC is the fuzzy rule base. Important issues in the development of the rule base involve determining and calibrating membership function's parameters. For these purposes, techniques in fuzzy partitioning and fuzzy rule generation including engineering knowledge, grid clustering, and Fuzzy C-Means clustering have been explored. The concept of fuzzy rule generation using the framework of Adaptive Neural Networks has been investigated.

The decision-making logic of the MS-FLC encompasses three stages: the evaluation of the state of traffic under consideration, the prediction of traffic variables and anticipation of incident conditions, and the recommendation of control strategies as well as control actions. The evaluation process is characterised by a comprehensive appraisal of congestion situation, including congestion level, congestion mobility and congestion status, based on speed and density, being primary measurements of traffic variables. In the evaluation stage, the fuzzy relation is used to establish the degrees of association between elements of fuzzy sets.

In the second stage, the anticipation of traffic conditions, the Support Vector Machine (SVM), an advanced technique in machine learning, has been investigated for short-term prediction of traffic variables. Results from the prediction experiments show that SVM significantly outperforms the base-line predictors in various traffic conditions. In particular, the technique provides pattern recognition capabilities to deal more efficiently with incident conditions at high data resolutions.

The recommendation of control strategies and control actions is the last stage of the MS-FLC. The stage proposes a systematic procedure in decision-making sequence, from general to specific. Rules are designed at both strategic and operational levels, ranging from intervention level and control strategy to control action. The stage receives the current and anticipated traffic conditions from the previous stages to recommend solutions. With these types of information, the MS-FLC seeks to operate on both reactive and proactive control modes.

For the evaluation of the MS-FLC, a traffic simulator and control (TSC) has been developed. The TSC consists of two main components: the car-following model

(CFM), and the traffic controller (TC). The CFM has been calibrated and validated using data from Pan-Island Expressway. In the evaluation of the MS-FLC, the local ramp control ALINEA is employed to compare with the MS-FLC in a generic network under various traffic and incident conditions. Results from the evaluation show that the MS-FLC allows significant improvements of travel conditions on the mainline, especially under critical conditions, and substantial reductions of ramp queues. The improvements with MS-FLC show that if properly designed, the MS-FLC will be a robust tool for traffic control under incident conditions.

LIST OF TABLES

	<u>PAGE</u>
TABLE 4.1: EXPRESSWAYS IN SINGAPORE	76
TABLE 4.2 CORRELATION ANALYSIS OF TIME-SERIES TRAFFIC VOLUME DATA	88
TABLE 4.3: NUMBER OF INCIDENTS FROM NOVEMBER 2002 TO NOVEMBER 2003	90
TABLE 4.4: STATISTICS OF INCIDENT DURATION OF VARIOUS TYPES (MIN.)	91
TABLE 4.5: TIME OF INCIDENT OCCURRENCE.....	92
TABLE 4.6: MEMBERSHIP GRADES IN EACH CLUSTER (C = 3).....	106
TABLE 4.7: FCM CLUSTER CENTRES (C = 5).....	107
TABLE 5.1: RELATIONAL MATRIX OF SPEED AND DENSITY	119
TABLE 5.2: RELATIONAL MATRIX OF DENSITY AND CONGESTION LEVEL	120
TABLE 5.3: RELATIONAL MATRIX OF SPEED AND CONGESTION LEVEL	121
TABLE 5.4: RELATIONAL MATRIX OF LOW SPEED AND HEAVY CONGESTION.....	124
TABLE 5.5: RELATIONAL MATRIX OF MEDIUM SPEED AND HEAVY CONGESTION.....	125
TABLE 5.6: REVISED RELATIONAL MATRIX OF SPEED AND DENSITY	132
TABLE 5.7: RULE DECISION MATRIX OF CONGESTION LEVEL	134
TABLE 5.8: EVALUATION OF CONGESTION LEVELS BY THREE METHODS	138
TABLE 6.1: DECISION TABLE FOR THE RISK FACTOR	149
TABLE 6.2: MAPEs (%) BY SVM FOR DIFFERENT TEST DAYS	157
TABLE 6.3: PREDICTION RESULTS FROM DIFFERENT TEST SETS.....	181
TABLE 6.4: COMPARISON BETWEEN THE HISTORICAL MEAN AND MEDIAN.....	181
TABLE 6.5: TVC PREDICTION USING NN, MEAN, AND MEDIAN VALUES	189
TABLE 6.6: RULE TEMPLATES FOR THE 1 ST SUB-STAGE	194
TABLE 6.7: RULE TEMPLATES FOR THE 2 ND SUB-STAGE.....	195
TABLE 7.1: DECISION TABLE FOR RULES WITH LOCAL RAMP CONTROL	217
TABLE 8.1: THE MOST INFLUENCING PARAMETERS IN THE CFM.....	232
TABLE 8.2: PCE EQUIVALENT FACTORS FOR SINGAPORE EXPRESSWAYS.....	237
TABLE 8.3: LATERAL DISTRIBUTIONS OF TRAFFIC ON PIE.....	238
TABLE 8.4: TRAFFIC COMPOSITION ON PIE IN AM PEAKS	239
TABLE 8.5: ACTUAL AND SIMULATED SPEEDS ON UPSTREAM SEGMENT (FREE-FLOW).....	241
TABLE 8.6: SUMMARY OF PAIRED T-TEST FOR SPEED ANALYSIS.....	242
TABLE 8.7: ERRORS OF SIMULATED FLOW RATES (MAPE, %)	246
TABLE 8.8: <i>MOEs</i> FOR LOW-MEDIUM DEMAND.....	260
TABLE 8.9: <i>MOEs</i> FOR MEDIUM-HIGH DEMAND.....	262
TABLE 8.10: <i>MOEs</i> FOR HIGH DEMAND - CASE 1	265
TABLE 8.11: <i>MOEs</i> FOR HIGH DEMAND - CASE 2	269
TABLE 8.12: <i>MOEs</i> WITH HIGH DEMAND - CASE 3	272
TABLE 8.13: <i>MOEs</i> FOR HIGH DEMAND - CASE 4	275

LIST OF FIGURES

	<u>PAGE</u>
FIGURE 1.1: A KB-DSS ARCHITECTURE.....	4
FIGURE 1.2: CRISP SETS VERSUS FUZZY SETS IN EVALUATION OF SPEED	9
FIGURE 2.1: BLOCK DIAGRAMS OF TRAFFIC RAMP CONTROL.....	21
FIGURE 2.2: THE FLOW-OCCUPANCY FUNDAMENTAL DIAGRAM	22
FIGURE 2.3: DEMAND-CAPACITY STRATEGY (OPEN LOOP)	23
FIGURE 2.4: A CLOSE-LOOP RAMP METERING CONTROL (ALINEA)	23
FIGURE 2.5: SCHEMATIC DIAGRAM OF A FUZZY LOGIC SYSTEM.....	26
FIGURE 2.6: CHARACTERISTICS OF FUZZY SETS.....	28
FIGURE 2.7: INTERSECTION AND UNION OPERATIONS OF FUZZY SETS A AND B	29
FIGURE 2.8: COMPOSITION OF MULTIPLE FUZZY RELATIONS.....	33
FIGURE 2.9: THREE PARTS OF A MEMBERSHIP FUNCTION	33
FIGURE 2.10: PRIMARY TYPES OF MEMBERSHIP FUNCTIONS	35
FIGURE 2.11: STRUCTURE OF A FUZZY VARIABLE DESIGN	36
FIGURE 2.12: THE FUZZY PARTITION IN THE UNIVERSE OF DISCOURSE	37
FIGURE 2.13: FUZZY IMPLICATION, AGGREGATION AND DEFUZZIFICATION	41
FIGURE 2.14: CONCEPTUAL DIAGRAM OF A FLC	42
FIGURE 2.15: OPTIMAL HYPERPLANE IN A TWO-DIMENSIONAL PROBLEM.....	49
FIGURE 3.1: COMPONENTS OF THE FUZZY KBS	59
FIGURE 3.2: TRAFFIC CONTROL PROCEDURE DURING INCIDENTS	61
FIGURE 3.3: HIERARCHICAL ARCHITECTURE OF A MS-FLC	64
FIGURE 3.4: FORMATION OF RULES.	65
FIGURE 3.5: AN EXAMPLE OF RULE BLOCK.....	67
FIGURE 3.6: SINGLE-STAGE PARALLEL STRUCTURE	67
FIGURE 3.7: MULTI-STAGE CASCADE STRUCTURE	68
FIGURE 3.8: MULTI-STAGE COMPOSITE STRUCTURE	68
FIGURE 3.9: CONCEPTUAL MODEL OF FLC FOR INCIDENT-RELATED TRAFFIC CONTROL	69
FIGURE 4.1: THE EXPRESSWAY NETWORK IN SINGAPORE	75
FIGURE 4.2: DELAY ON EXPRESSWAYS (JAN.-JUN., 2003)	77
FIGURE 4.3: DENSITY CONTOUR MAP OF A DOUBLED CONGESTION.....	78
FIGURE 4.4: A STUDIED STRETCH OF SEGMENTS ON PIE	78
FIGURE 4.5: INTERACTIVE MAP FOR EXPRESSWAY SYSTEM.....	79
FIGURE 4.6: SNAPSHOTS OF CAMERA VIEWS ON PIE	80
FIGURE 4.7: RELATIONSHIPS AMONG TABLES IN HDB	85
FIGURE 4.8: VOLUME PATTERNS ON TWO CONSECUTIVE SEGMENTS ON PIE	87
FIGURE 4.9: SPEED-FLOW RELATIONSHIP.....	89
FIGURE 4.10: DISTRIBUTION OF CONGESTION INDICES ON SEGMENT 80007766.....	90
FIGURE 4.11: DISTRIBUTION OF INCIDENTS ON EXPRESSWAYS.....	92
FIGURE 4.12: DISTRIBUTIONS OF INCIDENT DURATIONS	94
FIGURE 4.13: SURVIVAL FUNCTIONS OF INCIDENT DURATIONS	95
FIGURE 4.14: PROBABILISTIC PARTITIONING OF THE ACCIDENT DURATION	97
FIGURE 4.15: MEMBERSHIP FUNCTIONS FOR ACCIDENT DURATION.....	98
FIGURE 4.16: PARTITIONING OF SPEED (LOW RESOLUTION, THREE FUZZY SETS).....	99

FIGURE 4.17: MEMBERSHIP FUNCTIONS OF SPEED (THREE FUZZY SETS)	101
FIGURE 4.18: PARTITIONING OF SPEED (HIGHER RESOLUTION, FIVE FUZZY SETS).....	101
FIGURE 4.19: MEMBERSHIP FUNCTIONS OF SPEED (FIVE FUZZY SETS).....	102
FIGURE 4.20: EQUALIZED PARTITIONING OF DATA SPACE	103
FIGURE 4.21: INPUT DATA FOR FCM	105
FIGURE 4.22: FCM CLUSTERING FOR SPEED-DENSITY SPACES ($C = 3$)	105
FIGURE 4.23: FCM CLUSTERING ($C = 5$)	106
FIGURE 4.24: OBJECTIVE FUNCTIONS IN FCM.....	107
FIGURE 4.25: SPEED-VOLUME TRAINING DATA.....	109
FIGURE 4.26: ANFIS MEMBERSHIP FUNCTIONS OF SURGENO TYPE.....	110
FIGURE 4.27: CHECKING DATA VERSUS TESTING DATA IN VALIDATION.....	111
FIGURE 5.1: RULE BASE CONFIGURATION FOR THE FIRST STAGE.....	113
FIGURE 5.2: MEMBERSHIP FUNCTIONS OF THE SPEED DOMAIN ($n_{x1} = 3$).....	117
FIGURE 5.3: FUZZY PARTITIONING THE SPEED DOMAIN ($n_{x1} = 5$)	117
FIGURE 5.4: MEMBERSHIP FUNCTIONS OF THE OUTPUT SPACE	118
FIGURE 5.5: MEMBERSHIP FUNCTIONS OF CONGESTION MOBILITY	122
FIGURE 5.6: COMPOSITIONAL INFERENCE OF SLOW MOVING - HEAVY CONGESTION	123
FIGURE 5.7: COMPOSITIONAL INFERENCE OF MEDIUM MOVING - HEAVY CONGESTION	124
FIGURE 5.8: SURFACE KNOWLEDGE OF $CL(v \times k)$	126
FIGURE 5.9: LINGUISTIC VALUES OF QUEUE LENGTH.....	127
FIGURE 5.10: FUZZY SETS OF CONGESTION STATUS	127
FIGURE 5.11: FUZZY SETS OF SPEED ($n_{x1} = 5$)	131
FIGURE 5.12: FUZZY PARTITION OF DENSITY ($n_{x2} = 5$).....	131
FIGURE 5.13: FUZZY PARTITION OF CONGESTION LEVEL	132
FIGURE 5.14: RULE SET FOR CONGESTION LEVEL	133
FIGURE 5.15: RULE VIEWER FOR CONGESTION LEVEL (MAMDANI TYPE).....	135
FIGURE 5.16: SPEED AND DENSITY PROFILES.....	136
FIGURE 5.17: THE RULE VIEWER FOR $CL(v)$	138
FIGURE 5.18: THE RULE VIEWER FOR $CL(k)$	139
FIGURE 5.19: EVALUATION OF CONGESTION LEVELS BY DIFFERENT METHODS	140
FIGURE 5.20: STATE-CONTROL FUZZY RELATIONS USING A SINGLE VARIABLE	140
FIGURE 6.1: SCHEMATIC RULE FLOW FOR THE 2 ND STAGE	143
FIGURE 6.2: A CONFORMITY LEVEL TEMPLATE	145
FIGURE 6.3: MEMBERSHIP FUNCTIONS FOR CAPACITY REDUCTION.....	145
FIGURE 6.4: DETECTION OF DECISION POINTS FOR TOD MEMBERSHIP FUNCTIONS	146
FIGURE 6.5: MEMBERSHIP FUNCTIONS FOR TOD	147
FIGURE 6.6: MEMBERSHIP FUNCTIONS FOR THE RISK FACTOR.....	147
FIGURE 6.7: THE STUDY SEGMENT ON PIE	152
FIGURE 6.8: ERRORS OF THE OVERALL PREDICTION BY DIFFERENT METHODS	156
FIGURE 6.9: SVM-BASED PREDICTION IN COMPARISON WITH ACTUAL DATA.....	157
FIGURE 6.10: MAPES OF PREDICTORS ON AM PEAK.....	159
FIGURE 6.11: PREDICTED TRAFFIC PROFILES IN AM PEAK	159
FIGURE 6.12: EFFECT OF ROLLING HORIZON ON PREDICTION ACCURACY	160

FIGURE 6.13: MAPES OF NN-SVM WITH 60 → 15 PREDICTION	165
FIGURE 6.14: NN-SVM WITH SIMULATED INCIDENT DATA	167
FIGURE 6.15: PREDICTED PROFILES BY DIFFERENT TRAINING SETS.....	168
FIGURE 6.16: THE STUDY AREA.....	173
FIGURE 6.17: DAY-TO-DAY VARIATION OF TRAVEL TIMES IN THE STUDIED SEGMENT	174
FIGURE 6.18: DAY-TO-DAY TRAVEL TIME DISTRIBUTION PATTERN	175
FIGURE 6.19: SKEWNESS AND SPREAD OF THE TRAVEL TIME DISTRIBUTION	176
FIGURE 6.20: OVERALL PERFORMANCES OF PREDICTORS.....	180
FIGURE 6.21: PROFILES OF PREDICTED TRAVEL TIMES FROM DIFFERENT PREDICTORS	182
FIGURE 6.22: MAPES IN ASSOCIATION WITH THE OBSERVED TRAVEL TIME.....	183
FIGURE 6.23: PROPORTION OF AVERAGE PERCENTAGES OF ERRORS.....	183
FIGURE 6.24: PREDICTED PROFILES FOR RECURRING CONGESTION	184
FIGURE 6.25: PREDICTION UNDER INCIDENT CONDITION	185
FIGURE 6.26: SVM PREDICTION IN THE SIMULATED INCIDENT	186
FIGURE 6.27: EFFECTS OF DISTANCE THRESHOLD ON TRAINING.....	188
FIGURE 6.28: PROFILES OF THE DATA IN TEST SETS IN RELATION TO THE MEAN AND MEDIAN....	190
FIGURE 6.29: MEMBERSHIP FUNCTIONS FOR (ADJUSTED) V/C^* RATIO.....	192
FIGURE 6.30: FIS ARCHITECTURE FOR THE 1 ST SUB-STAGE	193
FIGURE 6.31: FIS ARCHITECTURE FOR THE 2 ND SUB-STAGE.....	194
FIGURE 6.32: KNOWLEDGE SURFACES FOR TWO SUB-STAGES.....	198
FIGURE 7.1: SCHEMATIC FLOW OF THE 3 RD STAGE	204
FIGURE 7.2: FUZZY SETS OF INTERVENTION LEVEL	206
FIGURE 7.3: LAYOUT OF AN EXPRESSWAY SECTION.....	207
FIGURE 7.4: OD ASSIGNMENT AND MEMBERSHIP FUNCTIONS	208
FIGURE 7.5: DECISION TREE FOR CONTROL STRATEGY	210
FIGURE 7.6: LAYOUT OF LOCAL RAMP CONTROL.....	212
FIGURE 7.7: MEMBERSHIP FUNCTIONS FOR RAMP QUEUE	214
FIGURE 7.8: TENTATIVE MEMBERSHIP FUNCTIONS FOR THE REMAINING TIME.....	215
FIGURE 7.9: MEMBERSHIP FUNCTIONS FOR RAMP FLOW.....	216
FIGURE 7.10: FIS FOR RAMP CONTROL USING MIN IMPLICATION METHOD.....	221
FIGURE 8.1: CONCEPTUAL MODEL OF THE TSC.....	224
FIGURE 8.2: ARCHITECTURE OF THE TSC IMPLEMENTATION MODEL.....	225
FIGURE 8.3: CAR-FOLLOWING DIAGRAM.....	225
FIGURE 8.4: THE BLOCK DIAGRAM FOR ACCELERATION/DECELERATION	227
FIGURE 8.5: CAR-FOLLOWING MODEL.....	229
FIGURE 8.6: THE FLC IN SIMULINK.....	230
FIGURE 8.7: A ZOOM IN THE SIMULATION PROFILES OF VEHICLES' POSITIONS AND SPEEDS.....	233
FIGURE 8.8: VEHICLE'S SPACING, TIME GAP, AND ACCELERATION (N-VEHICLE SYSTEM)	234
FIGURE 8.9: ACTUAL SITE FOR MODEL VALIDATION.....	235
FIGURE 8.10: SIMULATED VERSUS OBSERVED COUNT UNDER FREE-FLOW CONDITION	243
FIGURE 8.11: SIMULATED VERSUS OBSERVED COUNT UNDER MEDIUM CONGESTION.....	244
FIGURE 8.12: SIMULATED VERSUS OBSERVED COUNT UNDER HEAVY CONGESTION	245
FIGURE 8.13: ABSOLUTE PERCENTAGE OF ERRORS ON UPSTREAM SEGMENT.....	247
FIGURE 8.14: LAYOUT OF THE SIMULATED NETWORK.....	248
FIGURE 8.15: ARRIVAL-DEPARTURE CURVE.....	251

FIGURE 8.16: CALCULATION OF QUEUES ON EXPRESSWAY.....	251
FIGURE 8.17: CALCULATION OF RAMP QUEUE	253
FIGURE 8.18: DETERMINATION OF OPTIMAL SET POINT OCCUPANCY	257
FIGURE 8.19: RAMP CONTROL UNDER <i>ALINEA</i>	261
FIGURE 8.20: <i>MOEs</i> FOR MEDIUM-HIGH DEMAND, <i>ALINEA</i> CONTROL	263
FIGURE 8.21: TRAFFIC PARAMETERS FOR MEDIUM-HIGH DEMAND, <i>ALINEA</i> CONTROL	264
FIGURE 8.22: PROFILES OF <i>MOEs</i> UNDER <i>FLC</i> CONTROL - HIGH DEMAND, CASE 1	267
FIGURE 8.23: SIMULATED PROFILES BY <i>ALINEA/Q</i> - HIGH DEMAND, CASE 2.....	270
FIGURE 8.24: PROFILES OF RAMP CONTROL UNDER <i>FLC</i> - HIGH DEMAND, CASE 3	273
FIGURE 8.25: PROFILES OF <i>MOEs</i> UNDER <i>No CONTROL</i> - HIGH DEMAND, CASE 3.....	274
FIGURE 8.26: PROFILES OF SIMULATION RESULTS - HIGH DEMAND, CASE 4.....	276
FIGURE 8.27: Δ_{MS} VERSUS THE LENGTH OF UPSTREAM SECTION.....	280
FIGURE 8.28: Δ_{MS} VERSUS THE RAMP STORAGE CAPACITY	281
FIGURE 8.29: Δ_{MS} VERSUS THE SIMULATION PERIOD	283
FIGURE 8.30: CORRIDOR MS- <i>FLC</i> ARCHITECTURE	284
FIGURE 8.31: PROPOSED ARCHITECTURE OF THE KB-DSS FOR TRAFFIC CONTROL	285
FIGURE 8.32: THE ROLLING HORIZON APPROACH IN SVM PREDICTION	288
FIGURE 8.33: A TEMPLATE OF GUI FOR TRAFFIC CONTROL	290

LIST OF ABBREVIATIONS

AI	: Artificial Intelligence
AFC	: Adaptive Fuzzy Control
ANFIS	: Adaptive-Network-Based Fuzzy Inference System
APE	: Average Percentage of Errors
ATMS	: Advanced Traffic Management System
ATIS	: Advanced Traveller Information System
CFM	: Car-Following Model
CI	: Congestion Index
CL	: Congestion Level
(C_Mob)	: Congestion Mobility
COA	: Centre-Of-Area
C_Stat	: Congestion Status
CTP	: Current-Time based Predictor
CW-FLC	: Corridor-Wide Fuzzy Logic Controller
DBMS	: Database Management System
DSS	: Decision Support Systems
EMAS	: Expressway Monitoring and Advisory System
ESP	: Double Exponential Smoothing Predictor
FCM	: Fuzzy C-Means
FLC	: Fuzzy Logic Controller
FLS	: Fuzzy Logic System
FOM	: First-Of-Maximum
GUI	: Graphical User Interface
HDB	: Historical Database
HMP	: Historical Mean Predictor
ITS	: Intelligent Transportation Systems
KB-DSS	: Knowledge Based-Decision Support System
KBSs	: Knowledge-Based Systems
k-NN	: k-Nearest Neighbour
LOSs	: Levels Of Service

LP	: Linear-Programming
LQ-HC	: Long Queue - Heavy Congestion
LTA	: Land Transport Authority
MAPE	: Mean Absolute Percentage of Error
MD	: Mean Density
MISO	: Multiple-Inputs-Single-Output
MIMO	: Multiple-Inputs-Multiple-Outputs
MM-HC	: Medium Moving - Heavy Congestion
MOEs	: Measures Of Effectiveness
MOM	: Mean-Of-Maximums
MQ-HC	: Medium Queue - Heavy Congestion
MS	: Average Speed
MS-FLC	: Multi-Stage Fuzzy Logic Controller
NN	: Nearest Neighbour
NN-FLC	: Neural-Network-Based Fuzzy Logic Controllers
PCU	: Passenger Car Unit
RBF	: Radial Basic Function
RMSE	: Root Mean Square Error
RTDB	: Real-Time Database
SISO	: Single-Input-Single-Output
SM-HC	: Slow Moving - Heavy Congestion
SQ-HC	: Short Queue - Heavy Congestion
SQL	: Structural Query Language
SVM	: Support Vector Machine
SVR	: Support Vector Regression
TC	: Traffic Controller
TMC	: Traffic Management Centre
TOD	: Time Of Day
TSC	: Traffic Simulator and Control
TTD	: Total Travel Distance
TTS	: Total Time Spent
TTT	: Total Travel Time

TVC	: Time-Varying Coefficient
TWT	: Total Waiting Time
XML	: Extensible Markup Language

CHAPTER 1 INTRODUCTION

1.1 BACKGROUND

1.1.1 Critical incident management

Traffic congestion is a pervasive problem confronting many metropolitan areas in the world. Congestion creates excessive delay and deteriorates the functionality of transport networks and urban areas. Congestion can be broadly categorized into two types: recurring congestion and non-recurring congestion. Recurring congestion is a predictable problem that occurs when traffic demand regularly exceeds road capacity during a particular time of day at certain locations creating bottlenecks. Owing to the deterministic nature, recurring congestion can primarily be solved by long-term transportation planning such as building new roads to strike a balance between traffic demand and network capacity, or by suitable traffic management plans. Non-recurring congestion, on the other hand, is a problem caused by unpredictable events called incidents. Incidents can be categorised as lane-blockage incidents that result in a temporary drop of the road capacity, such as accident, vehicle breakdown, or abnormal rises in traffic demand. Due to the complex and critical nature, the management of non-recurring congestion is characterized by time-critical constraints, by the presence of imprecise data and information, by the uncertainty in evaluating traffic states and in setting rules that govern the relationships between the states and control measures. For these reasons, the problem of non-recurring congestion should primarily be solved by implementing efficient control measures to ameliorate traffic conditions on expressways while avoiding spreading congestion to urban streets. To be efficient, traffic control for incident management often requires robust techniques that deal efficiently with the problem of uncertainty, in association with human judgement skills. To be efficient, traffic control for incident management should be supported by flexible instruments such as knowledge-based systems that respond efficiently to unexpected situations.

1.1.2 Knowledge-based systems

Knowledge-based systems (KBSs) are systems that employ knowledge embodied in computer programs to provide problem solutions. A KBS usually consists of a knowledge base, an inference engine, and a user interface. The *knowledge base* is a collection of knowledge relating to a specific problem or specific domain. Knowledge in the knowledge base is encoded at a certain level of abstraction through the transformation of the inference engine. The *inference engine* is an instrument that performs reasoning functions or logic transformation. It employs forward-chaining or backward-chaining techniques to infer conclusions. Finally, the *user interface* is a platform that allows bi-directional communication between the system and the user. The interface supports the man-machine cooperation by providing facilities so that the system's resources are utilised properly and effectively.

There are several points that distinguish KBSs with conventional computer programs. First, conventional computer programs use mathematical algorithms and systematic solution procedures with pre-defined assumptions. The programs require strict formats and orders of input streams to produce “precise” solutions. It is very difficult to identify and remedy mistakes in conventional computer programs since the knowledge and expertise embodied in the programs are not explicitly stated in actual codes. Second, since the pre-defined assumptions upon which the programs are built may not hold under various future conditions, conventional computer programs are inflexible under unexpected situations. Last, since conventional computer programs are not robust in capturing human judgements and reasoning in case information is deficient, they are susceptible to uncertain environments and imprecise concepts. By contrast, in KBSs, knowledge representation is transparent and declarative since it is written in English-like syntaxes. In most KBSs, the order in which knowledge is represented and organised is not critical. In particular, KBSs are designed in expert-like manner, namely they may have errors but possess potential to learn from errors. More importantly, KBSs can be built in an incremental process, which allows for an accumulation of knowledge without

changing the system's structure. In this manner, the system can enrich and improve knowledge and expertise through its life cycle.

Knowledge in knowledge bases can be represented in a number of ways, such as rule-based reasoning, case-based reasoning, semantic networks, and frames (Turban, 1992). Among those, rule-based reasoning (presented in Section 1.2.2) is one of the most popular knowledge representation formalisms.

1.1.3 Knowledge Based-Decision Support Systems

Decision Support Systems (DSS) are interactive, flexible, and adaptable computerized information systems that combine models and data to support decision-making activities with extensive user involvement (Turban, 1992). Constrained by cognitive and time limits, the decision makers require a DSS to facilitate problem solving more smoothly and rapidly, and to improve the effectiveness of decision making. A DSS does not make decisions itself but provide computerised assistance to automate several tasks of the decision-making process. A DSS typically accomplishes data acquisition and analysis to derive recommended solutions, but gives full control to the operators to determine the final decision with their own insights and experiences. A DSS consists of a data base, a model base, and a human-machine interface (Figure 1.1).

Bidgoli (1989) classified human decisions into structured, unstructured, and semi-structured decisions. Structured decisions concern with a routine and repetitive process in an established context. They are programmable and characterized by well-defined standard operating procedures. Such decisions do not really need decision support. Unstructured decisions, on the other hand, are mostly non-repetitive and have no standard operating procedure. They are required in emergent contexts where decision making may be highly intuitive. The DSS in this case is an intelligent system that provides a great assistance to help the operators confronted with qualitative type of decisions. Finally, semi-structured decisions are not well-established as structured decisions but include structured modules for data collection, data analysis, and information processing, etc. These modules help the

operator avoid being overloaded by a huge amount of data and information to focus on critical issues (Toan and Lam, 2005).

Figure 1.1: A KB-DSS architecture

In semi-structure decisions, a DSS may also employ conventional optimisation programs to optimise several tasks. However, as stated earlier, conventional optimisation programs with pre-defined assumptions and boundary conditions are inflexible under unexpected situations. For real-time complex and uncertain problems, their recommended solutions are often deficient. Consequently, the employment of a KBS for decision making under such circumstances is often desirable. A DSS that uses a knowledge base as the inference engine is called a knowledge based-decision support system (KB-DSS).

The rationale behind the employment of the knowledge-based approach for decision support lies in the capability of KBSs to enhance the decision-making quality greatly. On the one hand, KBSs provide utilities for linguistic reasoning and explanation to the user. On the other hand, KBSs allow learning task-specific knowledge and expertise from domain experts to improve the problem-solving capabilities. Apart from that, KBSs contain large amounts of knowledge to speed up and ensure consistent decisions.

A DSS in the field of traffic monitoring and control is characterised by time-critical semi-structured decisions in complex and dynamic circumstances. The system must provide rapid access to databases and utilize various types of data, information and knowledge to interpret what is going on and to anticipate what may happen before proposing control actions. With advantages of KBSs in assisting decision making, KB-DSS is a suitable approach for the highly dynamic, complex, and ill-structured problem of traffic control in incident situations.

1.1.4 Concept of uncertainty

Human knowledge is inexact in nature. Very often, people are not completely sure about the truth of a statement. Their daily verbal communication is characterised by uncertainty, ambiguity and imprecision (Teodorović and Vukadinović, 1998). The uncertainty in their statements is reflected by a frequent adherence to linguistic hedges such as *about, almost, nearly, slightly, more-or-less, the same, and very*, etc. In Artificial Intelligence (AI), the term *uncertainty* is known as *approximate reasoning*. It refers to a wide range of problems and situations where relevant information is deficient. It represents how poorly people approximate the current problems or anticipate future events. In some cases uncertainty implies the partial belief of reasoning, the lack of knowledge about certain problems in a study or lack of knowledge in forecasting future events that reduces confidence in drawing conclusions. In other cases, uncertainty may refer to potential differences between recorded values and actual values, in other words - to the limit on precision of measurements.

One source of uncertainty (and imprecision) stems from *data* and *information*. Data may be inaccurately measured, and information may be incompletely collected. The imprecision in data measurement includes random errors that indicate statistical fluctuations due to limitations of measurement devices, and systematic errors - reproducible inaccuracies in the same direction due to latent factors that persist throughout the measurement. In addition, data measurement may incur sampling errors as the whole population is inferred from a measured sample (Montgomery and Runger, 2002), or processing errors in data aggregation or data smoothing.

Another source of uncertainty comes from *knowledge representation*. Languages for problem representation are usually imprecise, approximate, and vague in nature. Upon the same event, there may be different ways of treatment and representation. At a higher level, knowledge for *decision-making* may be inadequate, unjustifiable or conflicting.

A good example of uncertainty in traffic engineering is the variability in the perception and classification of traffic state. Data such as flow rate, speed, and density needed for evaluation of traffic state may be missing or incompatible. Incident attributes such as incident location, start time, and severity may be unreliable or approximate. Consequently, the traffic state may be imprecisely evaluated. Even in the presence of “accurate” data and information, the way to classify the traffic state is inherently arguable. For example, the term *heavy* traffic receives no universal perception. One may decide that a traffic stream on expressways with speed of v (km/h) is either *heavy* or *moderate*. Another may suggest that the speed characterises both *heavy* and *moderate* categories, with degrees of belief μ_H and μ_M , respectively. At the other extreme, there may be argument that it is density rather than speed the more important factor representing the traffic congestion status. In brief, the way to define traffic states and boundaries between states are vague and subjective in nature.

There are several methods for representing uncertainty. The *numeric* method uses scales with two extreme values to represent the degree of belief. For example, the numerical scale $[0, 1]$ covers the whole domain of belief, where 0 represents complete uncertainty and 1 represents the opposite. The *graphical* method represents uncertainty by a spectrum of colours that gradually changes from one extreme to the other. The *qualitative* approach offers a number of ways to “quantify” uncertain problems such as linguistic phrases and ranking, or the composite method that uses linguistic representation combined with number, like fuzzy logic.

Approaches in treatment of uncertainty can be broadly categorised into *parallel certainty inference techniques* and *uncertainty structuring techniques* (Turban,

1992). The former determines the level of uncertainty in a parallel process. Methods in this category include probability theory (Lindley, 1987; Dubois and Prade, 2000), certainty factors (Dubois and Prade, 2000; Ozbay and Kachroo, 1999), and Evidence Theory (Dubois and Prade, 2000). The latter formalizes uncertainty problems in structured goal-oriented procedures. Typical methods in this category include neural networks, generic algorithms, and fuzzy logic.

Although uncertainty is a pervasive and serious problem, its treatment is still limited. People are normally comfortable in dealing with concrete questions such as *black* or *white*, *true* or *false*, *good* or *bad*. They are reluctant in dealing with problems of uncertainty. Consequently, many knowledge-based systems avoid this issue. In traffic engineering, uncertainty exists exogenously in the data and in the way knowledge is represented, not in the person who manages traffic. Although none of the treatments is perfect, it is necessary to improve methods to solve the problem rather than to avoid the problem.

1.2 PROBLEM STATEMENT

1.2.1 Approaches in traffic control

The rapid development of Intelligent Transport System's technologies induces development of real-time traffic control, but at the same time imposes more specific requirements on attempts to seek efficient systems for real-time applications. Various approaches have been developed for real-time traffic management, including analytical optimisation and automatic control approaches (Logi and Ritchie, 2001). The analytical optimisation approach (Moreno-Banos et al., 1993; Papageoriou, 1994; Ran and Boyce, 1996) models the state of traffic systems based on specific assumptions about the system behaviour and dynamics, and projects the current network conditions into the future state (Ozbay and Kachroo, 1999). Given assumptions made in the problem formulation, solutions of the analytical method for traffic engineering problems may be questionable (Ben-Akiva et al., 2003). In order to offer systematic solution procedures for optimal routing strategies,

mathematical models are normally very complex, computationally expensive, thus they hardly satisfy real-time requirements. As opposed to the analytical optimisation approach, the automatic control approach (Papageorgiou et al., 1991; Diakaki et al., 1997; Kachroo et al., 1998; Kachroo and Özbay, 1999) claims for the ability to model the behaviour of measurable processes and classify large patterns of input data. However, the method does not provide an explanation instrument for the operators to determine appropriate control actions (Logi and Ritchie, 2001). To remedy this challenging problem, a comprehensive system that utilizes available data sources, knowledge and control techniques is needed to improve the quality of decision making for a multidisciplinary domain as traffic control.

In realization of such challenges, extensive research efforts have been made to employ knowledge representation and inference systems that are able to handle real-time traffic data to find solutions to traffic control problems in a human-like fashion. Such systems, known as knowledge-based systems, should be suitable for ill-structured problems in well-defined domains where algorithmic and automatic procedures are not appropriate.

A number of researchers have investigated knowledge-based systems for traffic management and control: Ritchie (1990) proposed an artificial intelligence-based solution approach to assist operators in freeway and arterial traffic management systems. Gupta et al. (1992) introduced a prototype knowledge-based expert system to assist non-recurrent congestion management on the Massachusetts Turnpike. Cuenca et al. (1995) used frame as a knowledge representation tool in complicated traffic problems. Sadek (1998) employed two AI paradigms - the Case-Based Reasoning and stochastic search algorithm to develop a routing decision support system that is designed to learn from the past in an attempt to handle the problem of missing data. Logi and Ritchie (2001) presented a real-time knowledge-based system for decision support in the selection of integrated traffic control plans after the occurrence of non-recurring congestion on freeway and arterial networks. The aforementioned literatures employed different types of knowledge representation and reasoning techniques and achieved different degrees of success.

1.2.2 Rationale of fuzzy logic for traffic control

Rule-based reasoning system is a branch of KBSs in which knowledge is represented in the form of condition-action pairs: IF *conditions (premises, antecedents)* satisfy, THEN *actions (conclusions, consequences)* occur. Rules can be developed and modified independently, but when combined in the inference engine they work together in specific rule sets to produce a common conclusion. Rule-based reasoning is applicable in domains that rely on heuristics to manage complex and ill-structured problems. The technique inherits a number of properties of a KBS including communicable, comprehensible, and upgradeable.

In the family of rule-based systems, there exist two types of rules: regular rules that evaluate state and control variables using crisp sets, and fuzzy rules that use fuzzy sets. It would be important to identify essential distinctions between the two types of rules in dealing with the uncertainty problem.

Considering the following example: classification of speed values using crisp sets in comparison with fuzzy sets (Figure 1.2).

Figure 1.2: Crisp sets versus fuzzy sets in evaluation of speed

Suppose that traffic speed is used to determine the term *congestion level* that is subsequently used to recommend control actions. Let's assume that the speed universe of discourse (V) is decomposed into three sets: *Low*, *Medium*, and *High*. In regular rules, each set is uniquely assigned a binary value of 1 for all inputs

associating with its domain, and 0 otherwise (Figure 1.2a). To evaluate the state of traffic, the following fact-state rules are used:

- *if $V \leq 40$ km/h then low_speed*
- *if 40 km/h $< V \leq 70$ km/h then medium_speed*
- *if 70 km/h $< V$ then high_speed.*

Then the state-state rules are used to determine the congestion level:

- *if low_speed then heavy_congestion*
- *if medium_speed then medium_congestion*
- *if high_speed then free_flow.*

Finally, the state-action rules are defined:

- *if heavy_congestion then control action A*
- *if medium_congestion then control action B*
- *if free_flow then no control action.*

It appears that using regular rules, the state of traffic is uniquely assigned with the same “weight” over a relatively large domain of inputs. For instance, speed of 40 km/h is classified as *Low*, the same as speed 0, and the same control action (A) is recommended, whereas speed of a little above 40 km/h (such as 40.1 km/h) is classified as *Medium* and a different control action (B) is triggered. This is not justified. It should be noted that traffic data are often not as accurately collected as expectation, thus there should be no significant difference between speed of 40 km/h and 40.1 km/h in representing traffic status. The data contamination eventually invalidates the attempt to classify “precisely” the state of traffic. Consequently, owing to the inability in dealing with imprecision in data measurements though approximate reasoning, regular rules expose a serious drawback for control applications.

The fuzzy logic approach remedies this problem using fuzzy sets associated with different degrees of membership: there is no clear boundary between fuzzy sets, and a specific numerical input may be assigned to more than one fuzzy set with different strengths known as degrees of fulfilment. For example, in Figure 1.2b, speed of 40 km/h is classified into both *heavy* and *medium* congestion with degrees of membership μ_L and μ_M , respectively. Different membership degrees are then aggregated using defuzzification method to obtain a single value for control.

Another advantage of fuzzy-logic over the regular rule-based approach lies in the way data is quantified. In general, data can be evaluated using granulation or quantification (Zadeh, 1994). In data quantification, the intervals are numerical values whereas in granulation the intervals are overlapping fuzzy sets. Numerical values are specific while fuzzy values, such as *heavy congestion*, are somewhat abstract. Nevertheless, *heavy* represents a choice from several sets, rather than, for example 100 in the numerical domain. Therefore, the use of linguistic instead of numerical values is a way of data compression. The advantages of granulation over quantification include: (i) it is more general, hence is easy to be perceived; (ii) it reflects the way in which people quantify data by likelihood of categories rather than numerical data; (iii) the transition from one fuzzy set to another is gradual rather than abrupt, representing continuity in human perception.

Traffic control is a multistage and multivariable problem. The control decision process is performed from a low level to high level of abstraction, namely from data to information to knowledge. Therefore, the uncertain and imprecise nature of traffic control is an accumulation of those features from data measurement, information processing, human perception, and decision-making. Traffic control under incident situations is even more uncertain and critical since it involves using various types of traffic and incident data to arrive at control decisions, given critical-time constraints.

Fuzzy logic is a qualitative approach that is close to human observation, reasoning and decision-making. Fuzzy logic has been widely applied in control systems, especially where data are collected by sensors (Turban, 1992). The key motivation

for the use of fuzzy logic rests on its capability in handling of uncertainty and the ability to solve complex non-linear problems within a reasonable amount of time (Hoogendoorn et al., 1998). For these reasons fuzzy logic should be a suitable solution for traffic control in incident management.

1.3 RESEARCH OBJECTIVES

The ultimate goal of this research is the development of a fuzzy KBS for local traffic control in incident situations on expressways. The fuzzy KBS acts as an engine for the KB-DSS in real-time control to ameliorate traffic conditions under incident occurrences. The fuzzy KBS system is designed so that the KB-DSS assists traffic operators in a systematic and structured manner.

In line with this goal, the following objectives are identified:

- (i) To establish a network data connection from the database server and to develop a historical database (HDB) and a real-time database (RTDB).
- (ii) To explore the HDB for knowledge on traffic and incident problems in the expressway network for subsequent steps in this study.
- (iii) To learn from the HDB parameters of membership functions of traffic and incident variables.
- (iv) To develop a multi-stage fuzzy logic controller (MS-FLC) for local ramp traffic control on expressways under incident occurrences.
- (v) To develop a simulation model for validation and evaluation of the MS-FLC.

Specific tasks associated with the MS-FLC include:

- To design the MS-FLC architecture.
- To establish a systematic procedure for evaluation of traffic conditions in the proximity of the ramp using fuzzy logic.

- To investigate an advanced technology in machine learning - the support vector machine (SVM) as an important instrument for forecasting of traffic variables, adapted to incident conditions. From the predicted traffic variables, the anticipation of traffic trend will be performed using fuzzy logic.
- To propose appropriate local traffic control strategies and actions associated with the evaluated and anticipated traffic conditions.

Scope of the research

The research focuses on the local ramp traffic control for management of traffic congestion during incidents on expressways. The fuzzy KBS developed in this thesis encompasses the data base, the fuzzy rule base and the fuzzy logic controller (see Figure 3.1). Some proposals regarding the architecture for a corridor-wide control and the connection between the KBS and the GUI (in the KB-DSS) will be made in Section 8.5.

Given the time available for this study, the complexity of the fuzzy KBS, and the need for exploring a considerable number of methods and tools for the fuzzy KBS, this study focuses on the development of an experimental system for local ramp traffic control for managing incidents of local scale. Most of the research findings are obtained from the evaluation of control performances in the exploratory work that uses simplified networks, and have not been tested on a larger network with several on/off ramps.

1.4 RESEARCH SIGNIFICANCE

The primary contributions of this research may include:

- (i) Development of a systematic fuzzy logic system for KB-DSS for local ramp traffic control on expressways during incidents. The decision-making sequence of the MS-FLC consists of three stages: the evaluation of the exiting state of traffic by a thorough appraisal of congestion situation, including congestion level, congestion mobility and congestion status; the

anticipation of the change in the traffic state using fuzzy rules based on the results from traffic prediction models adjusted by a risk factor; and the recommendation of control actions using fuzzy rules that evolve from general to specific, from strategic level to operational level.

- (ii) Development of a Traffic Simulator and Control (TSC) for the knowledge-base system validation and evaluation. The TSC includes the car-following model (CFM) that simulates the car-following behaviour, and the traffic controller (TC) that integrates with the embedded fuzzy rule blocks and implements control decisions.

The secondary contributions of this research may include:

- (i) Exploration and application of the various techniques for learning membership functions of traffic and incident variables. This learning has not been addressed in literature so far for developing a KB-DSS for non-recurring congestion management.
- (ii) Investigation and application of the SVM technique for traffic volume and travel time prediction for proactive control, with extension to incident conditions. Significant findings are obtained in studying the effect of data resolution, the training size, and the rolling horizon on the prediction accuracy.
- (iii) Investigation of the use of the Nearest Neighbour method for the improvement of SVM training speed. Findings are also obtained in investigating the effect of using Nearest Neighbour value instead of the historical mean and historical median on the accuracies of traffic prediction models.
- (iv) Application of various methods to handle the problem of uncertainty in management of non-recurring congestion. The use of fuzzy logic throughout the research, the choice of variables to characterise traffic state (Section 5.2.1), the use of the risk factor (Section 6.1.2), and the analysis of

uncertainty (Section 8.4.4) represent efforts to deal with the uncertainty problem in multistage-multivariable traffic control under incident situations.

1.5 OUTLINE OF THE THESIS REPORT

Chapter 2 reviews relevant works that have been done in the field of incident management and traffic control. In the Chapter, fundamental concepts and applications of fuzzy logic systems in traffic control are summarised, and theoretical background of Support Vector Machines is introduced.

Chapter 3 presents the proposed architecture of the incident management system - the MS-FLC, where the rule base structure of the controller is explained.

Chapter 4 describes the data collection and analysis, and illustrates the methodology of parametric learning for membership functions from data. In the Chapter, a number of methods used for fuzzy partitioning are employed.

Chapter 5 elaborates the first stage of the MS-FLC in quantification of incident traffic congestion. The concept of congestion level is discussed, the relation between fuzzy variables described and the formation of rules demonstrated.

Chapter 6 presents the second stage of the MS-FLC, including the prediction of traffic variables and the anticipation of traffic condition's evolution. For prediction of traffic variables, the SVM technique is investigated and employed.

Chapter 7 describes the decision-making logic for control measures and actions, being the last stage of the MS-FLC.

Chapter 8 presents the development and validation of the Traffic Simulator and Control model, the evaluation and uncertainty analysis of the MS-FLC.

Finally, Chapter 9 summarises the thesis report and highlights important findings from the research. The limitations of the MS-FLC are discussed and potential directions for further research are recommended.

CHAPTER 2 LITERATURE REVIEW

2.1 INCIDENT PROBLEMS

2.1.1 Overview

The terminology *incident* used in this research includes unpredictable events (accident, vehicle breakdown, etc.) that make temporary reduction in road capacity. From the traffic operator perspective, road maintenances can be considered as planned incidents since they also create temporary reduction in capacity and traffic management plans are required in advance. However, to drivers who face the situation the first time, such activities may be as unexpected as normal incidents.

In US, it was reported that an estimated 60% of vehicle loss can be attributed to incidents. Approximately 30% of these cases are unreported incidents, which were normally minor incidents with little impacts. Incidents associated with vehicle disablements accounted for 80% of the reported incidents, while only 10% of which were caused by accidents (Ozbay and Kachroo, 1999).

Incidents create temporal bottlenecks on highways and slow down traffic streams. As a lane is blocked, a queue may form and propagate upstream until the incident is cleared and normal traffic condition is restored. Under a severe incident, there may be a massive jam of traffic, and the congestion is aggravated if the demand is high. Once a long queue is built up, it may take a long time for the accumulated traffic to dissipate. Unless the normal traffic condition can be quickly restored, a more devastating secondary incident would likely be expected, and a substantial amount of time would be required to solve the problem. Apart from causing delay, increasing accident risk, and reducing the reliability of transport networks, long-lasting incidents create severe environmental consequences. These consequences can be reduced significantly by early incident detection and efficient incident management.

From an institutional perspective, incident management is a coordination of activities carried out by responsible agencies to bring traffic to normal conditions. An incident management programme involves four sequential stages: incident detection, response, clearance, and recovery (Ozbay and Kachroo, 1999). From the perspective of traffic control, incident management on expressways involves the implementation of real-time traffic monitoring and control measures to ameliorate traffic conditions in expressways and nearby urban streets, given incident occurrences.

Attributes that characterise an incident include *incident type*, *duration*, *location*, and *severity*. Incident type (accident, vehicle breakdown, obstacles, etc.) refers to the cause of incident. Incident duration is the elapse time from when the incident starts to the time traffic is restored. Ozbay and Kachroo (1999) divided incident duration into several components, including incident detection and verification time, incident response time, and time to normal flow.

Location of an incident can be specified by the link on which the incident takes place. For control purpose, the position on the expressway link at which the incident starts is important, since a specific position is associated with a specific process of queue development, and different counter-measures are required accordingly. For example, an incident at the upstream portion of the link may induce queue propagation across the upstream on-ramp, and hence strong ramp-control intervention may be needed. On the other hand, a minor incident occurring at the downstream portion of the link for a relatively short time may not require ramp control or diversion.

The term *incident severity* can be characterised by several indicators, including incident duration, the total number of vehicles involved, the total or average delay of vehicles, the length of queue, or the lane closure. With an exception of the lane closure, the above indicators are essentially governed by the traffic demand during the incident duration, or more precisely, the ratio of traffic demand to the remaining road capacity at the incident place. The issues in selecting this ratio as an indicator of congestion magnitude will be discussed in Section 5.2.1. In Section 7.3.1 (ii) the ratio will be employed as a primary control input.

2.1.2 Driver behaviour during incidents

Understanding driver's behaviour, e.g. route switching, during incident congestion is important for an efficient traffic management. Responses to traffic congestion can be classified into two types: *preventive* response and *reactive* response (Stern, 1999). The former is associated specifically with recurrent congestion or with the presence of *pre-trip* information (Section 2.2.1). Pre-trip travel decision reflects a static process of choice behaviour, during which drivers react with pre-trip information by changing their departure time, travel route, or travel mode. Reactive response, on the other hand, is the reaction of drivers to *en-route* information, such as incident congestion information. The en-route choice may be the primary reaction in case of incident since drivers have to choose either to proceed on the mainline or divert to an alternative route. Such decisions reflect a dynamic process during which drivers continually compare the utility of the current path to that of the alternatives. Drivers would switch to the current best route if the indifference band for route switching, defined as the maximum tolerable range of savings in travel times and costs, is exceeded (Jou et al., 2005). The threshold magnitude is time and location-specific, and differs considerably among drivers. Under critical situations, drivers may reduce thresholds, increase risk taking, or even make decisions irrationally.

Observations show that at incident sites, especially accidents, drivers tend to slow down to see what is happening nearby. This phenomenon is called the rubbernecking behaviour. Cragg and Demetsky (1995) investigated the effect of this phenomenon on capacity reduction and found that the additional capacity loss due to rubbernecking could be as much as 10-25%. The more congested the route and the more severe the accident, the higher the rubbernecking factor would be expected.

2.2 TRAFFIC CONTROL ON EXPRESSWAYS

2.2.1 Overview of Intelligent Transportation Systems

Intelligent Transportation Systems (ITS) are advanced and interdisciplinary technologies exploited to improve traffic mobility and safety. The main components of ITS for traffic control and monitoring include Advanced Traffic Management System (ATMS) and Advanced Traveller Information System (ATIS). The ATMS exploits advanced technologies to improve the operation of traffic systems, particularly in managing recurring and non-recurring congestion. The system uses strategies such as ramp metering and traffic signal for freeway operation and incident management. The ATIS, on the other hand, aims at assisting drivers in improving the efficiency and convenience of travel by disseminating useful pre-trip or en-route information (Jou et al., 2005). Pre-trip information provides travellers with information on traffic conditions and travel times in the network for them to select route, departure time, and travel mode prior to their trip. En-route information provides travellers with network conditions while travelling.

Information from the ATIS can be categorized into *descriptive* or *normative* types. The descriptive information disseminates the state of the network so that the travellers choose routes by themselves. Such type of information can be obtained from Highway Advisory Radios (HAR), VMS and In-Vehicle Navigational Systems (IVNS). The normative information conveys control directives such as diverting or lane changing, through in-vehicle devices or VMS.

Jou et al. (2005) broadly classified the information provided by the ATIS into *qualitative* and *quantitative* types. Since the traveler's perception towards numeric or descriptive information may be different, the effect towards route switching due to quantitative information may be more significant than qualitative information. The reason is that the quantitative type of information may help travelers understand better the network traffic conditions. The qualitative information typically has lower effects on route switching since such type of information cannot depict as clearly the traffic conditions to travelers as the quantitative information.

Types of traffic control on expressways

Available types of control typically employed on expressways include route control, access control, and integrated control (Ben-Akiva et al., 2003). Route control (guidance) is implemented by the ATIS. Access control is implemented by ATMS, typically ramp control that regulates flows at on ramps to monitor the mainline traffic, and sometimes at off-ramps to alleviate congestion on local streets. Integrated control is the combination of route control and ramp control. Ramp control is the most popular type of access control, and is the focus of this research. Since off-ramp control is seldom used, the term *access control* in this thesis refers to on-ramp control.

2.2.2 Access control

Ramp metering is the most direct and efficient way of access control on expressways (Papageorgiou, 1999). Ramp meters are traffic signals placed on entrances of expressways to regulate traffic flows. At the ramp, traffic is discharged at reasonable rates to maintain a demand-capacity balance so as to alleviate congestion downstream of the ramp. The primary objectives of ramp control on expressways include:

- Maintaining smooth movement of expressway traffic.
- Maximizing expressway throughput.
- Increasing speed of expressway traffic.
- Minimizing incident-induced congestion.

Ramp metering may increase delay and queue length at the on-ramp, but traffic conditions on expressways can significantly be improved, and the network-wide objectives are enhanced. Under heavy demands on expressway facilities, ramp metering can also improve the situation by encouraging diversion to the sub-networks as part of an integrated traffic control. However, the diversion is beneficial only when the alternatives have sufficient capacities to accommodate the diverted traffic. Otherwise, the congestion will be transferred from expressways to the urban streets, impairing the overall network performances.

Ramp control has been widely used for several decades recently. Attempts have been made toward the development of efficient ramp control strategies. These strategies can be broadly categorised into local control and area-wide control.

(i) Local ramp control

A local ramp controller regulates traffic based on the local measurements in the vicinity of the ramp. This type of control is suitable when local congestion is observed. In case congestion is pervasive in different sections of the expressway corridor, area-wide control should be considered. The controller may employ *fixed-time* or *traffic-responsive* control approaches.

Fixed-time ramp metering approach (Wattleworth, 1967; Wang, 1972; Wang and May, 1973 (in Papageorgiou, 1999)) develops offline ramp metering plans for a particular time of day based on historical traffic profiles. The advantages of this approach include the simple control algorithm and data availability to develop a system-wide control plan. The drawback of this approach is its lack of responsiveness to current traffic conditions, hence leading to potential overloading or under-utilization of the expressway.

The *traffic-responsive* ramp metering approach calculates ramp-metering rates using real-time data to maintain desirable conditions. The approach may adopt either *open-loop* or *close-loop* control system, illustrated in Figure 2.1 (Ozbay and Kachroo, 1999).

Figure 2.1: Block diagrams of ramp traffic control

In an *open-loop* control (Figure 2.1a), the control input $r(t)$ is independent of the system input $x(t)$. *Demand-capacity* is a typical open-loop controller, which determines the ramp flow at an interval k satisfying:

$$r(k) = \begin{cases} \max(q_{cap} - q_{in}(k-1), r_{min}) & \text{if } o_{out}(k) \leq o_{cr} \\ r_{min} & \text{otherwise} \end{cases} \quad (2.1)$$

where q_{cap} (see Figures 2.2 and 2.3 also) denotes the capacity of the expressway downstream of the ramp, q_{in} denotes the demand upstream of the ramp, o_{out} denotes the occupancy downstream of the ramp, o_{cr} denotes critical occupancy that corresponds to the expressway capacity, and r_{min} denotes the pre-specified minimum value of the ramp rate.

The strategy of the *open-loop* control (Figure 2.3) is to keep the traffic state downstream in stable operation while maximizing the ramp flow to reach the downstream capacity. If the occupancy $o_{out}(k)$ measured downstream exceeds the critical occupancy o_{cr} , congestion exists and the ramp rate should be reduced to minimum rate r_{min} to prevent traffic breakdown.

In a closed-loop system (Figure 2.1b) the system input $x(t)$ is a function of the control input $r(t)$ and the error $e(t)$. The controller is therefore more sensitive to its environment: the traffic sensors provide the controller with real-time data to compute ramp flows. The implemented control changes the traffic condition, the changed traffic information is fed back to the controller, and the cycle repeats. For this operation mechanism the closed-loop control system is called a feedback control (Papageorgiou et al., 1991).

Figure 2.2: The flow-occupancy fundamental diagram

Figure 2.3: Demand-capacity strategy (open loop)

The most widely used technique in the close-loop control is ALINEA (Figure 2.4). ALINEA determines the metering rates such that the traffic state on the expressway approaches a pre-defined condition.

$$r(k) = r(k-1) + K_R [\hat{o} - o_{out}(k)] \quad (2.2)$$

where K_R is a regulator parameter; $\hat{o}$ is the target value for the downstream occupancy.

While the demand-capacity strategy responds to excessive occupancy o_{out} only when the threshold value o_{cr} is exceeded, ALINEA reacts smoothly to the difference $\hat{o} - o_{out}(k)$, thus it prevents congestion downstream at a higher throughput level.

Figure 2.4: A close-loop ramp metering control (ALINEA)

Papageorgiou et al. (1997), in evaluating the ALINEA algorithm in several field tests, have identified advantageous features of ALINEA over other local ramp controllers, including simplicity, transferability, cost efficiency, and flexibility.

The benefits of the ALINEA ramp-metering algorithm on traffic conditions are even higher in the case of non-recurrent congestion. In the evaluation of the case study with a substantial number of incidents on the Boulevard Périphérique and Boulevard des Maréchaux in Paris, Papageorgiou et al. (1997) found that with respect to the total time spent, the ALINEA algorithm allowed saving of 11.6% and 10% for Boulevard Périphérique and Boulevard des Maréchaux, respectively, but led to an increase of 7.4% on the radial streets, compared to the no-control scenario.

Ben-Akiva et al. (2003) conducted an evaluation of the freeway ramp control in the Central Artery/Tunnel network in Boston using ALINEA algorithm with MITSIM microscopic traffic simulator. In their experiment, four levels of target occupancy (15%, 19%, 21% and 23%), and three levels of traffic demand (80%, 90% and 100%) were used for calibration. The results showed that low demands (80% and 90%) did not warrant metering. In contrast, significant travel time savings for higher level of demand (100%, 110% and 120%) were obtained.

(ii) Area-wide ramp control

This type of control considers the coordination of several controllers in an expressway corridor. The approach pursues the same goals as the local ramp-metering counterpart: the ramp flows are discharged so as to maintain desirable conditions. While local ramp metering is performed independently for each ramp based on local measurements, the area-wide ramp control makes use of all available information to calculate simultaneously the ramp flows for all controllable ramps within the corridor. This may provide potential system-wide improvements over local ramp metering since more comprehensive information is utilised and more robust control action is coordinated. Previous studies of coordinated ramp control were presented in Stephanedes and Chang (1993), and Zhang et al. (1996).

METALINE is a coordinated feedback control algorithm that is used for coordinated ramp control along expressway corridors. The control algorithm of METALINE (Papageorgiou et al., 1997) is described by Equation (2.3).

$$\mathbf{r}(k) = \mathbf{r}(k-1) - \mathbf{K}_1[\mathbf{o}(k) - \mathbf{o}(k-1)] - \mathbf{K}_2[\mathbf{O}(k) - \hat{\mathbf{O}}(k)] \quad (2.3)$$

where

$\mathbf{r} = [r_1 \dots r_m]^T$, being the vector of m controllable ramp volumes;

$\mathbf{o} = [o_1 \dots o_n]^T$, being the vector of n measured occupancies along the expressway;

$\mathbf{O} = [O_1 \dots O_m]^T$, being the vector of m measured occupancies, normally those immediately downstream of the controlled ramps;

$\hat{\mathbf{O}} = [\hat{O}_1 \dots \hat{O}_m]$, being the vector of m corresponding set values;

$\mathbf{K}_1 \in R^{m \times n}$, $\mathbf{K}_2 \in R^{m \times m}$ are the two gain matrices that return a vector of metering rates.

The use of sophisticated algorithms, however, does not necessarily lead to improvements of performance in every case. The ALINEA was tested against METALINE using its macroscopic traffic simulator for the Boulevard Peripherique in Paris (Papageorgiou et al., 1991). The results revealed that both feedback control strategies lead to roughly the same results under normal conditions, and the METALINE was only a little superior to ALINEA in case of unexpected incidents due to more comprehensive information.

2.3 FUZZY LOGIC SYSTEMS

2.3.1 Introduction

A fuzzy logic system (FLS) is a non-linear mapping of input to the output universe of discourse using fuzzy logic principles. There are endless possibilities for the mapping, and each of which requires a deep understanding of fuzzy logic theory and elements comprising the FLS (Mendel, 1995). FLSs provide foundations for

incorporating both subjective and objective knowledge, for handling both numerical data and linguistic information.

Figure 2.5 shows the general diagram of a FLS, which consists of 4 components: fuzzifier, fuzzy rule base, inference engine and defuzzifier. The fuzzifier matches numerical inputs with the conditions of the rules and converts them into fuzzy sets.

Figure 2.5: Schematic diagram of a fuzzy logic system

There is a degree of membership for each linguistic term applied to each input variable. The fuzzy rule base holds a collection of IF–THEN rules. For each rule the antecedent describes to what degree the rule satisfies, while the conclusion assigns a membership function to each of the output variables. The fuzzy inference engine maps input fuzzy sets to output fuzzy sets and combines outputs from constituent activated rules to obtain an aggregated fuzzy output. It handles the way in which rules are combined and executed. Finally, the defuzzifier converts the aggregated fuzzy output to non-fuzzy (crisp) values using defuzzification techniques such as the centroid and maxima, which are to be introduced later in this chapter.

FLSs have been widely applied to many control industrial processes (Mamdani, 1974; King and Mamdani, 1975; Ostergaad, 1976; Larsen, 1980; Sakai, 1985 - in Lee, 1990). In engineering, a great variety of FLS applications have been found, notably for control problems, such as aircraft control (Larkin, 1985 - in Lee, 1990; Teodorović et al, 1994), train operation control (Yasunobu et al., 1983; Yasunobu and Miyamoto, 1985 - in Lee, 1990), and traffic control (Section 2.3.6). Interested readers may refer to Lee (1990), Ross (1995), and Mendel (1995) for comprehensive reviews of the FLS's applications in engineering.

2.3.2 Fundamental concepts of FLSs

(i) Fuzzy sets

In fuzzy logic, the term *universe of discourse* indicates the collection of all elements constituting an input or output variable. A fuzzy set A defined on the universe of discourse X is a set of ordered pairs:

$$A = \{(x, \mu_A(x)) | x \in X\} \quad (2.4)$$

where x is the input value and $\mu_A(x)$ is the membership grade whose value is in the interval $[0, 1]$, denoted as $\mu_A : X \rightarrow [0, 1]$. As X is continuous, A can be written as:

$$A = \int_X \mu_A(x) / x \quad (2.5)$$

Equation (2.5) denotes the collection of all points $x \in X$ with corresponding membership function $\mu_A(x)$.

The membership function provides a measure for degree of inclusion of an element in X to the fuzzy set. The greater $\mu_A(x)$ the more certain is the statement that element x belongs to set A .

Given the above concept, a fuzzy set can be viewed as a generalization of a crisp set whose membership function only takes either 0 or 1. Unlike fuzzy sets, a crisp set is a collection of elements with precise bounds that separate the elements that belonged to a set from the elements outside the set (Figure 2.6a):

$$\lambda_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases} \quad (2.6)$$

where $\lambda_A(x)$ denotes the degree of belongings of x in set A .

A fuzzy set is called *normal* if there exist at least one element $x \in X$ such that $\mu_A(x) = 1$, more precisely, $\max(\mu_A(x)) = 1$. A fuzzy set that does not satisfy this

condition is called *subnormal* (Figure 2.6b). Typically fuzzy sets of input variables are normal, and fuzzy sets of output variables are subnormal (Kecman, 2001).

a) Crisp versus fuzzy sets

b) Normal versus subnormal fuzzy sets

c) Convex versus non-convex fuzzy sets

d) Alpha-cut of fuzzy set.

Figure 2.6: Characteristics of fuzzy sets

A fuzzy set A is a convex fuzzy set (Figure 2.6c) if:

$$\mu_A(\lambda x_1 + (1 - \lambda)x_2) \geq \mu_A(x_1) \cap \mu_A(x_2) \quad (2.7)$$

where $x_1, x_2 \in X$, $\lambda \in [0, 1]$.

An alpha cut (α - cut) of fuzzy set A is a crisp set A^α that contains all elements whose degrees of membership in fuzzy set A are greater than or equal to α , where $\alpha \in [0, 1]$, (Figure 2.6d).

$$A^\alpha = \{x | \mu_A(x) \geq \alpha\} \quad \forall x \in X \quad (2.8)$$

An alpha cut can be viewed as a lower bound threshold. Depending on the α value, the size of an alpha cut will be more or less smaller than its original size.

(ii) Operations on fuzzy sets

Basic operations on fuzzy sets include *intersection*, *union* and *complement* (Yager and Filev, 1994). The intersection between fuzzy sets A and B defined over the same universe of discourse X is a new fuzzy set $A \cap B$ (Figure 2.7) that contains elements belonged to both A and B , interpreted as a logical *AND*. The membership function of the new fuzzy set is obtained using the *MIN* operation:

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (2.9)$$

The intersection can be generalized for n fuzzy sets over the same universe of discourse:

$$\mu_{\cap}(x) = \mu_1(x) \cap \mu_2(x) \cap \dots \cap \mu_n(x) \quad (2.10)$$

Figure 2.7: Intersection and union operations of fuzzy sets A and B

The union of fuzzy sets A and B defined over the same universe of discourse X is a new fuzzy set $A \cup B$ that contains elements belonged to either A or B , interpreted as the logical operator *OR*. The membership function of the new fuzzy set is obtained using the *MAX* operation:

$$\mu_{A \cup B}(x) = \text{MAX}(\mu_A(x), \mu_B(x)) \quad (2.11)$$

The union can be generalized for n fuzzy sets over the same universe of discourse:

$$\mu_{\cup}(x) = \mu_1(x) \cup \mu_2(x) \cup \dots \cup \mu_n(x) \quad (2.12)$$

In fuzzy logic theory, the intersection operations are called *T-norms*, and the union operations are called *S-norms* (Mendel, 1995).

Finally, the complement of a fuzzy set is a new fuzzy set with a membership function representing the degree of exclusion or irrelevance from the current fuzzy set:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (2.13)$$

(iii) Defuzzification

The outputs of fuzzy inference are typically subnormal fuzzy sets. Consequently, the aggregation of these fuzzy sets is also a subnormal fuzzy set. For practical applications, a crisp value from this aggregation should be produced. The procedure to derive a crisp value from the aggregated fuzzy set is called defuzzification.

$$y' = \text{defuz}(Y) \quad (2.14)$$

where y' is the crisp value of fuzzy set Y ; *defuz* denotes the defuzzification operator.

A number of defuzzification methods are used in practice, including the Centre-Of-Area (COA), the Mean-Of-Maximums (MOM), and the First-Of-Maximum (FOM), each of which has specific advantages in terms of computational simplicity and computing speed. The most frequently used technique is the COA method. The

COA finds the central point of specific regions such as the overall aggregated output or the centre of the largest area. The method computes the crisp value as:

$$y' = \frac{\int_S y\mu(y)dy}{\int_S \mu(y)dy} \quad (2.15)$$

where S is the support of $\mu(y)$. If S is discrete the value y' can be approximated by the summation instead:

$$y' = \frac{\sum_{i=1}^n y_i\mu(y_i)}{\sum_{i=1}^n \mu(y_i)} \quad (2.16)$$

where n denotes the number of output membership functions. The approximation with Equation (2.16) is accurate when the membership functions are defined as fuzzy sets comprising singletons, where y_i represents the location of each singleton in the output universe of discourse.

(iv) Fuzzy relation

Fuzzy relations express the degree of association, interaction, or interconnectedness between elements of two or more fuzzy sets (Klir and Folger, 1988). Examples of fuzzy relations are: if x is small then y is large; if x is medium then y is medium; if x is large then y is small. The purpose of establishing relationships among fuzzy sets is to characterize the properties of connection from a fuzzy set to another.

Relation between two fuzzy sets on the same product space is also a fuzzy set. A fuzzy relation between fuzzy sets x_1 and x_2 can be expressed by a fuzzy relational matrix, whose elements are degrees of membership $\mu_R(x_1, x_2)$. Let X_1 and X_2 be two universes of discourse. A fuzzy relation $R(X_1, X_2)$ is a fuzzy set in the product space $X_1 \times X_2$ and is characterized by the membership function $\mu_R(x_1, x_2)$ where $x_1 \in X_1$ and $x_2 \in X_2$.

$$R(X_1, X_2) = \{(x_1, x_2), \mu_R(x_1, x_2) \mid (x_1, x_2) \in X_1 \times X_2\} \quad (2.17)$$

The fuzzy relations of different product spaces can be combined with each other by compositional inference. There are several compositional operations, most commonly used are MAX-MIN and MAX-PROD. Let $R_1(x, y), (x, y) \in X \times Y$ and $R_2(y, z), (y, z) \in Y \times Z$ be the two fuzzy relations. The MAX-MIN composition of the two fuzzy relations is the fuzzy set $R_1 \circ R_2(x, z)$ whose membership function is defined as:

$$\mu_{R_1 \circ R_2}(x, z) = \left\{ (x, z), \max_y [\min(\mu_{R_1}(x, y), \mu_{R_2}(y, z))] \right\} \quad (2.18)$$

and the MAX-PROD composition is the fuzzy set $R_1 \otimes R_2(x, z)$ whose membership function is defined as:

$$\mu_{R_1 \otimes R_2}(x, z) = \left\{ (x, z), \max_y [\mu_{R_1}(x, y) \cdot \mu_{R_2}(y, z)] \right\} \quad (2.19)$$

The composition of a fuzzy set and a fuzzy relation can be conducted in the same way. The values of relational composition can be represented in a matrix format, as will be introduced in Chapter 5.

For composition of multiple fuzzy relations, a simple way is to compose them subsequently in pairs. Figure 2.8 presents a schematic diagram of the compositional inference of three fuzzy relations $R_1(x, y), (x, y) \in X \times Y$, $R_2(y, z), (y, z) \in Y \times Z$, and $R_3(z, w), (z, w) \in Z \times W$. The compositional procedure starts with composing $R_1(\cdot)$ and $R_2(\cdot)$ for $R_1 \circ R_2(\cdot)$. The next step proceeds with composing $R_1 \circ R_2(\cdot)$ with $R_3(\cdot)$ for $R_1 \circ R_2 \circ R_3(\cdot)$. The procedure can be conducted in the same way for more relations, and in any order.

Figure 2.8: Composition of multiple fuzzy relations

2.3.3 Membership functions

A membership function generally consists of three main parts: the support, the boundary, and the core (Figure 2.9). The support of a membership function for a fuzzy set A is the crisp subset of X whose elements have nonzero membership values. The support is decomposed into the boundary and the core, where the boundary indicates the transitional part of membership function whose membership value is non-zero but less than unity, and the core is the crisp subset of X consisting of all elements with membership values equal the unity:

The support: $\mu_A(x) > 0, \forall x \in X$ (2.20)

The boundary: $0 < \mu_A(x) < 1, \forall x \in X$ (2.21)

The core: $\mu_A(x) = 1, \forall x \in X$ (2.22)

Figure 2.9: Three parts of a membership function

There are many geometric forms of membership functions used in practice, but they can be broadly grouped in two categories: piece-wise linear and curve (MATLAB 6.5, User Manual, 2002). The piece-wise linear consists of lines connected at different break points. Triangular and trapezoidal shapes are frequently used in this style.

(i) Triangular membership function is represented by a function of a vector X , and three parameters a, b, c (Figure 2.10a), compactly written as:

$$f(x, a, b, c) = \max \left(\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \quad (2.23)$$

(ii) Trapezoidal membership function is represented by a vector X and located by 4 parameters a, b, c, d (Figure 2.10b), compactly written as:

$$f(x, a, b, c, d) = \max \left(\min \left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right), 0 \right) \quad (2.24)$$

(iii) Sigmoidal membership function is a mapping on the vector X , and depends on two parameters, α and β (Figure 2.10c)

$$f(x, \alpha, \beta) = \frac{1}{1 + e^{-\alpha(x-\beta)}} \quad (2.25)$$

Depending on the sign of α the sigmoidal membership function is open to the right or to the left.

(iv) Symmetrical Gaussian membership function (Figure 2.10d) is defined by two parameters σ and c , given by:

$$f(x; \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}} \quad (2.26)$$

Figure 2.10: Primary types of membership functions

2.3.4 Fuzzy logic system design

(i) Fuzzy variables

Fuzzy variables are variables whose values are expressed in natural languages. For example, if speed V is a linguistic variable, its values may be represented as *slow*, *medium*, and *fast*. Since linguistic words are more general and communicable, they are commonly used in complex and ill-defined concepts, or in situations in which the application of numerical values is not suitable (Teodorović and Vukadinović, 1998). In a fuzzy rule, a linguistic variable in the premise forms a fuzzy input space, and in the conclusion forms a fuzzy output space.

Figure 2.11 illustrates the structural concept that addresses essential issues in fuzzy variable design. The design process encompasses several layers, including the selection of linguistic predicates, the design of membership functions, and the determination of the universe of discourse. The proper selection of linguistic variables and their associated predicates is important to the operation characterization, and has a substantial effect on the performance of the FLS. Reasoning logic and engineering knowledge play an important role in this selection stage. The specification of the universe of discourse of a fuzzy variable determines the domain in which the fuzzy rules are valid. The universe of discourse can be continuous or discontinuous, normalized or non-normalized. A common method to

specify the universe of discourse is to consider the range in which numerical data locate, defined by the *min* and the *max* values.

The second layer addresses the most important issue in a FLS: the design of membership functions, which represent the uncertainty distribution captured in each of fuzzy sets. The performance of a FLS is significantly dependent on the input-output mapping, which is in turn dependent on the membership function design. Unfortunately, design of membership functions is particularly challenging since it must employ a body of knowledge at different levels to address complicated issues such as the number of membership functions, the shape, the location, and the overlap among fuzzy sets.

Figure 2.11: Structure of a fuzzy variable design

(Source: Berkan and Trubatch, 1997)

Methodologies in membership function design can be basically categorised into expert-oriented approach (Turban, 1992, Hong and Chen, 1999), and data-driven approach (Lee, 1990; Hong and Lee, 1996; Hong and Chen, 1999). In Figure 2.11, the expert-oriented design is a top-down method, while the data-driven approach is a bottom-up procedure.

The traditional expert-oriented approach has been widely used in the initial stage of the FLS development. The approach requires heuristic logics and expert judgments

to define membership functions in absence of proven relationships between variables and distributions of fuzziness in corresponding fuzzy sets. Three methods frequently used include the prototype categorization, the degree of similarity, and the similarity of distances (see Ross (1995) for details). Applications of these methods require the assumptions that the FLS being considered has non-interactive inputs, and knowledge regarding the input-output relationships is at expert level.

The data-driven approach utilizes numerical data to identify inference mechanisms that define mapping a specific input region into a corresponding output region. This approach is more suitable when there exist a small number of variables and the rules are monotonically and structurally simple (Berkan and Trubatch, 1997). Methods in this approach include fuzzy C-Means (Chen and Wang, 1999; Sintas et al., 1999), genetic algorithm (Ross, 1995; Bogenberger and Keller, 2001), inductive reasoning (Ross, 1995), and measurement theory (Chen and Kevin, 1995).

The number of membership functions (granularity) or the number of predicates that suffice a fuzzy variable is determined through fuzzy partitioning. Linguistic hedges such as *very*, *slightly*, *almost*, *more or less*, *etc.*, are frequently used in fuzzy partition to modify fuzzy sets for a desirable granularity. For instance, the speed V universe of discourse described above can be represented by low or high resolution:

$V = \{Low, Medium, High\}$ (Figure 2.12a), or

$V = \{Very_low, Low, Medium, High, Very_high\}$ (Figure 2.12b).

Figure 2.12: The fuzzy partition in the universe of discourse

It should be noted that the partition of the input-output space has no unique solution. A heuristic cut-and-trial procedure is sometimes applied to find the optimal fuzzy partition for specific applications.

The next step in establishing membership functions is the proper selection of their shapes, including the line style and the width of the support. The geometric shape of a membership function reflects the distribution of fuzziness in the corresponding fuzzy variable, and specifies the way in which uncertainty is characterized. In comparison, the piece-wise linear style is simple and straightforward, while the curve style is gentler in reflecting transition of membership grade, but is more complex, hence more computational effort is required. The piece-wise linear is often used when there is a lack of knowledge on the fuzziness in corresponding fuzzy sets. In the piece-wise linear style (Figures 2.10a and 2.10b), however, the triangular form is more conservative where only one point qualifies the complete membership degree, while the trapezoidal shape is more tolerant in that it allows the corresponding rules to fire at full strength over a larger part of the universe of discourse.

In addition to the line style, it is necessary to consider the support of a membership function. The width of the support directly affects the fuzziness. The wider the support, the higher fuzziness the set has, and the higher number of elements to be categorized in the fuzzy set. Conversely, two fuzzy sets with the same support width but different line styles produce different levels of effect on the final output.

Berkan and Trubatch (1997) highlighted several important issues in designing membership functions, including the shape, the location, the granularity, and the overlap. Among those, the shape affects local performance, while the location and granularity govern the overall performance of the inference system. Particularly, the shape effect emerges when membership functions of the state variable are different from each other. When membership functions have the same shape, the effect is less profound. Consequently, the location and the granularity are relatively more important issues. If the location and granularity are poorly designed, the shape adjustment cannot compensate for the poor performance. Finally, the degree of overlapping specifies the degree of corporation between the corresponding rules.

More overlap means more cooperation between fuzzy sets in the condition, and smoother transition between the two crisp points in the consequent space.

(ii) Fuzzy rule base

The fuzzy rule base is the kernel of a FLS. It houses a collection of IF-THEN rules where the IF part contains a set of conditions and the THEN part contains a conclusion. If the conditions are satisfied, the conclusion is applied. Unlike the rule-based expert system, the rules comprising a rule-based system may be derived from sources other than experts, such as from intuition, commonsense reasoning, engineering knowledge, and data-driven learning.

Most rule bases consist of more than one rule. Each activated rule in the rule base has its consequent triggered by the fuzzy inference engine. The process of obtaining the output of a rule is known as fuzzy implication. Fuzzy implication uses the MIN and MAX operations to infer the output (Section 2.3.2, Equations (2.9) to (2.12)). By contrast, the process of obtaining the overall output from individual outputs of different rules is known as aggregation. There are two methods of aggregation: the conjunctive method and disjunctive method (Ross, 1995).

The conjunctive method: in the case when all rules must be simultaneously satisfied, the consequents of rules are joined with an *AND* connective. The aggregation of rules (y) is found by intersection operation, using *MIN* operator.

$$y = y_1 \cap y_2 \cap \dots \cap y_n \quad (2.27)$$

And the value of the aggregated membership function is defined by:

$$\mu_{\cap}(y) = \mu_1(y) \cap \mu_2(y) \cap \dots \cap \mu_n(y), \forall y \in Y \quad (2.28)$$

where n is the number of activated rules in the rule base, and $\mu_i(y)$, $i = 1, 2, \dots, n$ is the degree of fulfilment of each rule.

The disjunctive method: in the case where the condition is the satisfaction of at least one rule, the rules are connected by an *OR* connective. The aggregation of rules (y) is found by the union operation, using *MAX* operator.

$$y = y_1 \cup y_2 \cup \dots \cup y_n \quad (2.29)$$

which is defined by the membership function:

$$\mu_{\cup}(y) = \mu_1(y) \cup \mu_2(y) \cup \dots \cup \mu_n(y), \forall y \in Y \quad (2.30)$$

Figure 2.13 demonstrates a simple example illustrating fuzzy operations (*MAX*, *MIN*) and defuzzification methods (*COA*, *FOM*, and *MOM*). The input variables of the FLS include traffic speed (V) and density (K), which are used to determine the congestion level (CL) in the product space $V \times K$. Assume that the relationship between the input and output of the system is characterized by a set of 4 rules:

- Rule 1: if $V = Low$ OR $K = High$ then $CL = Heavy$
- Rule 2: if $V = Medium$ AND $K = Medium$ then $CL = Moderate$
- Rule 3: if $V = Medium$ AND $K = Low$ then $CL = Light$
- Rule 4: if $V = High$ AND $K = Low$ then $CL = F_Flow$.

In rule 1 the two conditions on V and K are joined by an *OR* connective, therefore, the membership values of rule 1 are calculated using the *MAX* operator (union operation):

$$\mu_{CL} = MAX(\mu_V(v^*), \mu_K(k^*)) \quad (2.31)$$

By contrast, in rules 2, 3, 4 the two conditions are joined by an *AND* connective, hence the membership values of the rules are calculated using the *MIN* operator (intersection operation):

$$\mu_{CL} = MIN(\mu_V(v^*), \mu_K(k^*)) \quad (2.32)$$

For the given values of $V (v^*)$ and $K (k^*)$ rules 1, 2, and 3 are active, rule 4 is inactive. The outputs from the system are subnormal fuzzy sets, whose results are aggregated using the *MAX* operator. The defuzzifier is then used to produce the single crisp output by one of *COA*, *FOM*, or *MOM* method.

Figure 2.13: Fuzzy implication, aggregation and defuzzification

2.3.5 Fuzzy logic control

Fuzzy logic control is a control law described by a rule-based system with vague predicates and a fuzzy logic inference mechanism (Yager and Filev, 1994). A fuzzy logic controller (FLC) provides a means to convert a linguistic control to an automatic control strategy. In designing a FLC, the following assumptions are implicitly made (Ross, 1995):

- The physical system under control is observable and controllable: the state, input and output variables are usually available for observation and measurement.
- There exists a body of knowledge comprising linguistic rules, engineering commonsense, intuition, input and output data that can be fuzzified and from which rules can be extracted.
- The goal of the FLC is sub-optimal, not necessarily the optimum solution.

Figure 2.14 shows the conceptual diagram of a FLC. A standard FLC consists of 4 components:

Figure 2.14: Conceptual diagram of a FLC

(i) The fuzzification measures the values of input data and performs a scale mapping that converts the data into suitable linguistic values with associating membership degrees.

(ii) The fuzzy knowledge base comprises knowledge of the application domain. It consists of a database and a fuzzy rule base. The database provides relevant historical data to derive membership functions and rules, and real-time data as input for the FLC operation. Basically, the fuzzy rule base characterizes the control policies of the system in the IF-THEN format. Depending on the number of state and control variables, the fuzzy rule base represents a single-input-single-output (SISO), multiple-inputs-single-output (MISO), or a multiple-inputs-multiple-outputs (MIMO) system.

There are four methods for extracting control rules (see Takagi & Sugeno (1983), in Lee (1990) for details), including:

- Rules defined by expert experience and control knowledge
- Rule extraction from operator's control actions
- Rule generation based on a fuzzy model of the process, and
- Rule derivation based on learning capability.

(iii) The inference engine is the kernel of a FLC. It has the capability of simulating human decision making based on fuzzy concepts and the capability of inferring fuzzy control actions employing the rules of inference. The inference engine performs the interpretation of the sentence connectives *AND* and *OR* to calculate the degree of fulfilment of each rule, then the conclusions of all active rules are aggregated using the *MIN* or *MAX* operator (Figure 2.13).

(iv) The defuzzification converts the aggregated values of control variables into single crisp values using one of defuzzification methods, as described in Section 2.3.2.

The FLC architecture presented in Figure 2.14 is an improvement of the traditional FLC architectures, such as Mamdani's FLC and Takagi-Sugeno's FLC (Jian, 2001). The main disadvantage of the traditional Mamdani's FLC is that the inputs and outputs are fuzzy sets. This creates problems particularly for engineering applications since in most engineering problems the inputs are available in

numerical data, and the outputs must be crisp. By contrast, in the Takagi-Sugeno's FLC the inputs and outputs are real values, so the learning of fuzzy rules from human perception is not facilitated. The FLC presented in Figure 2.14 circumvents these problems by adding fuzzification and defuzzification interfaces to facilitate the use of numerical inputs and outputs. In addition, it provides the natural framework to incorporate fuzzy IF-THEN rules and choices of alternative versions of fuzzifier, fuzzy inference engine, and defuzzifier so as to obtain the most suitable FLC.

2.3.6 Fuzzy logic systems for traffic control

(i) Fuzzy logic system's applications in transportation

A great number of problems in the fields of transportation and traffic engineering are characterized by variables that are uncertain, imprecise and ambiguous. It is well recognized that approximate reasoning is an intrinsic nature in driver behaviour, driver perception, route choice, vehicle route guidance, scheduling, project assessment, traffic safety, and especially - traffic control. Fuzzy logic is an attractive and feasible approach in handling these problems. Teodorović and Vukadinović (1998), Teodorović (1999) and Hoogendoorn et al. (1998) made comprehensive reviews of successful applications of fuzzy logic systems in transportation and traffic engineering:

Pappis and Mamdani (1977) were the pioneers in applying the fuzzy logic theory for traffic control for isolated signalised intersection. Subsequently, Nakatsuyama et al. (1983), Sugeno and Nishida (1985), Sasaki and Akiyama (1987, 1988) made great contributions to fuzzy set theory applications in traffic and transportation (Teodorović and Vukadinović, 1998). In the field of transportation engineering, fuzzy logic systems have been widely employed, ranging from trip generation (Chan et al., 1986; Xu and Chan, 1993), modal split (Teodorović and Kalić (1996) in Teodorović, 1999), and vehicle routing and scheduling (Olaru and Smith, 2005). In traffic engineering, there has been a great deal of work for various applications such as incident detection (Lin and Chang, 1998; Xu et al., 1998; Sheu, 2002) and route choice (Henn, 2000; Arslan and Khisty, 2005).

Teodorović (1999) made a good review on the merits of fuzzy logic applications in transportation and traffic engineering: in the afore-mentioned applications, the fuzzy logic systems in general have produced promising results. Particularly, in trip distribution and modal split “very small relative deviations between the values calculated using the approximate reasoning algorithms and the real values”. In route choice problem “in particular cases, fuzzy logic models gave considerably better results than those obtained from the Logit model”. However, the use of hypothetical numerical examples may hinder the merits of the models for actual applications.

Subsequent research in the area of traffic engineering also found evidences that the fuzzy logic applications in congestion management and incident detection, intersection control and ramp-metering control are generally very promising (Hoogendoorn et al., 1998). The following section makes a brief review of fuzzy logic systems’ applications for traffic control.

(ii) Fuzzy logic systems for traffic control

Traffic control is one of the earliest applications of fuzzy logic systems in traffic engineering. In this area, attempts have been made in investigating fuzzy logic for traffic control at isolated intersections in urban networks as well as ramp control, speed control, and corridor control in urban expressways.

Trabia et al. (1999) presented the design and evaluation of a two-stage FLC for traffic signals at an isolated four-approach intersection with through and left-turning movements. The FLC estimated relative traffic intensities and queues in the competing approaches at regular time intervals in the first stage, and then the estimated traffic intensities were used in the second stage to determine whether the current signal phase should be extended or terminated. The FLC has the ability to make adjustments for signal timing in response to observed changes in the approach flows and queue lengths. The FLC is found to produce lower vehicle delays than the conventional traffic-actuated controller while maintaining the same fraction of stopped vehicles.

Sasaki and Akiyama (1988) described the operators' judgments by fuzzy logic rules and build the process of traffic control that automatically determines the pattern of onramp of the expressway. The data obtained from Hanshin expressway in 5-minute interval were used to investigate the performance of the FLC that used two fuzzy inputs - the length of traffic congestion and the expected traffic demand - to determine the number of open booths of each on-ramp. The results showed that the FLC was effective in describing the operators' judgments.

Chen et al. (1990) presented the application of an expert fuzzy controller to entrance ramp control at the San Francisco-Oakland Bay Bridge. A fuzzy rule base was developed in response to both incident and non-incident situations using expert knowledge from bridge operators and an existing automatic controller. Primary control objectives include maximized overall system throughput, safety, and minimized ramp queuing. The FLC used six input variables including "congestion level", "change in congestion level", "control area", "incident", "non-incident" and "size of control area queue" to produce a change in ramp flow from the nominal base rate. In most cases, the FLC showed significant improvements over the existing automatic controller. Tests under a variety of scenarios with different incident locations and capacity reductions showed that the FLC could obtain from 40 to 100% of the travel time savings. In addition, the FLC exhibited smooth response to incidents, and significantly reduced the magnitude of congestion.

Ngo and Victor (1994) described a bi-level model-free control scheme using neural-network-based fuzzy logic controllers (NN-FLC) to regulate the speed of the freeway. The NN-FLC was capable of handling qualitative information by fuzzy logic and numerical data by neural-network so as to utilize both qualitative and quantitative information. The advantage of the NN-FLC was that the control strategy is computationally simple and easy to implement. Their initial-phase implementation with the local controller indicated that the results were comparable with those obtained by an analytical controller.

Taylor and Meldrum (2000) carried out a comparative study that evaluated performances of three ramp-metering algorithms named as Fuzzy Logic, Local, and Bottleneck algorithms at two study sites in the Greater Seattle area. The Local

algorithm used data only in the vicinity of the ramp, whereas the Bottleneck algorithm included the downstream inputs. The Fuzzy Logic algorithm used the same detector data as the Local algorithm and Bottleneck algorithm at the corresponding sites. The performances of Fuzzy Logic algorithm were compared to that of the Local algorithm on the moderately congested Westbound I-90, and to that of the Bottleneck algorithm on the heavily congested Southbound I-405 over a broad range of congestion conditions. The results showed that at the first study site the Fuzzy Metering produced lower mainline occupancies, higher throughput, but slightly longer queues. At the second study site, the Fuzzy Metering produced higher mainline occupancies and throughput, and significantly shorter queues. The authors claimed that with the combination of better mainline efficiency, and similar or better queue management, the Fuzzy Logic Ramp Metering algorithm proved to be a suitable tool for expressway traffic control.

Bogenberger and Keller (2001) proposed a nonlinear approach for designing traffic responsive and coordinated ramp control using a self-adapting fuzzy system, which was used to determine the metering rate at every minute using the local speed, flow and occupancy on the mainline upstream of an on-ramp, the volume/capacity ratio downstream, and the queue length on the ramp. The coordination between several on-ramps was conducted by the integration of all ramp controllers upstream of a bottleneck into a common input. A Genetic algorithm was used to determine the optimal coordinated parameters of the fuzzy ramp metering controllers every 15 minutes based on a macroscopic traffic flow model. Typical fuzzy rules in the rule base were defined as follows:

If local occupancy is small then metering rate is high

.....

If check-in occupancy or queue occupancy is very high then metering rate is high.

The evolutionary fuzzy system was tested for the A9 Munich Autobahn (Germany) using the FREQ model - a macroscopic, deterministic simulation model for a directional freeway corridor. Three scenarios were studied: the adaptive-fuzzy-control-scenario (AFC), the no-control scenario (NC) and the linear-programming

scenario (LP). The result showed that the AFC allowed a total travel time saving of 6.1% and 4% compared to the NC and the LP respectively, while the mainline speed was 2.9% higher than the NC, and the same as that of the LP.

2.4 SUPPORT VECTOR MACHINES

Support Vector Machine is a family of learning algorithms, currently considered among the most efficient instruments for pattern recognition and regression problems in many applications. It possesses a good generalization capability, computational efficiency, and is very robust in high dimensions. It provides a natural framework for solving non-linear time-series problems, an inherent characteristic of traffic flows. For these reasons, this study investigates the SVM's application potentials for forecasting of traffic variables.

In the following paragraphs, the essence of the SVM theoretical background will be briefly introduced in light of those provided by Kecman (2001), Vert (2001) and Gunn (1998). The introduction starts with the linear classification, which is subsequently extended to the non-linear classification problem (SVM), and the non-linear regression problem (SVR). Interested readers may refer to Vapnik (1995), Smola and Scholkopf (1998), Cristianini (2000), and Scholkopf and Smola (2002) for more details on the mathematical foundation of SVM theory.

2.4.1 Linear classification problem of SVM

Consider a set of training data S with N observations $(\vec{x}_i, y_i)$

$$S = \{(\vec{x}_1, y_1), \dots, (\vec{x}_N, y_N)\} \quad (2.33)$$

where $\vec{x}_i \in \mathbb{R}^m$, $y_i \in \{-1, +1\}$, $\forall i = 1, \dots, N$

The goal is to find an optimal hyperplane (linear classifier) from the data set that correctly classifies all members in S . The linear classifier is defined by two vectors $\vec{w}$ and b .

$$\begin{cases} \bar{w}_i \bar{x}_i + b \geq 1, \forall y_i = +1 \\ \bar{w}_i \bar{x}_i + b \leq -1, \forall y_i = -1 \end{cases} \quad (2.34)$$

The optimal hyperplane is the one that minimizes the empirical risk and the Vapnik-Chervonenkis (VC) dimension (see Kecman (2001) for definition). In SVM theory, the VC dimension is related to the smallest distance from a point to the separating hyperplane in the input space, known as the margin (Vert, 2001), denoted as γ in Figure 2.15a. The optimal hyperplane is the one that has the largest margin for a given training set.

Figure 2.15: Optimal hyperplane in a two-dimensional problem

The optimal hyperplane is defined by a pair $(\bar{w}, b)$ that satisfies the constraints (2.34) such that $\|\bar{w}\|$ is minimum. This is a constrained optimisation problem that can be formulated as minimizing a quadratic function under linear constraints:

$$\begin{aligned} &\text{Minimize: } \|\bar{w}\|^2 \\ &\text{Subject to: } y_i(\bar{w} \cdot \bar{x}_i + b) - 1 \geq 0, \text{ for } i=1, \dots, N. \end{aligned} \quad (2.35)$$

In order to generalize to non-separable training sets and non-linear SVM, this classical optimisation problem is formulated as a dual formulation by introducing the Lagrangian function:

$$L(\vec{w}, b, \vec{\lambda}) = \|\vec{w}\|^2 - \sum_{i=1}^N \lambda_i [y_i (\vec{w} \cdot \vec{x}_i + b) - 1] \quad (2.36)$$

where the dual vector $\vec{\lambda} = (\lambda_1, \dots, \lambda_N)$ is the Lagrange multiplier, and b is a real number. The minimization problem requires the differentiation of L with respect to $\vec{w}$ and b :

$$\frac{\partial L}{\partial \vec{w}}(\vec{w}, b, \vec{\lambda}) = 0 \Rightarrow \vec{w}_{\vec{\lambda}} = \frac{1}{2} \sum_{i=1}^N \lambda_i y_i \vec{x}_i \quad (2.37)$$

$$\frac{\partial L}{\partial b}(\vec{w}, b, \vec{\lambda}) = 0 \Rightarrow \sum_{i=1}^N \lambda_i y_i = 0 \quad (2.38)$$

Substitute Equations (2.37) and (2.38) into (2.36) and let $W(\vec{\lambda})$ denote the minimum value of L when $\vec{\lambda}$ is fixed and $(\vec{w}, b)$ varies without constraint, and $\vec{\lambda}^* = (\lambda_1^*, \dots, \lambda_N^*)$ denote the vector of solution. The above minimization problem is summarised as follows (Vert, 2001):

Maximize

$$W(\vec{\lambda}) = L(\vec{w}_{\vec{\lambda}}, b_{\vec{\lambda}}, \vec{\lambda}) = \sum_{i=1}^N \lambda_i - \frac{1}{4} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \lambda_i \lambda_j \vec{x}_i \cdot \vec{x}_j \quad (2.39)$$

subject to:

$$\sum_{i=1}^N \lambda_i y_i = 0 \quad (2.40)$$

$$\lambda_i \geq 0, \forall i = 1, \dots, N. \quad (2.41)$$

then the pair $\vec{w}^*, b^*$ defines the hyperplane:

$$\begin{cases} \vec{w}^* = \sum_{i=1}^N \lambda_i^* y_i \vec{x}_i, \\ b^* = -\frac{1}{2} \left[\min_{i: y_i = +1} (\vec{w}^* \cdot \vec{x}_i) + \max_{i: y_i = -1} (\vec{w}^* \cdot \vec{x}_i) \right] \end{cases} \quad (2.42)$$

In the above dual formulation problem, the dual vector $\vec{\lambda}^* = (\vec{\lambda}_1^*, \dots, \vec{\lambda}_N^*)$ is found to determine the optimal hyperplane $(\vec{w}^*, b^*)$. Each element of the dual vector represents one constraint. There are two kinds of constraints: active constraints for $\lambda_i > 0$ and inactive constraints for $\lambda_i = 0$. The active constraints correspond to the points whose distance to the optimal hyperplane is exactly equal to the margin. They are called *support vectors* since they give shape to the solution (the points enclosed in circles in Figure 2.15a). The smaller the fraction of support vectors the more general the solution is, and less computation is required to evaluate the solution for new objects. On the other hand, training points with inactive constraints do not contribute to the solution. If these training points were removed in advance, the same solution would have been obtained.

Once the parameters λ^* and b^* have been found, the classification of a new observation $\vec{x}$ only requires the computation of the dot product between $\vec{x}$ and every support vector. Since the number of support vectors is very small compared to the training data, the classification of the new problem is often very fast.

2.4.2 Non-linear classification problem of SVM

In the above section the training set $S = \{(\vec{x}_1, y_1), \dots, (\vec{x}_N, y_N)\}$ is assumed linearly separable. In actual applications this assumption usually does not hold since the classification of two classes requires more sophisticated shape. SVM is extended to handle such cases by non-linear mapping using feature space.

Suppose that one can define a set of feature functions $\phi_1, \dots, \phi_M$ on the object space. Then an object $\vec{x}$ can be mapped to a higher dimensional feature space through the following transformation:

$$\vec{x} = (x_1, \dots, x_m) \rightarrow \phi(\vec{x}) = (\phi_1(\vec{x}), \dots, \phi_M(\vec{x})) \quad (2.43)$$

Having mapped all the points from the training set to the feature space, the following set of points:

$$\phi(S) = \{\phi(\vec{x}_1, y_1), \dots, \phi(\vec{x}_N, y_N)\} \quad (2.44)$$

is obtained in the feature space $\Re^M$. An interesting feature is that the training set $\phi(S)$ can be linearly separated in the feature space. In the previous section it has been found that the classifiers classify new example $\vec{x}$ depending on the sign of the function $f(\vec{x})$ defined in Equation (2.45):

$$f(\vec{x}) = \sum_{i=1}^N y_i \lambda_i^* \phi(\vec{x}_i) \phi(\vec{x}) + b^* \quad (2.45)$$

Equation (2.45) shows that the mapping can be constructed using the dot product $\phi(\vec{x}_i) \bullet \phi(\vec{x})$. The Kernel function $K(\vec{x}, \vec{x}') = \phi(\vec{x}) \bullet \phi(\vec{x}')$ is introduced to implicitly map the input data into the feature space. Kernels popularly used in practice include Polynomial, Radial Basic Function (RBF), and Sigmoid kernels. Since traffic flow relationships are highly non-linear, the RBF will be used in this study. The RBF has the form:

$$K(\vec{x}, \vec{x}') = \exp\left(-\frac{\|\vec{x} - \vec{x}'\|^2}{2\sigma^2}\right) \quad (2.46)$$

where σ is a kernel parameter.

2.4.3 Support vector regression

The SVM theory presented above for binary classification problems has been extended to solving regression problems. Unlike classification problems, in SVR outputs y_i are real values (Kecman, 2001). Given a training data set:

$$S = \{(\vec{x}_i, y_i) \in \Re^m \times \Re, i = 1, \dots, N\} \quad (2.47)$$

where the input $\vec{x}$ are m -dimensional vectors, $\vec{x} \in \mathfrak{R}^m$, and the responses y_i are continuous values. The basic idea is to map the input data $\vec{x}$ into a high dimensional feature space through a non-linear mapping ϕ :

$$f(\vec{x}, \vec{w}) = \sum w_i \phi_i(\vec{x}) \quad (2.48)$$

where the functions $\phi(\vec{x})$ are called features, and the weights $\vec{w}$ are the subjects of learning. Therefore, linear regression in a high dimensional feature space corresponds to non-linear regression in a low dimensional input space.

SVM is extended to regression problems by introducing an alternative loss function that is modified to include a distance measure (Smola and Scholkopf, 1998). There are 4 popular types of loss functions: Quadratic, Laplace, Huber and ε -insensitive.

For linear regression, the optimal regression is given by:

$$\phi(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N (\xi_i^- + \xi_i^+) \quad (2.49)$$

where C is a pre-specified value; ξ^+, ξ^- are slack variables representing the lower and upper constraints on the system outputs.

Similar to classification problems, in regression problems a non-linear model is usually required for non-linearly mapping the input data into a high dimensional feature space, and the Kernel approach is adopted. In non-linear regression, the ε -insensitive loss function type is used, and the regression function is given by:

$$\max_{\lambda, \lambda^*} W(\lambda, \lambda^*) = \max_{\lambda, \lambda^*} \sum_{i=1}^N \lambda_i^* (y_i - \varepsilon) - \lambda_i (y_i + \varepsilon) - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i^* - \lambda_i)(\lambda_j^* - \lambda_j) K(x_i, x_j) \quad (2.50)$$

under the constraints:

$$\begin{cases} 0 \leq \lambda_i, \lambda_i^* \leq C, \forall i \in \{1, \dots, N\} \\ \sum_{i=1}^N (\lambda_i - \lambda_i^*) = 0 \end{cases} \quad (2.51)$$

The Lagrange multipliers λ_i, λ_i^* are found by solving Equation (2.50) with constraints (2.51). The solution is given by:

$$f(x) = \sum_{i=1}^N (\bar{\lambda}_i - \bar{\lambda}_i^*) K(x_i, x) + \bar{b} \quad (2.52)$$

where

$$\bar{b} = -\frac{1}{2} \sum_{i=1}^N (\lambda_i - \lambda_i^*) (K(x_i, x_r) + K(x_i, x_s)) \quad (2.53)$$

The equality constraints may be dropped if the Kernel contains a bias term b that has been considered in the Kernel function. In this case, the regression function is reduced to:

$$f(x) = \sum_{i=1}^N (\bar{\lambda}_i - \bar{\lambda}_i^*) K(x_i, x) \quad (2.54)$$

Since the classification problem SVM and regression problem SVR share the same theoretical foundation, they are named interchangeably, as is used in this thesis.

2.5 SUMMARY

In this chapter, a number of research works in the field of traffic control have been reviewed. They represent various control approaches and provide important contributions towards the state-of-art control techniques. Nevertheless, the reviewed literatures exposed a number of problems if applied for incident management. It is important to identify critical issues that need to be considered in this research.

(i) In feedback control approach

- Most of the feedback control algorithms were designed to tackle recurrent problems, but not specifically designed for traffic control under incident conditions. Consequently, many incident-related factors such as incident severity, capacity reduction, time of day, queue length, etc., have not yet been considered.
- Most control algorithms used point measurements such as occupancy obtained from detectors. For instance, the ALINEA algorithm uses the occupancy obtained from a detector downstream of the ramp to calculate the ramp flow. The point measurements are not very useful for incident management since they do not properly reflect the prevailing conditions for the whole section. For example, for distant incidents the congested condition upstream of the incident is not visible until the disturbances propagate to the detector.
- The control algorithms are overly biased toward the mainline: the control objectives target the amelioration of the mainline traffic condition at the expense of the ramp traffic. From the systematic point of view, the control objectives should be balanced. It should be noted that the word *balance* mentioned in subsequent related parts in this report does not mean that the mainline and the ramp are equally considered, but it implies a control strategy that gives priority toward the mainline but with a consideration to the ramp condition.

(ii) In FLC approach

The reviewed past works in traffic control following fuzzy logic approach have attempted to utilize the advantages of fuzzy logic in handling multivariable control problem, and the results were encouraging. However, a number of problems are identified:

- The past works generally did not provide a systematic procedure in learning parameters of membership functions. Most of the rules are subjectively established and overly simplified without justifiable reasoning steps.

- Like feedback control, the control algorithms following FLC approach employed only a few input variables, such as volume and occupancy. Many important incident attributes have not been mentioned. Consequently, the multivariable incident management problem was not successfully addressed.
- Most of the reviewed control algorithms were reactive control using past measurements. Sasaki and Akiyama (1988), for instance, estimated the expected demands by gathering historical profiles. As will be investigated in Chapter 6, the historical means incur serious errors in representing the current problem, especially under incident conditions. Without predicted traffic variables, it is very likely that the control scheme suffers severe side effects.

The review of the merits as well as problems of the past works highlights the following issues that need to be considered in this research:

- Utilisation of the available techniques in parametric learning of membership functions. Apart from expert judgements and engineering knowledge, data-based approach such as statistics and fuzzy partitioning should be explored in establishing membership functions and fuzzy logic system as a whole.
- Provision of a systematic procedure in deriving control actions. The control decisions represent a high level of knowledge, thus decision making should experience a systematic reasoning process, from data and information to knowledge level.
- Full consideration of relevant variables. The traffic control for incident situation represents a complex multivariable problem. Therefore, the primary traffic and incident attributes should be thoroughly studied and evaluated in the control scheme.
- The control algorithms should target a proper balance among different objectives, with a reasonable priority toward the mainline. Therefore, besides the mainline associated objectives such as total travel time, average speed and throughput, the system-wide objectives such as the total time

spent, the waiting time on ramps, the queue length, etc., should be thoroughly considered in deriving the control solution.

- The anticipation of incident traffic conditions should be emphasised for proactive control. Although the prediction of incident traffic condition is challenging, it would be very significant if the control decisions are made based on reliable anticipated condition.

In brief, although a lot of works have been done in traffic control, little effort has been specifically targeted to traffic control for incident management following fuzzy logic approach. Essential issues in this field including the parametric learning in membership function design, the systematic procedure in deriving control decisions, the consideration of relevant incident attributes, and the incident-related traffic forecasting, etc., have not yet been adequately explored. These issues will be pursued in this research.

CHAPTER 3 STRUCTURE OF THE PROPOSED INCIDENT MANAGEMENT SYSTEM

3.1 OVERVIEW

This chapter presents the methodological framework of the fuzzy KBS for incident management. The fuzzy KBS has several components, among which the FLC is the core. The FLC receives real-time traffic and incident data to analyse and anticipate the traffic conditions, and to recommend feasible alternative control measures to the human operator in the form of linguistic expressions. Upon the selected control measure, the FLC calculates control settings to manage traffic.

As described in Chapter 1, incidents represent critical and complex multivariable problems where many types of data and information are available, both quantitatively and qualitatively. Under the dynamically changing and unexpected situations, the solutions of traffic control for incident management typically involve semi-structured decisions. Therefore, the KB-DSS should be designed to assist the operator's decision making in a consistent and structured manner.

The decision-making process for traffic control during incidents on expressways involves three major tasks:

- Evaluation of traffic conditions upon an incident occurrence.
- Prediction of traffic congestion tendency during the incident.
- Determination of control strategies and implementation of control actions to alleviate traffic congestion.

The details of the tasks will be elaborated in Section 3.2.2.

The remainder of this chapter is summarized as follows: Section 3.2 presents the conceptual framework of the fuzzy KBS and explains the associated components. Section 3.3 explains the rule formation mechanism and concept of rule block, and reviews typical multi-stage FLC architectures. Section 3.4 proposes a three-stage

FLC for the application in traffic control for incident management and describes the rule base structure of the proposed MS-FLC.

3.2 CONCEPTUAL MODEL

3.2.1 General concept

Figure 3.1 describes the complete structure of the fuzzy KBS's. Being elaborated from the overall architecture of a KB-DSS (Figure 1.1), the proposed KB-DSS consists of 5 main components: the data base, the rule base, the FLC, the GUI and the operator. The following describes three main components of the KBS that lie in the scope of this research (Section 1.3): the data base, the rule base, and the FLC. The data base and the rule base constitute the *Knowledge base*, and the FLC forms the *Model base* of the fuzzy KBS.

Figure 3.1: Components of the fuzzy KBS

The **data base** provides data as facts for the KBS development and implementation. Data from the server database is collected and transformed to develop the HDB and RTDB databases. The HDB stores data for knowledge acquisition following the data-driven approach, such as learning parameters of membership functions and extracting rules. In addition, traffic data from the HDB can be used to construct historical profiles for traffic prediction. Historical incident data can be explored to identify incident-sensitive locations and to learn important incident attributes in the network including the distributions of incidents according to type and duration. The RTDB, on the other hand, provides data as direct input for the FLC. The RTDB is continually rolled forward so that the state variables are continually updated with the latest data and information.

The **rule base** is a crucial component of the knowledge base in the KBS. It is a repository that stores knowledge in the IF-THEN rule format. Rules in the rule base represent specific knowledge in the domain of traffic control and incident management. As described in Chapter 2, two formal approaches to learn rules in the rule base include the data-driven and the expert-oriented. The data-driven approach utilizes data from the HDB for input-output mapping. Other important sources for rule extraction include expert judgments, engineering knowledge, commonsense reasoning, and documented sources. For traffic control, a particularly important source for rule extraction is control experiences from traffic operators.

The **FLC**, alternatively known as the Model base, is the key component of the KBS. The FLC contains a fuzzifier to convert numerical data to fuzzy sets with associating membership degrees, an inference engine to perform input-output mapping, and a defuzzifier to transform aggregated consequents to crisp values (Figure 2.14). It handles the mapping mechanism between state and control variables. For traffic control, inputs of the FLC include traffic data, incident information, and network attributes; outputs from the FLC include control settings, i.e. recommended ramp flows.

3.2.2 Traffic control procedure in incident events

Figure 3.2 outlines the process of traffic control during incidents using FLC. The process involves three steps:

Figure 3.2: Traffic control procedure during incidents

(i) Evaluation of the current incident traffic congestion

A traffic stream is characterized by its *state* and the *change in state*. This step involves the evaluation of the state of traffic prevailing at the current time. The purpose is to answer the questions *what is happening*, and *how critical* is the event. To estimate the current traffic conditions, the FLC uses traffic data and incident attributes upstream of the incident location. Depending on the critical level of the congestion, the FLC continues the second step – the prediction of traffic tendency, or proceeds to the third step – recommendation of control actions. Details will be explained in Section 3.4.1.

(ii) The prediction of congestion tendency

This step involves the prediction of the *change in state* of traffic as well as the evolution of the incident problem. Short-term prediction of traffic variables is an important step in evaluating the evolution of incident-related traffic condition, a prerequisite condition for proactive traffic control.

(iii) Recommendation of control strategies and actions

Given the outcomes from the first two steps, the FLC performs a sequential analysis to arrive at recommended solutions. During the control implementation, the traffic surveillance system continually observes and provides updated data and information to the FLC. Since the control input is a function of the system input, the FLC behaves like a closed-loop control system.

3.3 MULTI-STAGE FLC

3.3.1 Overview

A FLC employs a number of state (input) variables to produce control actions. The number of rules in the rule base depends on the number of state variables and the number of fuzzy sets in each variable. As the number of variables increases, the number of rules increases exponentially (Yeh, and Li, 2004; Raju and Zhou, 1993 – in Chen and Parng, 1996). For example, for a system consisting of M state variables each has N fuzzy sets, the maximum number of rules is N^M . Consequently, for complicated applications with a large number of input and output variables, the single-stage fuzzy inference may not be suitable since the rule base becomes too cumbersome to handle effectively. To tackle such complex multivariable control problems, the decision-making process is decomposed into stages as a multi-stage structure: output variables from the previous stage are used as input variables of the current stage, and output variables from the current stage are used as input variables for the next stage. Such variables are called intermediate variables, and a FLC with intermediate variables is known as a multi-stage fuzzy logic controller (MS-FLC).

For a multivariable FLC, the decomposition into several stages is preferable since it reduces the complexity of the problem (Chen and Parng, 1996) and provides a hierarchically organized concept of association and reasoning. By using hierarchical structures, the number of rules only increases linearly with the number of variables (Raju and Zhou, 1993 - in Chen and Parng, 1996), instead of increasing exponentially as in the single stage FLC. Multi-stage FLCs have been introduced in various fields (Chung and Duan, 2000; Yeh and Li, 2004). In traffic signal control, a two-stage FLC has been introduced by Trabia et al. (1999).

The decision-making logic in traffic control during incidents represents a complex multivariable control process, which employs “knowledge” at a low level as fact (data/information) to produce control solutions that require a higher knowledge level through reasoning. As stated in Section 3.2.2, the process is decomposed into three stages, each of which is engaged in specific tasks, but functionally dependent on its previous stage: the second stage uses outputs from the first stage and external intermediate variables as its inputs, and the output from the second stage is used as the input of the third stage. In brief, due to the nature of problem, the traffic control process under incidents needs to be conducted sequentially, from *evaluation* to *anticipation* to *recommendation* so as to achieve a systematic and logical control process. Although the employment of many intermediate variables may not be desirable, the decomposition into several stages helps reduce the problem complexity and the number of rules, therefore the reasoning time can be reduced, and the overall performance enhanced significantly.

3.3.2 Overall rule base architecture

In general, the rule base in a MS-FLC consists of several stages. Each stage has a number of rule blocks and each block contains a number of rules. Figure 3.3 illustrates the generic hierarchical structure of a MS-FLC in which the rule base is decomposed into N stages, each stage has M blocks, and each block has O rules.

Figure 3.3: Hierarchical architecture of a MS-FLC

Formation of individual rules

The antecedent of a rule may have several conditions, connected by logical operators *AND*, *OR*, or *NOT*. The compositional inference is implemented using compositional operators such as *MAX-MIN* and *MAX-PROD*. The rule is classified as *homogeneous* if the logical operators are of the same type, and *non-homogeneous* if the logical operators are of different types.

The homogeneous rule type is more frequently used in the rule compositional operation, especially those rules that are connected with logical *AND* in the antecedent part. For example, the following rule includes three variables congestion_level, time_of_day, and predicted_traffic_condition connected with *AND* operator to produce the control_level:

$R_{\text{hom}o}$: If congestion_level is light AND time_of_day is off_peak AND predicted_traffic_condition is downtrend THEN control_level is low.

The rule can be generalized linguistically as:

$$\text{if } A_i \text{ is } P_{i,j} \text{ AND } A_k \text{ is } P_{k,l} \text{ AND...} A_m \text{ is } P_{m,n} \text{ then } Y_p \text{ is } P_{p,x} \quad (3.1)$$

It can also be restated with symbolical expression as follows:

$$(A_i \bullet P_{i,j}) \Theta_{AND} (A_k \bullet P_{k,l}) \dots \Theta_{AND} (A_m \bullet P_{m,n}) \rightarrow Y_p \bullet P_{p,x} \quad (3.2)$$

where A_i denotes the input variable i , and $P_{i,j}$ denotes predicate j in its corresponding variable i . The symbol $\bullet$ denotes the word *is*, and symbol Θ denotes a logic operator that can be either *AND* or *OR* or *NOT*.

Figure 3.4a describes a graphical expression of a homogeneous rule with the *AND* operation.

a) Homogeneous rules with the AND operation

b) Non-homogeneous rules with *AND* and *OR* operations.

Figure 3.4: Formation of rules.

In non-homogeneous rules the logical operators are of different types. For example, the following rule uses both *AND* and *OR* operators in the antecedent:

$$R_{non-homo} :$$

- *If congestion_level is heavy AND (time_of_day is peak OR predicted_traffic_condition is uptrend) THEN control_level is high.*

The rule is equivalent to two homogeneous rules:

- *If congestion_level is heavy AND time_of_day is peak THEN control_level is high.*

Also:

- *If congestion_level is heavy AND predicted_traffic_condition is uptrend THEN control_level is high.*

The rule $R_{non-hom o}$ can be generalized linguistically as:

$$\text{if } A_i \text{ is } P_{i,j} \text{ AND } A_k \text{ is } P_{k,l} \text{ OR...} A_m \text{ is } P_{m,n} \text{ then } Y_p \text{ is } P_{p,x} \quad (3.3)$$

or is restated with symbolical expression:

$$(A_i \bullet P_{i,j}) \Theta_{AND} (A_k \bullet P_{k,l}) \dots \Theta_{OR} (A_m \bullet P_{m,n}) \rightarrow Y_p \bullet P_{p,x} \quad (3.4)$$

A graphical expression of a non-homogeneous rule with *AND* and *OR* operations is presented in Figure 3.4b.

Concept of rule block

A rule block is a set of rules that employs one or more inputs to yield a single output. A rule block is one of the key components of a rule base. It acts as a relatively independent unit in the rule base since rules in one block have no relation with rules in other blocks. Figure 3.5 illustrates two rule blocks. Rule block 1 uses two input variables (speed and density) to supply congestion level. Rule block 2 uses the queue length and congestion level provided by rule block 1 to yield congestion status.

Figure 3.5: An example of rule block

3.3.3 Rule block architectures

Berkan and Trubatch (1997) introduced several types of rule block formation, applicable for both single and multi-stage FLC architectures. In the single-stage parallel structure (Figure 3.6), each of the rule blocks uses a set of input variables (X_i) to produce a specific output (Y_i). The outputs are independent from each others.

Figure 3.6: Single-stage parallel structure

The second type of rule block formation is the multi-stage cascade structure (Figure 3.7). In this structure, the fuzzy rule inference system is decomposed into several stages, with each of them fulfilling specific functionalities. An immediate stage receives output from the previous stage to produce output for the following stage. One stage typically consists of several rule blocks, and the rule integration may follow the parallel or composite style: outputs from rule blocks may be aggregated or used straightforwardly to deliver inputs for the next stage.

Figure 3.7: Multi-stage cascade structure

The third rule block architecture is the multi-stage composite structure (Figure 3.8). This architecture is similar to the multi-stage cascade style, except that rule blocks in intermediate stages may employ both internal and external inputs (i.e Block 2, m). The output from a rule block may be used directly or aggregated before being delivered to the next stage.

Figure 3.8: Multi-stage composite structure

3.4 PROPOSED MS-FLC

3.4.1 Conceptual model of the MS-FLC

Figure 3.9 describes the proposed architecture of the FLC for incident management. The model reflects a complex sequential structure of the decision-

making logic for the multivariable traffic control problem. The rule base of the MS-FLC consists of 3 stages, each of which deals with specific functionalities as described in Figure 3.2. The rules in the first stage need to be executed first to give results to the second stage that receives the output from the first stage as its internal input, and external inputs from traffic forecasting. Similarly, the third stage employs both internal and external inputs to produce output in the form of control actions.

Figure 3.9: Conceptual model of FLC for incident-related traffic control

Stage 1: Evaluation of incident traffic congestion

The objective of this stage is to evaluate the current *state* of traffic in incident occurrence. Proper evaluation of the current state of traffic is important in understanding *how severe* the traffic congestion is, on this basis reasonable advices

in the next step can be made. The state of traffic is prescribed by three principal quantities: congestion *level*, congestion *mobility*, and congestion *status*: congestion level reflects the severity of traffic, estimated by traffic speed and density; congestion mobility determines the dynamics of the congestion, quantifiable by traffic speed; and the congestion status refers to the existence and magnitude of the queue length on the expressway. The congestion mobility and congestion status blocks deal specifically with the heavy congestion category. Each component (rule block) requires various treatments in the subsequent stages. If the congestion problem is critical, urgent control interventions need to be implemented immediately, and the corresponding rules in stage 3 are executed. By contrast, if the traffic congestion is not yet critical, the system proceeds with traffic forecasting module and rules in the second stage will be fired. The rules in this stage can be categorized as *fact-state* rules since the reasoning logic uses numerical data to estimate the state of traffic.

Stage 2: Prediction of incident traffic conditions

Given the outcome from the first stage, the second stage continues to anticipate the traffic and incident conditions in the immediate future, which is typically 5, 10, 15 minutes, known as short-term traffic prediction. The prediction of short-term traffic conditions is crucial to ensuring success in any proactive-traffic control scheme. In anticipation of traffic and incident conditions, the essence is the prediction of short-term traffic and incident variables. This task involves the employment of the new and advanced traffic forecasting technique based on SVM, which will be introduced in Chapter 6. The rationale behind the integration of fuzzy logic and SVM technique relies on the fact that: although both techniques stem from Artificial Intelligence, they represent different approaches in machine learning (Kecman, 2001): the black-box approach (SVM) and the white-box approach (Fuzzy logic). Although the SVM approach has very powerful learning capability, it has no ability to interpret the results they produce. They are not equipped with explanatory tool to speak out by themselves using linguistic expressions. By contrast, although fuzzy logic systems are efficient in handling problems of approximate reasoning and explanation to interpret the results clearly, their learning capability is limited.

Therefore, the integration of the two machine learning techniques may be significant in the sense that it allows inheriting the advantages from one component to compensate for shortcomings of the other. More concisely, SVM is used to support the fuzzy KBS, whose expressive power is strong but learning capability is weak. With this integration, the fuzzy KBS is expected to achieve higher performances.

In the KBS, SVM is used specifically for predictions of traffic variables, and fuzzy logic is used for data pre-processing and reasoning the results supplied by SVM: the predicted data from SVM will be adjusted by a risk factor that takes account of unexpected and unknown parameters (those cannot be considered in the SVM prediction), including the feedback from the control system being implemented (the dotted line in Figure 3.9). The rules in this stage are typically *state-state* rules, since the reasoning sequence infers the future state from the current state using external variables from the traffic-forecasting module.

Stage 3: Recommendation of control measures

The outputs from stages 1 and 2 will firstly be used to determine the strength of necessary control intervention, upon which an appropriate control strategy is recommended. The control strategy rule block brings forward a broad view of alternative control solutions based on the estimated control intervention and the availability of control facilities. The traffic operator may consider isolated, coordinated, or integrated control strategy (as reported in Chapter 2). Once a control strategy is selected, concrete control actions are translated. The outputs of the FLC system should be defuzzified to deliver crisp values. Given this reasoning process, the rules in stage 3 pertain to both strategic level (for intervention level, control strategy) and operational level (for control settings). The rules for control actions are basically *state-action* rules for the given input-output mapping.

3.4.2 Rule base structure of the proposed MS-FLC

Given the prescribed functionalities and relationships, the rules in the proposed MS-FLC can be expressed in the general form:

$$\mathbf{Y} = \mathbf{f}(\mathbf{X}, \mathbf{U}) \quad (3.5)$$

where $\mathbf{X}$ is the vector of input variables, $\mathbf{U}$ is the vector of intermediate variables, and $\mathbf{Y}$ is the vector of output variables.

$$\mathbf{X} = (x_1, x_2, \dots, x_n)^T \quad (3.6)$$

$$\mathbf{U} = (u_1, u_2, \dots, u_m)^T \quad (3.7)$$

$$\mathbf{Y} = (y_1, y_2, \dots, y_n)^T \quad (3.8)$$

$$\text{where } y_i = f_i(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m); \forall i = 1, \dots, n \quad (3.9)$$

$$u_j = \psi_j(x_1, x_2, \dots, x_n); \forall j = 1, \dots, m \quad (3.10)$$

Equations (3.6) to (3.10) represent non-linear relationships of a fuzzy multivariable control model. In the MS-FLC, the primary parameters of input variables are employed in the first stage, while in the second and the third stages both intermediate inputs from the first stage as well as external variables are utilized. Basically, the rules in this MS-FLC have MISO structure, where multiple inputs are used to produce a single output. Given these, the formation of rules in three stages can be described as follows:

$$\text{Stage 1: } \left\{ \begin{array}{l} R_1 : \text{If } X_1 \text{ is } A_{1,x}^1 \cap \dots \cap X_N^1 \text{ is } A_{1,x}^1 \text{ then } Y_1 \text{ is } C_{1,y}^1 \\ \dots \\ R_{n_1} : \text{If } X_1 \text{ is } A_{n_1,x}^1 \cap \dots \cap X_N^1 \text{ is } A_{n_1,x}^1 \text{ then } Y_1 \text{ is } C_{n_1,y}^1 \end{array} \right\} \Rightarrow \text{to the 2}^{\text{nd}} \text{ stage} \quad (3.11)$$

$$\text{Stage 2: } \left\{ \begin{array}{l} R_1 : \text{If } Y_1 \text{ is } A_{1,x}^2 \cap \dots \cap X_E^2 \text{ is } A_{1,x}^2 \text{ then } Y_2 \text{ is } C_{1,y}^2 \\ \dots \\ R_{n_2} : \text{If } Y_1 \text{ is } A_{n_2,x}^2 \cap \dots \cap X_E^2 \text{ is } A_{n_2,x}^2 \text{ then } Y_2 \text{ is } C_{n_2,y}^2 \end{array} \right\} \Rightarrow \text{to the 3}^{\text{rd}} \text{ stage} \quad (3.12)$$

$$\text{Stage 3: } \left\{ \begin{array}{l} R_1 : \text{If } Y_2 \text{ is } A_{1,x}^3 \cap \dots \cap X_E^3 \text{ is } A_{1,x}^3 \text{ then } Y_3 \text{ is } C_{1,y}^3 \\ \dots \\ R_{n_3} : \text{If } Y_2 \text{ is } A_{n_3,x}^3 \cap \dots \cap X_E^3 \text{ is } A_{n_3,x}^3 \text{ then } Y_3 \text{ is } C_{n_3,y}^3 \end{array} \right\} \Rightarrow \text{defuzzification}$$

(3.13)

where

$X_{(.)}$: input variables

$Y_{(.)}$: output variables

 $A_{j,x}^i$: fuzzy number in the antecedent part

$i = 1, 2, 3$: the stage

j : the rule j^{th} in each stage

$j = 1, 2, \dots, n_1$, where n_1 is the number of rules in stage 1

$j = 1, 2, \dots, n_2$, where n_2 is the number of rules in stage 2

$j = 1, 2, \dots, n_3$, where n_3 is the number of rules in stage 3

$x = 1, 2, \dots, M$ indicates *any fuzzy number* in the antecedent fuzzy sets; M is the number of fuzzy sets in each input variable

N : the number of input variables employed by the 1st stage

E : the number of external input variables employed by stages 2 and 3, beside the input variables produced from the previous stage (Figure 3.8).

$C_{i,y}^i$: fuzzy number in the conclusion part

$y = 1, 2, \dots, O$ indicates *any fuzzy number* in the conclusion fuzzy sets; O is the number of fuzzy sets in each output variable.

The rules in the three stages above can be restated symbolically as follows:

$$\text{Stage } 1: \left\{ \begin{array}{l} R_1 : X_1 \bullet A_{1,x}^1 \ominus ... \ominus X_N^1 \bullet A_{1,x}^1 \rightarrow Y_1 \bullet C_{1,y}^1 \\ \\ R_{n_l} : X_1 \bullet A_{n_l,x}^1 \ominus ... \ominus X_N^1 \bullet A_{n_l,x}^1 \rightarrow Y_1 \bullet C_{n_l,y}^1 \end{array} \right\} \quad (3.14)$$

$$\text{Stage 2: } \left\{ \begin{array}{l} R_1 : Y_1 \bullet A_{1,x}^2 \ominus \dots \ominus X_E^2 \bullet A_{1,x}^2 \rightarrow Y_2 \bullet C_{1,y}^2 \\ \dots \\ R_{n_2} : Y_1 \bullet A_{n_2,x}^2 \ominus \dots \ominus X_E^2 \bullet A_{n_2,x}^2 \rightarrow Y_2 \bullet C_{n_2,y}^2 \end{array} \right\} \quad (3.15)$$

$$\text{Stage 3: } \left\{ \begin{array}{l} R_1 : Y_2 \bullet A_{1,x}^3 \ominus \dots \ominus X_E^3 \bullet A_{1,x}^3 \rightarrow Y_3 \bullet C_{1,y}^3 \\ \dots \\ R_{n_3} : Y_2 \bullet A_{n_3,x}^3 \ominus \dots \ominus X_E^3 \bullet A_{n_3,x}^3 \rightarrow Y_3 \bullet C_{n_3,y}^3 \end{array} \right\} \quad (3.16)$$

Note that in Equations (3.11) to (3.16) the rules are assumed homogeneous using the AND operator for simplicity. As will be seen in the following chapters, in this MS-FLC the AND operator is predominant in the compositional operation, even though the OR operator are occasionally used. The rule structure of the three stages represented by Equations (3.14), (3.15), and (3.16) will be elaborated in Chapters 5, 6, and 7 respectively.

CHAPTER 4 DATA COLLECTION AND ANALYSIS

4.1 INTRODUCTION

4.1.1 The expressway network in Singapore

The expressway network in Singapore (Figure 4.1) consists of eight expressways with a total length of 150 km, among which Pan Island Expressway (PIE) is the longest expressway with the length of 44 km, followed by Ayer Rajah Expressway (AYE) with the length of 26 km (Table 4.1). Most of the expressway sections have 3 lanes in each direction, but an additional lane is sometimes provided at sections with on/off-ramps to accommodate demand changes. Being the backbone of the land transport system in a densely populated urban island, the expressway network is characterized by a high frequency of access from urban arterials.

Figure 4.1: The expressway network in Singapore

Most of sections of Bukit Timah Expressway, Kranji Expressway, Seletar Expressway, and Tampines Expressway can handle traffic in good operating

conditions, being operational conditions within the traffic stream that are qualitatively indicated by smooth traffic movement in which levels of service A, B, and C (free flow and stable flow) prevail. However, heavy traffic is frequently observed on Central Expressway, Pan Island Expressway, and East Coast Expressway, with average daily demand of more than 100,000 vehicles per day (*vpd*) in one direction, while demands of up to 150,000 *vpd* are sometimes observed. High demands are probably the major cause of traffic breakdowns and incident occurrences on these expressways.

Table 4.1: Expressways in Singapore

(Source: MOT, 2002)

No.	Expressway	Abbreviation	Length (km)
1	Ayer Rajar Expressway	(AYE)	26
2	Bukit Timah Expressway	(BKE)	11
3	Central Expressway	(CTE)	16
4	East Coast Expressway	(ECP)	19
5	Kranji Expressway	(KJE)	8
6	Pan Island Expressway	(PIE)	44
7	Seletar Expressway	(SLE)	12
8	Tampines Expressway (TPE)	(TPE)	14
	Total		150

The PIE is one of the most congested expressways. An estimation of delay is made for each expressway based on the HDB data for the first 6 months of the year 2003. The delay is defined as the additional vehicles' travel time with reference to free-flow travel times on each section. Delay (D_t) of a road segment of n links in time interval t is calculated by the formulae:

$$D_t = \sum_{i=1}^n V_{it} * L_i \left(\frac{1}{v_{it}} - \frac{1}{v_{if}} \right) \quad (4.1)$$

where: V_{it} denotes traffic volume at station on link i during time t ; L_i denotes the length of link i ; v_{it} denotes the average speed at station on link i during time t , and v_{if} denotes the free-flow speed of link i .

The free-flow speed of link i , v_{if} , was obtained from regression analyses using data in the HDB under free-flow conditions (low volume, high speed).

The result from the delay estimation showed that PIE accounted for 31.7% of the total delay of the expressway network (Figure 4.2), followed by CTE and AYE.

Figure 4.2: Delay on expressways (Jan.-Jun., 2003)

Shockwave propagation can frequently be observed at bottlenecks. The congestion is occasionally aggravated by the combined effect of both recurring and non-recurring congestion. Figure 4.3 presents an example of the doubled effect of the two types of congestion on a studied stretch of segments on PIE (Figure 4.4): A recurring congestion occurred at 4.20 PM at section 80007762 and propagated upstream (shockwave 1). While the recurring congestion occurred, an accident took place downstream at section 80007770. The accident congestion moved upstream

(shockwave 2) and aggravated the recurring congestion, making the congestion more severe and long lasting over an extended expressway section.

Figure 4.3: Density contour map of a doubled congestion

1: recurring; 2: non-recurring

(Segment 80007758-80007774, Nov 3rd, 2003)

Figure 4.4: A studied stretch of segments on PIE

4.1.2 ITS systems in Singapore

The expressways are monitored under the Expressway Monitoring and Advisory System (EMAS) that provides travellers with real-time traffic and incident information to facilitate smooth movement of traffic. For traffic management purpose, expressways are divided into segments of lengths ranging from 0.9-1.7 km, each covered by several video-based stations, uniquely recognised by its identity. An online interactive map (Figure 4.5) is available on the web to provide travellers with real-time information on expressways.

Figure 4.5: Interactive map for expressway system

(Source: <http://www.onemotoring.com.sg/>; Captured time: 10/07/2005 11:07 AM)

The data used in this study are obtained from the Land Transport Authority (LTA) over a 12-month period. There are two major types of data: traffic data and incident information. Traffic data include traffic flow rate (vehicles/h) and speed (km/h). Attributes conveyed by incident information include the ID, type, location, date, start time, and end time of each incident. In order to facilitate data distribution, since July 2004 the LTA has deployed Extensible Markup Language (XML) for describing and delivering data through the network. The real-time data, updated

every 5 minutes, include traffic data, traffic news, incident information and image files from different camera locations (Figure 4.6). Through traffic sensors and surveillance cameras, the operators obtain necessary real-time data and images for monitoring the system.

a) PIE: Mount Pleasant

b) PIE: Kalang

(Captured time: 10/04/2004 07:18 AM)

(Captured time: 10/07/2005 11:07 AM)

Figure 4.6: Snapshots of camera views on PIE

4.1.3 Issues in the LTA's data

Traffic data

Although a large amount of data can be retrieved, some issues are encountered in the LTA's data that requires further data processing for applications. Issues in the LTA's traffic data include:

- i. The EMAS system uses video-camera technology for data acquisition. Such technology does not allow measurement over a length of road, but is measurement over a short section. Short section measurements provide data on volumes and speeds (time mean speeds), but no data on density is provided. Instead, density must be derived from traffic flow rate and space mean speed, which is approximated from the measured time mean speed, as will be described in Section 5.5.3;

- ii. LTA's traffic data are aggregated across all traffic lanes. Therefore, for studies that require lane-specific measurements, additional field surveys need to be carried out;
- iii. With 5-minute data aggregation, LTA's traffic data are rather coarse for traffic flow forecast and for real-time traffic control, especially under incident conditions. Consequently, in applications that require high-data resolution such as the evaluation of incident conditions (Chapter 6), simulated data must be used. The impacts of data resolution on the accuracy of prediction models will be discussed in Sections 6.3.2, 6.3.6, and 6.3.8.

Incident data

Three major issues are encountered in the incident data reported in the LTA's database:

- i. Incident "data" are actually real-time information described in the text format, which is difficult for data handling, analysing, and linking with traffic data. A programme (Appendix A) is coded to transform the data from the text to tabular format for the data manipulation;
- ii. Incident data convey all types of "incidents", including expected events (Roadwork) and traffic congestion (Heavy traffic) without stating whether the congestion is recurring or non-recurring. Since this study focuses on non-recurring congestion management, only genuine incidents are of concern. Therefore, mapping between database objects (Section 4.2.2) is required to investigate if a congestion under consideration is recurring or non-recurring in nature;
- iii. Incident data do not provide actual start and end times of incidents. Instead, the incident start time is the time reported by the incident detection system, and the incident end time is the time when the incident is removed from the site, not the time traffic is restored to normal conditions.

A common method to determine the (reported) start time of an incident relies on the timestamp at that the earliest traffic disturbance at the nearest upstream detector is observed by graphical representation. Dia (1996) assumed that the "threshold" of

this disturbance is 20% changes in traffic parameters. Since it takes time for the incident-inducing disturbance to propagate upstream, there is a lag from when the incident actually occurs to the time it is detected/reported. Mak (2002) assumed that the actual incident start time is defined to be one time interval (one minute) earlier than the time when disturbance of traffic parameters is observed at the adjacent detector station. This assumption reflects the time required for the shockwave to travel from the incident location to the nearest detector station.

There are a number of issues associated with the visual determination method:

- Not all incidents induce shockwave propagation upstream. From the traffic engineering perspective, only incidents with significant capacity reduction that leads to an unbalance between the remaining road capacity and traffic demand $\left(\frac{V}{C^*} \geq 1 \right)$ would result in this phenomenon. There is a significant proportion of incidents that do not create significant capacity reduction, such as minor incidents happen during off-peak periods or nighttimes;
- The visual determination of incident start time is labour intensive and requires skill and experience, and may result in inaccurate and inconsistent results (Mak, 2002) due to the fact that this time lag depends on many factors, including traffic variables, incident severity, the distance from the incident location to the upstream detector, and the pre-incident traffic condition;
- Although it may be possible to estimate the range of time needed for the incident-inducing shockwave to propagate to the upstream detector, the time derived from this method may not necessarily reflect the true incident start time because of the complex interactions between various flow conditions, incident location and the distance to the nearest upstream detector. In many cases, traffic disturbances identified in the database are the results of unknown reasons, not necessarily of the incident being investigated. Therefore, it is not possible to determine the actual start time for each

incident by merely examining the traffic and incident databases (Mak, 2002).

In addition to the incident start time, the actual end time of the incident may be earlier or later than the reported end time. The actual end time could be earlier than the reported end time if the incident is temporarily moved to the shoulder lane to restore normal traffic before the incident is actually removed from the site. In contrast, the actual end time may be later than the reported end time in case the incident is physically removed from the site, but it takes time for the incident congestion to dissipate.

Given the issues in the determination of incident start time and incident end time, it is hardly that the LTA's incident data provide true durations of incidents that have occurred in Singapore expressway network. Because it is not possible to ascertain the true duration for every incident, in the subsequent parts of this Chapter, in particular Section 4.3.1, the incident durations provided by the LTA's incident data are only used to illustrate the concept of parametric learning in this experimental model.

4.2 DATABASE DEVELOPMENT

4.2.1 Overview

In this study, data are captured into two relational databases, including the HDB and the RTDB. The HDB stores traffic data, incident information and network attributes. In brief, the key functions of the HDB include:

- It provides a data storage, from which patterns on traffic congestion problems and rules can be learnt. Time-series profiles derived from the HDB can be used to identify irregular patterns in actual real-time situations.
- It provides data to identify incident-sensitive locations. Detailed analysis behind traffic and incident problems on these locations should be carried out in developing traffic management plans.

- Statistical averages in the HDB can be used as a proxy for real-time data in case of missing or contaminated data.
- Data in the HDB can be used in conjunction with real-time data for short-term traffic prediction. Particularly, the historical data set is used to locate similar patterns for the SVM training.
- Data in the HDB can be used in simulation models to evaluate traffic control schemes.

The HDB has data collected since October 2002 and hosts millions of records. Having been normalized and cleaned to remove contaminated data (see Appendix A), the HDB is used for the above-mentioned tasks.

Like the HDB, Traffic and Incident tables are key building blocks of the RTDB. Unlike the HDB that collects and stores data over a long period for qualified quantitative parameters, the RTDB collects and maintains only relevant real-time data. Programs (Appendices A and B) were coded to establish the connection to the LTA databases. The programs offer the RTDB a capability to move forward with time and automatically dump outdated records whose timestamps lie beyond a pre-defined rolling horizon. The programs are able to accomplish transactions within a stringent time constraint by which a new transaction is about to be executed.

4.2.2 Relationship between database objects

The HDB stores and archives traffic data, incident information and network attributes in the form of relational tables. The functional dependency of relational databases allows mapping common attributes in different tables. For example, in Figure 4.7 the Traffic and Incident tables work collectively by mapping different records with common attributes in the Network table following different types of relationships, using Structural Query Language (SQL).

Select {set of selected attributes} from (databases' tables) where {criteria};

The mapping between the Traffic and Incident Tables allow associating the traffic data with incident data for the subsequent investigation and analyses of incidents in the HDB.

The figure shows three overlapping database tables from the HDB system. The 'Traffic : Table' window is at the top, showing columns: roadID, speed, vol, date, time. The 'Incident : Table' window is in the middle, showing columns: incidentID, linkID, startDate, startTime, type, endDate, endTime. The 'Network : Table' window is at the bottom, showing columns: roadID, linkID, upStr, downStr, from-to, laneI, cap, bNeck. Red lines and numbered circles (1-4) indicate specific relationships between fields in different tables.

roadID	speed	vol	date	time
80007758	73	3281	10/4/2004	9:05:00
80007762	57	2593	10/4/2004	9:05:00

incidentID	linkID	startDate	startTime	type	endDate	endTime
EM154770	10029386	10/4/2004	8:45:55 AM	V	10/4/2004	9:25:56 AM
EM154771	10094063	10/4/2004	8:47:00 AM	A	10/4/2004	9:15:25 AM

roadID	linkID	upStr	downStr	from-to	laneI	cap	bNeck
80007760	10029138	10043942	10029142	from JALAN F	4	8993	0
80007762	10036770	10029138	10036770	from JALAN F	4	8993	0
80007762	10090463	10036770	10029150	from JALAN F	4	8993	1
80007766	10029150	10090463	10029243	from JALAN F	4	8993	0

Figure 4.7: Relationships among tables in HDB

(1): one-to-many relationship: matching roadIDs between the Traffic and Network tables.

(2) and (3): one-to-one relationships: matching in linkIDs and dates, respectively.

(4): The report time of traffic data lies between incident start time and end time.

4.3 DATA ANALYSIS

The data analysis explores the available data and information to learn the intrinsic nature of traffic and incident problems in the expressway network. Through the data analysis, typical traffic patterns and their evolution under normal and incident conditions can be identified, and the relationships among traffic variables can be established. In the knowledge acquisition process, data analysis is the preliminary step for subsequent parametric learning of membership functions and rules in the fuzzy rule base.

Techniques employed in the data analysis are primarily statistical analysis with the aid of SQL. An example associated with the recognition of traffic patterns: a traffic

pattern can be represented by statistical parameters such as *means*, whereas the deviation of traffic flow is characterized by *variances*, or *coefficients of variation*. The representation of traffic features by time-series patterns consolidates the huge amount of data, given the fact that traffic patterns tend to exhibit a cyclical phenomenon. In addition, the knowledge on traffic pattern helps to identify distinction between recurrent and non-recurrent congestion, using pattern recognition.

4.3.1 Traffic patterns

Typical volume patterns can be established for various road segments on different weekdays and weekends. Figure 4.8 plots the patterns on two consecutive segments, 80007762 and 80007766, along PIE (see Figure 4.4). It can be observed that traffic patterns on the two segments are similar, particularly on working days.

A correlation analysis on traffic volumes for segment 80007762 is conducted to see how strongly various traffic patterns correlate. Traffic volumes in one day constitute a time-series pattern from 288 consecutive intervals. The correlation coefficient ξ between two random patterns X and Y is statistically computed as:

$$\xi = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X) \text{var}(Y)}} \quad (4.2)$$

where $\text{cov}(X, Y) = E(XY) - E(X)E(Y)$, being the covariance of two random patterns X and Y ; $\text{Var}(X)$ and $\text{Var}(Y)$ denote the variances of patterns X and Y , respectively.

The correlation coefficient ξ ranges between -1 to 1. Values close to 1 indicate that the two patterns X and Y are highly and positively correlated. Values close to 0 signify that the two patterns are highly uncorrelated. At the other extreme, values close to -1 imply that they are highly and negatively correlated.

a) Segment 80007762

b) Segment 80007766

Figure 4.8: Volume patterns on two consecutive segments on PIE

Table 4.2 lists the average correlation coefficients of traffic volumes on segment 80007762 for two cases: "within day of week" and "between days of week ". The former reflects the correlation between a day and the other days (the same weekday) in the analysis, for example between one Monday with the other Mondays in the analysed data, while the latter indicates correlation between a day and the other days in the week (different weekdays). The results show that the correlation coefficient values are pretty high in the case of *within-days-of-week*, from 0.93 to 0.95. In the second case *between-days-of-week*, the values range from 0.90 to 0.94 for weekday-weekday relationships, and from 0.76 to 0.77 for weekday-weekend relationship. Note that the data in the correlation analysis (vectors X and Y , Formulae 4.1) are mutually exclusive, thus the coefficients in the diagonal do not equal exactly 1.

Table 4.2: Correlation analysis of time-series traffic volume data

($i, j = 1 \div n$, $i \neq j$, where n is the number of weeks in the analysis period)

Date	Date i							
Date j	Day of week	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.
	Mon.	0.9348						
	Tue.	0.9316	0.9475					
	Wed.	0.9356	0.9396	0.9427				
	Thu.	0.9367	0.9389	0.9345	0.9443			
	Fri.	0.9378	0.9385	0.8988	0.9406	0.9368		
	Sat.	0.8207	0.8389	0.8174	0.8537	0.8678	0.9301	
	Sun.	0.7694	0.7746	0.7665	0.7579	0.7957	0.7865	0.9305

These high degrees of correlation of traffic patterns signify that historical patterns, to some extent, can be used as a baseline for prediction of traffic flow in case real-time data is missing or contaminated. Nevertheless, the creditability of historical data for short-term traffic prediction should be subjected to further investigation, as will be presented in Chapter 6.

Figure 4.9 illustrates the speed-flow relationship for segment 80007766. The figure shows common characteristics of speed-flow relationships in conventional traffic

engineering practices: the data seem to populate in clusters, associating with three distinct regions: region A represents un-congested traffic, region B exhibits transitional state, and region C reflects congested conditions. The clustering features can be explored to determine input-output mapping using Fuzzy C-Means clustering method, as introduced in Section 4.4.2.

Figure 4.9: Speed-flow relationship

Figure 4.10 presents an analysis of traffic conditions based on almost 60,000 speed records on the above segment, using the concept of congestion index described in Ishak and Al-Deek (2002): traffic conditions are defined by classifying the speed range into congestion indices (CI), each of which corresponds to 16 km/h. In this way, the speed range is divided into 8 congestion indices from 0 to 7, where 0 indicates very severe congestion (speed range from 0 to 16 km/h), and 7 indicates the maximum speed range (speeds above 112 km/h).

The figure shows that the distributions of speeds are normal-like, with the mean significantly shifted to the left under incident conditions.

Figure 4.10: Distribution of congestion indices on segment 80007766

4.3.2 Incident problems

Table 4.3 lists the number of incidents reported in the LTA database from November 2002 to November 2003 and the corresponding incident durations for various types. Vehicle breakdown is the most frequent type of incident, accounting for more than half of all incident occurrences, followed by accident and roadwork.

Table 4.3: Number of incidents from November 2002 to November 2003

Incident type	Number of incident		Duration	
	Number	Proportion (%)	Total duration (min.)	Proportion (%)
Accident	1093	22.28	63268	28.98
Vehicle breakdown	2643	53.87	66938	30.66
Roadwork	926	18.87	83486	38.24
Others	244	4.97	4643	2.13
Total	4906	100	218335	100

Although classified as expected events that require pre-planned traffic management scheme, roadwork is reported in the LTA's database as "incident" and account for a high proportion of the incidents, representing one of the major disturbances to traffic, hence it is included in the data analysis to see the differences between *genuine* incidents with *planned* events, especially with respect to the distribution of incident duration and the time of incident occurrences. In Table 4.3, *others* refer to miscellaneous disturbances, unspecified in the LTA's database.

Table 4.4 summarises the key statistics of incident durations of various types, from the incident data with 5-min aggregation. The data shows that for all incident types the distributions of the durations are not symmetrical but positively skewed, especially for vehicle breakdowns. The high ratios of the standard deviations and the means indicate that incident durations fluctuate widely. It should be noted that incident durations in this chapter are calculated from the start and end times of incidents reported in the LTA database. Given the issues in incident data discussed in Section 4.1.3, the calculated incident durations are roughly estimated, and they act as a proxy of actual incident durations.

Table 4.4: Statistics of incident duration of various types (min.)

Parameters	Accident	Vehicle breakdown	Roadwork	Others	All types
Mean	57.99	25.33	90.25	19.10	47.63
Median	46.00	18.00	85.00	12.00	33.00
Mode	15.00	15.00	70.00	10.00	15.00
Standard Deviation	45.17	23.07	56.73	23.20	43.27
Skewness	0.90	2.50	0.40	3.38	1.47
Minimum	1.00	1.00	5.00	1.00	1.00
Maximum	215	199	216	174	216
Sum	63268	66938	83486	4643	218335
Count	1093	2643	926	244	4906

Table 4.5 shows the proportion of incidents with respect to the time of occurrence. Accidents and vehicle breakdowns occurred primarily during the peaks, while roadwork activities were planned in off-peak periods to mitigate disturbances.

Table 4.5: Time of incident occurrence

Time of occurrence	Accident	Vehicle Breakdown	Roadwork	Others
AM peak	42.92	34.15	16.38	25.71
PM peak	39.89	43.31	6.03	48.71
Off peak	17.19	22.54	77.59	25.58

Figure 4.11 shows the distribution of various incident types among expressways. Vehicle breakdown is prevalent in CTE, PIE, and BKE, while the CTE and PIE are locations with the highest accident occurrences.

Figure 4.11: Distribution of incidents on expressways

4.4 LEARNING PARAMETERS OF MEMBERSHIP FUNCTIONS

As reviewed in Chapter 2, essential issues in designing membership functions include the determination of the universes of discourse, fuzzy partitioning, and fuzzy rules. The following presents the concept of learning these parameters from the traffic and incident data in the HDB.

4.4.1 Universe of discourse

As defined in Chapter 2, the *universe of discourse* of a variable is the collection of all elements constituting that variable. It indicates the active domain in which events occur during the studied period. Proper determination of the universe domain is important before designing the number of rules as well as the input-output mapping. Actual historical data could be a good reference source in the selection of the universe domain. It should be noted that the universe of discourse is determined based on the collected historical data. In the long run, if the active domain of the universe changes, the rules should be adjusted accordingly.

Table 4.4 provides direct estimations of the universes of discourse for incident durations. Let T^A , T^V , T^R , and T^O denote the domains of durations (in minutes) of accident, vehicle breakdown, roadwork, and others, respectively. It follows that:

$$T_{\min}^A = 1; T_{\max}^A = 215$$

$$T_{\min}^V = 1; T_{\max}^V = 199$$

$$T_{\min}^R = 5; T_{\max}^R = 216$$

$$T_{\min}^O = 1; T_{\max}^O = 174$$

However, it would be useful to take a closer look at the frequency distributions to discover the ranges that the events occurred with significant number of times. Figure 4.12 plots the relative frequencies for incidents of various types. As can be seen, the distributions (except for road work) are highly positively skewed, and the

values in the extreme right are statistically insignificant, particularly for vehicle breakdown and others (with skewnesses of 2.50 and 3.38 respectively, as shown in Table 4.4). The formula of skewness is introduced in Equation (6.18).

Figure 4.12: Distributions of incident durations

This feature is better reflected using the survival curves shown in Figure 4.13. A survival curve presents the proportional distribution of a hypothetical incident that has duration greater than or equal to the actual duration t . It appears that the curves for incident types b) and d) approach zero at considerably longer ranges. Using the ad-hoc visual inspection of the curves, the following values are obtained:

$$T_{\max}^A = 210$$

$$T_{\max}^V = 150$$

$$T_{\max}^R = 215$$

$$T_{\max}^O = 112$$

a) Accident

b) Vehicle breakdown

c) Roadwork

d) "Other" types

Figure 4.13: Survival functions of incident durations

4.4.2 Fuzzy partitioning

(i) Fuzzy partitioning using engineering knowledge

With the determination of the universes of discourse in the first layer, the next layer in fuzzy logic system design following the data-driven approach (Figure 2.11) is the design of fuzzy membership functions. As described earlier, important issues in designing membership functions include the determination of the number of fuzzy sets, the locations of centres of membership functions, the line style, and the overlap between fuzzy sets. Since the line style has only local impacts on system behaviour, for simplicity, in the following examples the line styles are chosen as linear and the overlaps are determined so that there are only overlaps between two adjacent fuzzy sets.

The determination of the number of fuzzy sets in a fuzzy variable is called fuzzy partition. Fuzzy partition categorizes a variable domain into a finite number of homogeneous classes as well as defines decision boundaries between fuzzy sets in the product space. For simple problems, categorization can be made by human expertise or by commonsense reasoning, namely membership functions are derived in the absence of data. This method may be sensible since the human brain forms decision-making logics based on likelihoods of categories rather than numerical values (Berkan and Trubatch, 1997). However, as the use of data becomes intensive and the identification of decision boundaries requires sophisticated techniques, categorization should not be made without a reference to actual measurements. In the following, membership function design following the data-intensive approach in combination with expert judgements is presented.

Figure 4.14 illustrates the concept of fuzzy partition that evaluates accident durations using data in the year 2003 (Table 4.4). The durations are categorised into three fuzzy sets *short*, *medium*, and *long*.

$$T^A = \{Short, Medium, Long\} \quad (4.3)$$

Figure 4.14: Probabilistic partitioning of the accident duration

In the data-driven approach, the frequency of occurrence indicates the number of times the accident duration takes place. The categorization using purely subjective reasoning in this case may be argumentative, and available data could be a good reference since it provides the *fact* to support the reasoning. The cumulative frequency curve can be constructed from relative frequencies of occurrence. The decision boundaries in this case include the lower bounds, upper bounds, and the centres of the membership functions, which are determined by assigning a certain threshold percentile value. The corresponding thresholds for the mid points of three fuzzy sets are defined at 15th, 50th, and 85th percentile values of the cumulative frequency, respectively.

$$(a) = T15^{th} = 12 \text{ min.}$$

$$(b) = T50^{th} = 45 \text{ min.}$$

$$(c) = T85^{th} = 107 \text{ min.}$$

The parameters (a), (b), and (c) are data points at which the membership degrees equal 1. Duration ranges smaller than (a) and greater than (c) should receive full strength. We keep in mind that the 15th, 50th, and 85th percentile values are assigned using subjective judgements. This set of choice is not unique in the sense that there are an infinite number of choices from the given data set.

The next step is to determine the overlaps, which can be determined in a number of ways, however most of them are unproven. Particularly, Kosko (1992, in Chen and Wang, 1999) proposed that the overlaps between adjacent membership functions should be approximately 25%. In this study, we propose determining overlaps with a reference to probability: membership grade represents the possibility of occurrence or the subjective degree of belief, and probability represents the weight of evidence. Although possibility and probability represent two different concepts, there are many formal correspondences between them. In this regard, let's assume that overlaps are designed such that the sum of membership grades for adjacent fuzzy sets at any point in the overlapping sections equals one, which is similar to the probabilistic characteristics of occurrence. It follows that the membership grade of a particular fuzzy set approaches zero as that of the adjacent fuzzy set approaches one. In this way, the resulting initial membership functions for accident duration are presented in Figure 4.15.

Figure 4.15: Membership functions for accident duration

If the universe of discourse of accident duration is divided into 5-minute intervals, the partitioned fuzzy sets can be represented as:

$$\mu_{Short}^A = \frac{1}{5} \cup \frac{1}{10} \cup \dots \cup \frac{0.15}{40} \cup \frac{0}{45} \quad (4.4)$$

$$\mu_{Medium}^A = \frac{0}{12} \cup \frac{0.09}{15} \cup \dots \cup \frac{0.03}{105} \cup \frac{0}{107} \quad (4.5)$$

$$\mu_{Long}^A = \frac{0}{45} \cup \frac{0.08}{50} \cup \dots \cup \frac{0.96}{105} \cup \frac{1}{107} \quad (4.6)$$

An implication from Equations (4.4) to (4.6) is that a specific duration in the overlap can be categorised in 2 fuzzy sets. For example, an accident lasting 30 minutes can be classified as *short* with a degree of fulfilment $\mu_{Short}^A(T = 30) = 0.46$ and as *medium* with a degree of fulfilment $\mu_{Medium}^A(T = 30) = 0.54$.

This composite approach in membership function design can be extended to traffic variables. Figure 4.16 plots the distribution of historical speeds on segment 80007762, classified into 5-km/h intervals. The values on the horizontal axis are the mid-point of intervals. The fuzzy linguistic predicates are initiated as:

$$V_1 = \{Low, Medium, High\} \quad (4.7)$$

Figure 4.16: Partitioning of speed (low resolution, three fuzzy sets)

It would be ideally that the determination of decision boundaries is backed up by measured data. For example, it will be easier if data were grouped in clusters. However, in this case there is no indication that the data can be classified into clusters.

In designing horizontal curves, the AASHTO 1984 chooses the 15-percentile speed (V_{15}^{th}) as the hands-off speed that corresponds to the hands-off condition in the curves: For a given radius R and superelevation e , there is one speed at which

drivers can negotiate the curve such that the side friction factor f between the vehicle's tyres and the road surface equals 0. At the hands-off speed no steering is required to generate friction forces. Consequently, 15% of drivers have to steer their vehicles away from the bend of the curve, and 85% of drivers have to steer into the bend.

The reason behind the selection of V_{15}^{th} as a benchmark for low-speed vehicles is not explained, but in highway design the superelevation of a curve is determined so as to ensure safety: the superelevation should be sufficiently high so that vehicles (of high speeds) do not turn away from the curve due to the radial force, but should not be too high so that low-speed vehicles do not slide inward on the road surface (low-speed vehicles have low radial force acting outward the curve, hence they may slide to the inner side of the curve if the superelevation is too high), and the proportion of low-speed vehicles is approximated as 15% in the traffic stream. Aarts and Schagen (2006) also classified those who drive at a speed below V_{15}^{th} as slow drivers, and above V_{85}^{th} as fast drivers.

Similarly, road design standards in many countries select the 85-percentile operating speed (V_{85}^{th}) as the speed limit, being the maximum safe speed at which motorists can drive along a road when conditions are such that the geometric features of the road govern the speed. An implication of this selection is that speeds greater than V_{85}^{th} are considered high speed.

Given these, the speeds V_{15}^{th} , V_{50}^{th} , and V_{85}^{th} are proposed as decision boundaries in the categorization of the speed universe into *Low*, *Medium*, and *High* speed, respectively. The parameters of the membership functions are obtained from Figure 4.16:

$$V_{15}^{th} = 32 \text{ (km/h)}; V_{50}^{th} = 55 \text{ (km/h)}; V_{85}^{th} = 78 \text{ (km/h)}$$

The membership functions are plotted in Figure 4.17.

For the same set of data, the speed domain can be partitioned with higher resolution, depending on the design requirements. For example, the speed variable may also be partitioned into 5 fuzzy sets:

$$V_2 = \{\text{Very_low}, \text{Low}, \text{Medium}, \text{High}, \text{Very_high}\} \quad (4.8)$$

Figure 4.17: Membership functions of speed (three fuzzy sets)

As in the case of three fuzzy sets, there is no indication that the data can be a good reference for the determination of boundaries of the fuzzy terms. The only possible solution is to rely on the likelihood of categories, estimated by commonsense reasoning without extensive mathematical proof.

In the case of 5 fuzzy sets, the decision boundaries are proposed at 10th, 30th, 50th, 70th, and 90th percentile values, respectively, and are obtained from Figure 4.18:

$$V10^{th} = 25(\text{km/h}); V30^{th} = 44(\text{km/h}); V50^{th} = 55(\text{km/h});$$

$$V70^{th} = 67(\text{km/h}); V90^{th} = 83(\text{km/h}).$$

Figure 4.18: Partitioning of speed (higher resolution, five fuzzy sets)

In the same way, the membership functions for speed are constructed (Figure 4.19).

Figure 4.19: Membership functions of speed (five fuzzy sets)

(ii) Fuzzy partition by clustering method

Grid clustering

Cluster is defined as a collection of data points that have similar semantical properties. The number of clusters determines the number of membership functions in the input and output spaces, hence the number of rules in the rule base. If the number of clusters is small, the number of rules is small, but the partition may not provide adequate representation of the state and control variables. On the other hand, if the number of clusters is too large, the system may overfit and the number of rules increases exponentially, impairing system performance.

The most straightforward way to initialise the membership functions is to equally partition the universe of discourse into grids with a selected number of clusters (Figure 4.20): the centres of membership functions are equally spaced along each variable domain. Given c clusters, the cluster centres of membership functions are determined as follows (Chen and Wang, 1999):

$$o_i = x_{\min} + \frac{x_{\max} - x_{\min}}{c - 1}(i - 1), \quad \forall i \in [1, \dots, c] \quad (4.9)$$

where $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the input/output spaces.

Figure 4.20: Equalized partitioning of data space

The method is simple and justifiable if the data are uniformly distributed. If the data tend to populate into regions, another version of clustering method based on certain similarity measures would be appropriate, such as fuzzy C-Means Clustering.

Fuzzy C-Means Clustering

The fuzzy C-Means (FCM) is one of the most popular fuzzy clustering algorithms. This technique provides a method that populates a multidimensional space into a number of clusters. In FCM each data point belongs to a cluster to some extent specified by a membership degree. The clustering procedure starts with an initial guess for cluster centres, which are usually the mean locations of clusters. Subsequently, FCM assigns every data point a membership grade for each cluster. By iteratively updating the cluster centres and the membership grades for each data point, FCM iteratively moves the cluster centres toward the proper locations within the given data set. This iterative procedure aims at minimizing an objective function that measures the distance from any given data point to a cluster centre weighted by the corresponding membership values.

FCM algorithm partitions a collection of n vectors in a p -dimensional space of real numbers $(\vec{X} = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n\} \in \mathbb{R}^p)$ into c fuzzy groups such that the total weighted sum of squared error within groups is minimized (Chen and Wang, 1999). The objective function for FCM is defined as:

$$J_m = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m d^2(\vec{o}_i, \vec{x}_k) \quad (4.10)$$

subject to:

$$\sum_{i=1}^c u_{ik} = 1; \quad u_{ik} \in [0, 1] \quad (4.11)$$

$$0 < \sum_{k=1}^n u_{ik} < n \quad (4.12)$$

where u_{ik} is the membership value of the k^{th} data point in the i^{th} cluster; $\vec{o}_i$ is the i^{th} cluster centre; $d(\vec{o}_i, \vec{x}_k)$ is the distance between $\vec{o}_i$ and $\vec{x}_k$; and m is the fuzzy exponent, $m \in (1, \infty)$ that controls the amount of fuzziness. Usually $m = 2$ is chosen (Timm et al., 2004).

The objective function J_m is minimized using an iterative algorithm, which alternately optimises the clusters and the membership grades. By differentiating J_m with respect to o_i for fixed u_{ik} and u_{ik} for fixed o_i , the necessary conditions for minimisation of objective function J_m is obtained:

$$o_i = \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m} \quad (4.13)$$

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d^2(o_i, x_k)}{d^2(o_j, x_k)} \right)^{(1/(1-m))}} \quad (4.14)$$

$$\text{where } i \in [1, c] \text{ and } k \in [1, n] \quad (4.15)$$

The procedure is repeated until the rate of change of the objective function value (Equation (4.10)) in successive iterations is less than a pre-specified threshold.

Application of FCM for speed-density spaces

Figures 4.21 to 4.24 illustrate a simple example in the implementation of the FCM algorithm for fuzzy clustering the product space of speed and density for 135 instances ($n = 135, p = 2$). The input data are plotted in Figure 4.21.

Figure 4.21: Input data for FCM

Figure 4.22: FCM clustering for speed-density spaces ($c = 3$)

Figure 4.22 plots the result of clustering for a chosen number of clusters $c = 3$. Three different symbols indicate three clusters with the highest values of membership functions illustrated in Table 4.6. The number of vectors in each cluster are $n_1 = 31$, $n_2 = 56$, $n_3 = 48$, for clusters 1, 2, 3 respectively. The thick and dark symbols indicate the cluster centres.

Table 4.6: Membership grades in each cluster ($c = 3$)

k	u_{1k}	u_{2k}	u_{3k}
1	0.0094	0.9546	0.0359
2	0.0102	0.9504	0.0394
....			
134	0.8805	0.0324	0.0871
135	0.8624	0.0382	0.0995

Figure 4.23 plots the result of fuzzy clustering as the number of clusters increases to $c = 5$. The centres of clusters represent the weighted physical locations of each cluster, whose coordinates are listed in Table 4.7.

Figure 4.23: FCM clustering ($c = 5$)

Table 4.7: FCM cluster centres ($c = 5$)

Centre	Density (k)	Speed (v)
c_1	38.691	64.751
c_2	14.280	87.698
c_3	121.110	14.001
c_4	84.744	25.513
c_5	65.608	49.271

Figure 4.24 compares the profiles of the objective function J_m with $c = 3$ versus $c = 5$. The objective function with $c = 5$ reaches its minimum value after 8 iterations. As compared to $c = 3$, the objective function value in this case is substantially lower. This indicates that the fuzzy partitioning of the product universe $V \times K$ into 5 clusters is more appropriate than into 3 clusters, for the given data set.

Figure 4.24: Objective functions in FCM

4.4.3 Fuzzy rule generation

Adaptive-Network-based Fuzzy Inference System (ANFIS) is a technique for developing FISs using the framework of adaptive neural networks. The technique

provides a learning procedure for membership function parameters by tracking a given input-output data. The goal of an ANFIS system is to generalize a relationship of the following form (Nayaka et al., 2004):

$$Y^m = f(X^n) \quad (4.16)$$

where $X^n = \{x_1, \dots, x_n\}$ is an n -dimensional input vector; $Y^m = \{y_1, \dots, y_m\}$ is an m -dimensional output vector.

Using a given input-output data set, the ANFIS constructs a FIS, whose membership function's parameters are tuned using either a backpropagation algorithm or a hybrid algorithm that combines backpropagation with the least squares method. The computation and tuning parameters of membership functions follow the adaptive learning process, facilitated by a gradient descent vector. For parameters of membership functions on consequent part, ANFIS uses the least-squares method. Once the gradient vector is obtained, an optimisation routine will be applied to adjust the parameters so as to minimize errors between actual and estimated outputs.

The modelling procedure proceeds through training and testing steps, and validated with checking data, which are presented to the trained ANFIS model to see how well the ANFIS model predicts the output. In this way, the checking data can be used to control the possibility that the model overfitting the data, and the membership functions are chosen so as to minimize the checking errors. In general, the model errors for the checking data set tend to decrease as the training takes place up to the point that overfitting begins, and then the model errors for the checking data increase. This type of model works satisfactorily if the training data is a good representation of the data to be modelled. In contrast, if data is contaminated or the training data do not represent the test data well, the validation by checking data is necessary.

ANFIS is, however, much more complex than conventional FISs, and is not available for all fuzzy inference systems. Shortcomings of ANFIS are that the systems only support first or zero order, and produces only single linear out (Figure

4.26). These, to some extent, render the popularity of the approach for rule generation applications.

The following example illustrates a simple application of the ANFIS technique for learning rules from speed-volume data. The training data (Figure 4.25) comprises a random sampling of some 260 records of speed and volume on segment 80007766, while the test data involves an exclusive small set of data on the segment. In the figure, the “Output” implies the flow rate (veh/h), and the horizontal axis specifies the index of each data pair in the training set, namely the row in the input data table.

Figure 4.25: Speed-Volume training data

Figure 4.26 shows a snapshot of the output of the ANFIS system. The state and control variables consist of 5 and 3 fuzzy sets, respectively. The membership functions of the input data are selected as the Gaussian type, while that of the output are of linear type because ANFIS only operates on the Sugeno system.

Figure 4.26: ANFIS membership functions of Surgeno type

(Input denotes traffic speed; output denotes flow rate)

The following rules are generated from the training data:

R_1 : If speed is Very-low then volume is Low

R_2 : If speed is Low then volume is Medium

R_3 : If speed is Medium then volume is High

R_4 : If speed is High then volume is Medium

R_5 : If speed is Very-High then volume is Low.

The rules look simple and sound commonsense-like, in particular rules R_1 , R_2 , and R_3 seem to be not meaningful. Nevertheless, this example illustrates the concept of learning fuzzy rules from data, and for this particular example the above rule set is adequate to represent the input-output relationship. The complexity increases with the number of fuzzy sets and input variables. Note that the example does not reflect the cause-and-effect relationships between speed and volume variables in traffic engineering since the flow rate depends on traffic demand rather than speed. More precisely, it is traffic speed whose magnitude is dependent on traffic volume and the

road capacity, especially in congested conditions. Therefore, the rules merely prescribe the observed mapping between the two variables.

One drawback of the ANFIS technique is that when data are used for mapping, all information embedded in the data will be translated to fuzzy rules regardless of whether the rules make sense (Berkan and Trubatch, 1997). In other words, the rules are automatically generated in the system by input-output mapping. Consequently, the rules are sometimes not significant but should be revised for use.

Figure 4.27 plots the checking data against the FIS output. It can be seen that the training data represents the actual data reasonably well. The training accuracy can be improved by either modifying the set of membership functions to obtain the best choice for modelling the training data, or more data should be selected. Alternatively, the system can be retrained without the checking data if the training data sufficiently represent the system to be modelled.

Figure 4.27: Checking data versus testing data in validation

4.5 SUMMARY

This Chapter presents the preliminary step in the development of the KBS: the construction of the database component and data analysis. The database consists of the HDB and RTDB that store traffic and incident data obtained on the expressway network in Singapore. The HDB provides a data storage to learn traffic patterns and parameters of membership functions, as well as fuzzy rules through the knowledge acquisition process. The HDB data will also be utilised in subsequent chapters for various applications in the KBS development. The RTDB, on the other hand, stores real-time data to provide inputs for the MS-FLC execution.

The Chapter addresses important issues in designing membership functions, including the establishment of the universes of discourse, fuzzy partitioning, and fuzzy rule generation. The universes of discourse of state and control variables are primarily determined by the inspection of the historical data from the HDB, while the number of membership functions is defined by various fuzzy partitioning methods including engineering knowledge, grid clustering, and fuzzy C-Means. Particularly, the fuzzy partitioning using cumulative frequency distributions is first introduced. Although no unique solution is produced, the method reflects the way in which human brain perceives the likelihood of categories.

The concept of fuzzy rule generation that develops FIS systems using adaptive neural networks (ANFIS) has been investigated. The technique provides an advanced learning procedure for membership function parameters by tracking input-output data. The method is particularly applicable when data is rich and the identification of decision thresholds requires sophisticated techniques.

CHAPTER 5 QUANTIFICATION OF INCIDENT

TRAFFIC CONGESTION

5.1 INTRODUCTION

As stated in Chapter 3, attributes that represent traffic conditions include the state of traffic and the change in state. Proper evaluation of state of traffic is important before making any recommendation for traffic control. Figure 5.1 outlines a schematic representation of the first stage of the FLC. The stage consists of three components, known as three blocks: congestion level (CL), congestion mobility (C_Mob) and congestion status (C_Stat). Each of the blocks constitutes a sub-system of MISO type, which employs several state variables to supply a single output. The CL evaluates the current level of traffic congestion based on speed and density; C_Mob estimates the dynamics of traffic stream given the speed, and the C_Stat determines the spatial extent of congestion, given the queue length.

Figure 5.1: Rule base configuration for the first stage

The current traffic congestion in the CL block is quantified into levels such as free flow, light, medium, and heavy congestion. Under free flow, light, and medium congestion, the MS-FLC proceeds to the second stage that forecasts the evolution of the traffic condition. If the congestion is heavy, the rules will be fired for congestion mobility and congestion status in the second and the third rule blocks, respectively, before being delivered to the third stage.

5.2 EVALUATION OF CONGESTION LEVEL

5.2.1 Choice of state and control variables

A review of literature shows that a number of quantifiable measures have been used, in isolation or combination, in an attempt to quantify congestion level, including macroscopic traffic variables (volume, speed, density), travel time, and travel delay. In addition, traffic congestion has also been evaluated using the length of congested segments or the length of queues. Still, the term *congestion level* remains an abstract terminology since the way to evaluate congestion level depends on specific applications. The following section discusses the approach to evaluate congestion level for traffic control.

Speed is an important measure of the quality of expressway service. The use of speed as an indicator of congestion is straightforward and intuitive. It defines the rate of vehicular motion and is simple to measure. As presented in Section 4.3.1, Ishak and Al-Deek (2002) used speed as the only attribute to define *congestion index* (CI) on freeways and urban arterials: the speed domain is decomposed into 8 CIs, each corresponds to a range of 10 mile/h and represents a congestion level. A value of CI = 0 (speeds in the range 0-16 km/h) indicates *very severe* congestion, while CI = 7 (speeds greater than 112 km/h) refers to maximum speed in *free flow* conditions.

Density is another principal measure for characterizing operational conditions on expressways. Density indicates the spatial vehicular concentration, measured by the number of vehicles per unit length of road. Density reflects driver's freedom to manoeuvre and internal friction within a traffic stream. For these reasons, density is used together with speed to define levels of service for basic freeway sections (TRB, 1997). Similar to density, occupancy that estimates temporal concentration of vehicles is also a reliable indicator of the traffic intensity on a road.

Apart from speed, density, and occupancy, V/C ratio could be another indication of congestion magnitude. However, the use of the ratio has a number of drawbacks.

Firstly, it is difficult to estimate traffic demand, especially in congested conditions where the "true" demand could not be translated directly from the measured flow rate, and is sensitive to the road capacity (see the explanation in Section 7.3.1 (ii) also); Secondly, the method requires conversion of different types of vehicles to passenger car unit (PCU), which is troublesome since the equivalent conversion factor depends on traffic composition, traffic condition, driver behaviour, and topography. Thirdly, the method requires determination of the temporarily reduced capacity that is hard to estimate properly under incident conditions.

For these reasons, there has been no single measure that adequately quantifies level of congestion. An inspection of data revealed that in many cases actual relationships between speed and density could not simply be fitted by straight lines or curves. It shows from different measurements (at different times) that a particular value of density may be associated with a considerably wide range of speeds, and a particular value of speed may be associated with a relatively wide range of density. Therefore, in defining congestion level, both traffic speed and density should be considered simultaneously. The use of speed and density has the advantage that they are primary measurements. Speed is easily obtainable from traffic sensors, while density can be determined by the measurement over a length of road of about several hundreds of meters from cameras mounted on tall buildings or on poles using a single frame. More importantly, the use of two out of three interdependent macroscopic traffic variables (volume, speed, density) is the necessary and sufficient conditions to represent the status of a traffic stream.

In the first rule block, state variables (speed and density) are evaluated using numerical data while the control variable (congestion level) is partitioned into a normalized scale $[0, CL_{\max}]$ where $CL_{\max}$ represent the maximum degree of belief, or the ultimate level of congestion:

$$CL = [0, CL_{\max}] = [0, 1] \text{ or} \quad (5.1)$$

$$CL = [0, CL_{\max}] = [0, 10] \quad (5.2)$$

5.2.2 Membership functions

The learning parameters of membership functions for speed and density has been conceptually presented in Chapter 4. Although expert judgements play an important role in this learning process, the selection of these parameters should also be made using available data. For example, the data inspection from the HDB helps identify the universes of discourse for speeds and densities, while the partitioning of the universes of discourse requires knowledge of traffic engineering as well as fuzzy clustering method following the data-oriented approach.

If n_{x1} denotes the number of linguistic predicates in the speed variable, n_{x2} denotes the number of linguistic predicates in the density variable, n_{x1} and n_{x2} will determine the number of rules in the first block. The maximum number of rules in this block will be $n_x = n_{x1} \times n_{x2}$. Assume that $n_{x1} = n_{x2} = 3$, and the linguistic predicates of the state and control variables are set as:

$$V = \{Low, Medium, High\} \quad (5.3)$$

$$K = \{Low, Medium, High\} \quad (5.4)$$

$$CL = \{Free_flow, Light, Moderate, Heavy\}. \quad (5.5)$$

Figure 5.2 plots initial membership functions for the speed domain. The piecewise linear shape (Figure 5.2a) has the advantage of simplicity, being formed by the collection of three data points (a, b, c). The membership functions at the lower and upper bounds are trapezoids with the core covering a segment. On the other hand, the curve style presented in Figure 5.2b has an advantage that the membership values are smoothed, more convenient in specifying fuzziness. Apart from that, the supports of membership functions of this style cover wider ranges to cater for noise, typically encountered in traffic data.

Figure 5.2: Membership functions of the speed domain ($n_{x1} = 3$)

Alternatively, the input space can be partitioned with a higher resolution, depending on the selection of predicates representing the space. Figure 5.3 describes the partition of the speed domain with $n_{x1} = 5$, corresponding to 5 fuzzy sets:

$$V = \{\text{Very_low}, \text{Low}, \text{Medium}, \text{High}, \text{Very_high}\} \quad (5.6)$$

Figure 5.3: Fuzzy partitioning the speed domain ($n_{x1} = 5$)

The same partitioning concept can be applied to the output universe of discourse. Again, the number of fuzzy sets is selected depending on specific applications and the design of the control actions. For example, in Figure 5.4a the control variable *congestion level* is represented by 4 fuzzy sets:

$$CL = \{\text{Free_flow}, \text{Light}, \text{Moderate}, \text{Heavy}\} \quad (5.7)$$

while in Figure 5.4b the variable is divided into 5 fuzzy sets:

$$CL = \{\text{Free_flow}, \text{Pre_con}, \text{Light}, \text{Moderate}, \text{Heavy}\} \quad (5.8)$$

Figure 5.4: Membership functions of the output space

The fuzzy sets in Equation (5.8) describe the whole range of congestion level. *Free_flow* may be associated with Levels Of Service (LOSs) A and B; *Pre-congestion* may be associated with LOS C where speed begins to decline slightly and density increases quickly with increasing flow, and freedom to manoeuvre within the traffic stream is noticeably limited; *Light* congestion may correspond to LOS D where speed begins to decline with increasing flow; *Moderate* congestion describes operation that approaches the road capacity (LOS E), where manoeuvrability within the traffic stream is very limited; and *Heavy* congestion describes breakdowns in vehicular flow, that corresponds to the LOS F. There exist queues forming behind breakdown points with potential propagation upstream for significant distances. It should be noted the terms are imprecisely defined and there are no clear cuts between the fuzzy congestion levels.

5.2.3 Fuzzy relations

Fuzzy relation

Fuzzy relation represents the association or interconnection between elements of two or more fuzzy sets (see Section 2.3.2). In traffic engineering, the relationship between speed and density is commonly described by the following statements:

R_1 : If speed is low then density is high

R_2 : If speed is medium then density is medium

R_3 : If speed is high then density is low

However, data inspections show that the statements should be confined to describing generic rule-of-thumb relationships between these variables in a broad sense. Specifically, they are not in total agreement with observed data in many cases. A low speed may be associated with medium density, or even low density of traffic. On the other hand, high speed may go in pair with medium or high density with different possibilities. From the perspective of fuzzy logic, the relationships can be described in the form of relational matrix. Let $V = \{Low, Medium, High\}$ denote the crisp set of speed, and $K = \{Low, Medium, High\}$ denote the crisp set of density. Assume that the interconnection between speed and density can be stated in Table 5.1.

Table 5.1: Relational matrix of speed and density

		Density		
Speed	R	Low	Medium	High
	Low	0.15	0.5	1
	Medium	0.5	1	0.5
	High	1	0.5	0.15

From Table 5.1, the rules R_1 , R_2 , and R_3 can be restated as R'_1 , R'_2 , and R'_3 , respectively:

R'_1 : If speed is low then density is high, less likely to be medium, and rarely to be low.

R'_2 : If speed is medium, then density is medium, less likely to be low or high.

R'_3 : If speed is high, then density is low, less likely to be medium, and rarely to be high.

Fuzzy compositional relations

The above concept of fuzzy relation between speed and density can be extended to define fuzzy relation between speed (or density) and congestion level, given the known relation between either of the quantities with congestion level. Understanding these relations is important particularly in the case of missing data since the congestion level can be estimated relying purely on a single variable. Let $R_1(v, k), (v, k) \in V \times K$ be the fuzzy relation in the product space $V \times K$, $R_2(k, cl), (k, cl) \in K \times CL$ be the fuzzy relation in the product space $K \times CL$, the fuzzy relation $R(v, cl), (v, cl) \in V \times CL$ in the product space $V \times CL$ can be derived using *MAX-MIN* or *MAX-PROD* compositions, as described in Chapter 2.

Assume that the relational matrix $R_1(v, k)$ is given as in Table 5.1, and the relational matrix $R_2(k, cl)$ is given as in Table 5.2.

Table 5.2: Relational matrix of density and congestion level

		Congestion level			
Density	R_2	Free_flow	Light	Moderate	Heavy
	Low	1	0.7	0.35	0
	Medium	0.25	0.75	0.75	0.25
	High	0	0.35	0.65	1

The membership grades of the relational matrix R can be obtained using *MAX-MIN* composition (see Equation (2.18)):

$$\mu_{R_1 \circ R_2}(v, cl) = \left\{ (v, cl), \max_k [\min(\mu_{R_1}(v, k), \mu_{R_2}(k, cl))] \right\} \quad (5.9)$$

The entries of the matrix R can be calculated as:

$$r_{11} = \max(\min(0.15, 1), \min(0.5, 0.25), \min(1, 0)) = \max(0.15, 0.25, 0) = 0.25 \quad (5.10)$$

... ..

$$r_{34} = \max(\min(1, 0), \min(0.5, 0.25), \min(0.15, 1)) = \max(0, 0.25, 0.15) = 0.25 \quad (5.11)$$

In this way of inference, the resulting relational matrix R is summarized in Table 5.3.

Table 5.3: Relational matrix of speed and congestion level

	R	Congestion level			
		Free_flow	Light	Moderate	Heavy
Speed	Low	0.25	0.5	0.65	1
	Medium	0.5	0.75	0.75	0.5
	High	1	0.7	0.5	0.25

The result in the relational matrix can be interpreted linguistically in the form of rules using *hedges* such as *possibly*, *likely*, or *unlikely*, *etc.*, for clarity:

If speed is *low*, then congestion level is *heavy*, possibly *moderate*, less likely *light*, and rarely *free flow*.

If speed is *medium*, then congestion level is possibly *light* or *moderate*, less likely *free flow* or *heavy*.

If speed is *high*, then congestion level is *free flow*, possibly *light*, less likely *moderate*, and rarely *heavy*.

5.3 CONGESTION MOBILITY

5.3.1 Membership functions

The congestion mobility rule block examines another aspect of incident traffic condition: the dynamics of congestion. Having evaluated the congestion level in the first rule block, congestion type *heavy* is tracked in another block and treated together with traffic speed to see how fast the so call *heavy traffic* moves. This rule block takes two input variables, speed and congestion level, to evaluate one output variable: (C_Mob). The membership functions of the state variables are the same as in the first block. The universe of discourse of the control variable C_Mob is normalized in scale:

$$C_Mob = [0, CM_{\max}] = [0, 10] \quad (5.12)$$

The congestion mobility consists of two fuzzy sets:

$$C_Mob = \{SM_HC, MV_HC\} \quad (5.13)$$

The abbreviations stand for *slow moving - heavy congestion* and *medium moving - heavy congestion*, respectively. The term "*fast moving - heavy congestion*" is not included since *fast moving* and *heavy congestion* are mutually exclusive.

The membership functions of C_Mob are all convex and normal, constructed by equally partitioning the output space (Figure 5.5).

Figure 5.5: Membership functions of congestion mobility

5.3.2 Fuzzy relations

Given that the relationships between variables in Section 5.2.3 can be learnt, in case of missing data the congestion level may solely be defined by speed rather than by speed and density together. Let R be a fuzzy relation from the universe of discourse V to the universe of discourse CL , and $v_{()}$ a fuzzy subset of V , then the fuzzy subset $cl_{()}$ of CL is obtained from Equation (5.14):

$$cl_{()} = v_{()} \circ R \quad (5.14)$$

This compositional inference can be used to define rules governing the relationship between a fuzzy subset $v_{()}$ in V and a fuzzy subset $cl_{()}$ in CL .

Relations between fuzzy subsets

Define the relational matrix of the concept “slow moving - heavy congestion”.

Since the universes of discourse (V and K) are continuous sets comprising infinite number of elements, the membership functions of the fuzzy relation is a surface over the Cartesian product $V \times K$, and the relational matrix of the fuzzy relation is a (∞, ∞) matrix. For simplicity the universes of discourse are represented by discretized intervals (Figure 5.6).

a) *Low* fuzzy subset of speed

b) *Heavy* subset of congestion level

Figure 5.6: Compositional inference of slow moving - heavy congestion

If the interval of 10 (km/h) is used to discretize the speed domain (Figure 5.6a), the fuzzy subset *low speed* can be represented by the crisp set $V_{Low} = [0, 10, 20, 30, 40, 50]$.

Similarly, the fuzzy subset *heavy congestion* is represented by the crisp set $CL_{Heavy} = [6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10]$ using the granularity of 0.5 (Figure 5.6b).

The corresponding membership grades of the two fuzzy sets are:

$$\mu_{Low}(V) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0.67 \\ 0.33 \\ 0 \end{bmatrix} \quad (5.15)$$

$$\mu_{Heavy}^T(CL) = [0 \quad 0.2 \quad 0.4 \quad 0.6 \quad 0.8 \quad 1 \quad 1 \quad 1 \quad 1] \quad (5.16)$$

Since the concept *slow moving - heavy congestion* implies *low speed* AND *heavy congestion*, the intersection operation is used to combine these two fuzzy subsets on the universes of discourse formed by speed V and congestion level CL . The membership values of the relational matrix of the two fuzzy sets can be estimated as:

$$\mu_R(V, CL) = \min(\mu_{Low}(V), \mu_{Heavy}(CL)) \quad (5.17)$$

The elements of the relational matrix $R_{SM-HC} = \mu_{Low}(V) \times \mu_{Heavy}^T(CL)$ can be calculated using Equation (5.17). Results are summarised in Table 5.4.

Table 5.4: Relational matrix of low speed and heavy congestion

		Heavy congestion								
Low speed	R_{SM-HC}	6	6.5	7	7.5	8	8.5	9	9.5	10
	0	0	0.20	0.40	0.60	0.80	1	1	1	1
	10	0	0.20	0.40	0.60	0.80	1	1	1	1
	20	0	0.20	0.40	0.60	0.80	1	1	1	1
	30	0	0.13	0.27	0.40	0.54	0.67	0.67	0.67	0.67
	40	0	0.07	0.13	0.20	0.26	0.33	0.33	0.33	0.33
	50	0	0	0	0	0	0	0	0	0

The concept of *medium moving - heavy congestion* can be defined in the same way (Figure 5.7).

Figure 5.7: Compositional inference of medium moving - heavy congestion

$$V_{Medium} = [20, 30, 40, 50, 60, 70] \quad (5.18)$$

$$CL_{Heavy} = [6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10] \quad (5.19)$$

$$\mu_{Medium}(V) = \begin{bmatrix} 0 \\ 0.33 \\ 0.67 \\ 1 \\ 0.67 \\ 0.33 \\ 0 \end{bmatrix} \quad (5.20)$$

$$\mu_{Heavy}^T(CL) = [0 \quad 0.2 \quad 0.4 \quad 0.6 \quad 0.8 \quad 1 \quad 1 \quad 1 \quad 1] \quad (5.21)$$

The results of $R_{MV-HC} = \mu_{Medium}(V) \times \mu_{Heavy}^T(CL)$ are summarised in Table 5.5:

Table 5.5: Relational matrix of medium speed and heavy congestion

		Heavy congestion								
	R _{MV-HC}	6	6.5	7	7.5	8	8.5	9	9.5	10
Medium speed	20	0	0	0	0	0	0	0	0	0
	30	0	0.066	0.132	0.198	0.264	0.330	0.330	0.330	0.330
	40	0	0.134	0.268	0.402	0.536	0.670	0.670	0.670	0.670
	50	0	0.2	0.4	0.6	0.8	1	1	1	1
	60	0	0.134	0.268	0.402	0.536	0.670	0.670	0.670	0.670
	70	0	0.066	0.132	0.198	0.264	0.330	0.330	0.330	0.330
	80	0	0	0	0	0	0	0	0	0

The values of elements in the relational matrices R_{SM-HC} and R_{MM-HC} indicate the degrees of fulfilment of the relations in the product universes of discourse $V \times CL$. They reflect the structured-human knowledge and can be transformed into fuzzy *IF-THEN* rules. In case of two input variables and one output variable, the relation matrix will be a triple-fuzzy relation having membership function represented by a hyper surface over three-dimensional space: speed-density-congestion_level, known as the surface knowledge (Figure 5.8). With an exception of a few local variations, the surface knowledge indicates a smooth transition (a $\Re^2 \rightarrow \Re^1$ mapping) from speed and density to congestion level.

Figure 5.8: Surface knowledge of $CL(v \times k)$

5.4 CONGESTION STATUS

If *congestion level* evaluates the intensity of congestion and *congestion mobility* estimates how fast the congestion moves, *congestion status* quantifies the spatial magnitude of the congestion being considered given the number of vehicles in queue. A queue starts as demand exceeds available capacity. Under severe incidents, a lane blockage temporarily reduces the capacity of the road, possibly leading to traffic breakdown and formation of queue upstream of the incident. The queue length or the number of vehicles in queues signifies the severity of the incident congestion; thus the evaluation of the queuing status is important before proposing control actions.

It is probable that queues only form under heavily congested situations. Therefore, in the third rule block, the MS-FLC evaluates the status of congestion based upon the queue length under the heavy congestion category. Figure 5.9 plots tentative membership functions for queue length on expressways.

Figure 5.9: Linguistic values of queue length

The linguistic values of the queue length variable are set as:

$$Queue = \{Short, Medium, Long\} \quad (5.22)$$

and linguistic values for the congestion status are set as (Figure 5.10):

$$C_Stat = \{SQ - HC, MQ - HC, LQ - HC\} \quad (5.23)$$

The abbreviations stand for *short queue – heavy congestion*, *medium queue – heavy congestion*, and *long queue – heavy congestion*, respectively.

Figure 5.10: Fuzzy sets of congestion status

The parameters of membership functions of the queue length should primarily be learnt from traffic control practice, whereas the partitioning of the domain of congestion status may be made using grid clustering. It should be noted that although the term *queue* is associated with heavy congestion, it does not necessarily imply the standstill traffic. Rather, it refers to slow-moving traffic streams. As an example, the queue configuration set for average spacing is between 10 and 20 m, and for speed range between 5-10 km/h (Quadstone, 2004).

5.5 FORMATION OF RULES

5.5.1 Preliminary rule set

The concept of fuzzy relation presented above provides insight into mapping mechanisms between state and control variables. Rules in the first block are characterized by two predicates in the antecedent, connected with an *AND* operator, and one predicate in the consequent. The general expression of rules is of the form:

$$\text{If speed is } V_{(x)} \text{ AND density is } K_{(x)} \text{ then congestion level is } CL_{(x)} \quad (5.24)$$

that can be symbolically represented as:

$$(V \bullet V_{(x)}(v)) \Theta_{\min} (K \bullet K_{(x)}(k)) \rightarrow CL \bullet CL_{(x)}(v, k) \quad (5.25)$$

where the suffice (x) indicates any of the linguistic predicates in the corresponding variable.

The generic rule in Equation (5.24) can be restated in a more concrete form:

$$R_1 : \text{if speed is } A_1 \text{ AND density is } B_1 \text{ then congestion level is } C_1$$

Also

$$R_2 : \text{if speed is } A_2 \text{ AND density is } B_2 \text{ then congestion level is } C_2$$

... ..

Also

$$R_n : \text{if speed is } A_n \text{ AND density is } B_n \text{ then congestion level is } C_n$$

where A_i, B_i and C_i ($i = 1, \dots, n$) are the linguistic values of speed, density, and congestion level in the universes of discourse V, K and CL , respectively.

Being connected with *AND* operator, the membership values of the rules are calculated using *MIN* operation, namely the intersection $\mu_{(x)}(V) \cap \mu_{(x)}(K)$:

$$\mu_{V \cap K} = \min(\mu_{(x)}(V), \mu_{(x)}(K)) \quad (5.26)$$

Collection of rules in the first stage is listed as follows:

- R_1 : *If V is high and K is low then CL is Free_flow*
- R_2 : *If V is high and K is medium then CL is Light*
- R_3 : *If V is high and K is high then CL is Moderate*
- R_4 : *If V is medium and K is low then CL is Light*
- R_5 : *If V is medium and K is medium then CL is Moderate*
- R_6 : *If V is medium and K is high then CL is Heavy*
- R_7 : *If V is low and K is low then CL is Moderate*
- R_8 : *If V is low and K is medium then CL is Heavy*
- R_9 : *If V is low and K is high then CL is Heavy*
- R_{10} : *If V is low and CL is heavy then C_Mob is SM-HC*
- R_{11} : *If V is medium and CL is heavy then C_Mob is MM-HC*
- R_{12} : *If Queue is short and CL is heavy then C_Stat is SQ-HC*
- R_{13} : *If Queue is medium and CL is heavy then C-Stat is MQ-HC*
- R_{14} : *If Queue is long and CL is heavy then C-Stat is LQ-HC*

where *SM-HC*, *MM-HC* stand for *slow moving - heavy congestion* and *medium moving - heavy congestion*, and *SQ-HC*, *MQ-HC*, and *LQ-HC* stand for *short queue – heavy congestion*, *medium queue – heavy congestion*, and *long queue – heavy congestion*, respectively.

The collection describes the maximum number of rules triggered by the three blocks. The block congestion level consists of 9 rules ($R_1 - R_9$) since each state variable has three values ($n_1 \times n_2 = 9$). The blocks congestion status and congestion mobility represent the simple combinations of speed with heavy congestion as well as queue length with heavy congestion, respectively, leading to a total of $9+2+3 =$

14 rules. In other words, the elementary rule base considers the maximum combinations of state variables.

A question may arise at this point: whether the rule collection overfits data? In conventional rule base system, as stated in Section 5.2.3, the relationship between speed and density is typically stated as:

If speed is low then density is high

If speed is medium then density is medium

If speed is high then density is low.

If speed and density were so strongly and uniquely correlated, the use of either of them would suffice, and the collection of 9 rules in the first block would be reduced to 3 rules:

R_1^* : *If V is high OR K is low then CL is Free_flow*

R_2^* : *If V is medium OR K is medium then CL is Light*

R_3^* : *If V is low OR K is high then CL is heavy.*

Nevertheless, the link between the two variables is not so straightforward. From the fuzzy logic's perspective, any combination may be possible with the strength exhibited by membership grades, being the possibilities of occurrence. In Section 5.2.3 it is reviewed that the relationship between speed and density is described as:

If speed is low then density is high, less likely to be medium, and rarely to be low

If speed is medium then density is medium, less likely to be low or high

If speed is high then density is low, less likely to be medium, and rarely to be high

Note that the term *high*, *medium*, and *low* are vague linguistics in fuzzy logic since they are ill-defined and there may be no clear cuts between them.

If the above statements hold true, R_3 (if V is *high* and K is *high* then...), R_7 (if V is *low* and K is *low* then...), and R_{12} (if V is *high* and CL is *heavy* then...) rarely happen in practice. Consequently, in a complex rule base, they are occasionally discarded for simplicity.

5.5.2 Rule tuning

The preliminary rule set constructed above is associated with a low resolution of the fuzzy partition, where each state variable is represented by three fuzzy values. If the speed and density encompasses large ranges, and the MS-FLC requires a finer linguistic definition, the fuzzy variables can be partitioned with a higher resolution. Assume that speed and density are now re-partitioned into $n_{x1} = n_{x2} = 5$, represented by collections of fuzzy sets (Figures 5.11 and 5.12):

$$V = \{\text{Very_low}, \text{Low}, \text{Medium}, \text{High}, \text{Very_high}\} \quad (5.27)$$

$$K = \{\text{Very_low}, \text{Low}, \text{Medium}, \text{High}, \text{Very_high}\} \quad (5.28)$$

Figure 5.11: Fuzzy sets of speed ($n_{x1} = 5$)

Figure 5.12: Fuzzy partition of density ($n_{x2} = 5$)

Whereas the control variable is partitioned into 5 fuzzy sets (Figure 5.13):

$$CL = \{Free_flow, Pre_con, Light, Moderate, Heavy\} \quad (5.29)$$

Figure 5.13: Fuzzy partition of congestion level

All the membership function's line styles are now the curves (Gaussian) with wide supports to capture the noises in data measurement. The relationships between speed and density are now revised with an empirical assignment of fuzzy values in Table 5.6:

Table 5.6: Revised relational matrix of speed and density

R	Density				
	Very_low	Low	Medium	High	Very_high
Very_low	0	0.25	0.5	0.75	1
Low	0.25	0.5	0.75	1	0.75
Medium	0.5	0.75	1	0.75	0.5
High	0.75	1	0.75	0.5	0.25
Very_high	1	0.75	0.5	0.25	0

The values of the matrix can be transformed linguistically in the following rules:

If speed is very low then density is very high, possibly high, less likely medium, rarely low, and unlikely very low

If speed is low then density is high, possibly medium or very high, less likely low, and rarely very low

If speed is medium then density is medium, possibly low or high, and less likely very low or very high

If speed is high then density is low, possibly very low or medium, less likely high, and rarely very high

If speed is very high then density is very low, possibly low, less likely medium, rarely high, and unlikely very high.

Figure 5.14 shows a snapshot of the rule set for congestion level, generated by the FIS. The rule set is the collection of all possible relational combinations of speed and density, except the cases *very low* speed – *very low* density and *very high* speed - *very high* density that are unlikely possible in practice, having learnt from the above conclusion.

-
1. If (Speed is Very-low) and (Density is Low) then (Congestion_level is Moderate) (1)
 2. If (Speed is Very-low) and (Density is Medium) then (Congestion_level is Heavy) (1)
 3. If (Speed is Very-low) and (Density is High) then (Congestion_level is Heavy) (1)
 4. If (Speed is Very-low) and (Density is Very-high) then (Congestion_level is Heavy) (1)
 5. If (Speed is Low) and (Density is Very-low) then (Congestion_level is Light) (1)
 6. If (Speed is Low) and (Density is Low) then (Congestion_level is Light) (1)
 7. If (Speed is Low) and (Density is Medium) then (Congestion_level is Moderate) (1)
 8. If (Speed is Low) and (Density is High) then (Congestion_level is Heavy) (1)
 9. If (Speed is Low) and (Density is Very-high) then (Congestion_level is Heavy) (1)
 10. If (Speed is Medium) and (Density is Very-low) then (Congestion_level is Pre-con) (1)
 11. If (Speed is Medium) and (Density is Low) then (Congestion_level is Pre-con) (1)
 12. If (Speed is Medium) and (Density is Medium) then (Congestion_level is Light) (1)
 13. If (Speed is Medium) and (Density is High) then (Congestion_level is Moderate) (1)
 14. If (Speed is Medium) and (Density is Very-high) then (Congestion_level is Heavy) (1)
 15. If (Speed is High) and (Density is Very-low) then (Congestion_level is Free-flow) (1)
 16. If (Speed is High) and (Density is Low) then (Congestion_level is Pre-con) (1)
 17. If (Speed is High) and (Density is Medium) then (Congestion_level is Light) (1)
 18. If (Speed is High) and (Density is High) then (Congestion_level is Light) (1)
 19. If (Speed is High) and (Density is Very-high) then (Congestion_level is Moderate) (1)
 20. If (Speed is Very-high) and (Density is Very-low) then (Congestion_level is Free-flow) (1)
 21. If (Speed is Very-high) and (Density is Low) then (Congestion_level is Free-flow) (1)
 22. If (Speed is Very-high) and (Density is Medium) then (Congestion_level is Pre-con) (1)
 23. If (Speed is Very-high) and (Density is High) then (Congestion_level is Light) (1)

Figure 5.14: Rule set for congestion level

The above rule set can be summarized in the form of decision matrix (Table 5.7).

Table 5.7: Rule decision matrix of congestion level

(FF: Free_flow, Pre: Pre-congestion, L: Light congestion,
M: Moderate congestion, H: Heavy congestion)

	R	Density				
		Very_low	Low	Medium	High	Very_high
Speed	Very_low	---	M	H	H	H
	Low	L	L	M	H	H
	Medium	Pre	Pre	L	M	H
	High	FF	Pre	L	L	M
	Very_high	FF	FF	Pre	L	---

Figure 5.15 shows the rule interface for congestion level using the above rule set. With speed $V = 35.8$ and density $K = 66.1$ most of the rules, except rules 15, 16, 20, 21, are active. The outputs from activated rules are aggregated using the *MAX* operation. The defuzzification uses the Centroid method that produces a crisp value of congestion level $CL = 7.58$. This corresponds to heavy congestion with degree of membership of $\mu_{Heavy}(CL) = 0.72$, and medium congestion with degree of membership of $\mu_{Medium}(CL) = 0.24$. These degrees of membership will be used as internal inputs in the second stage.

Figure 5.15: Rule viewer for congestion level (Mamdani type)

Example with rule 7:

If speed is Low AND density is Medium then congestion level is Moderate

With the given values of speed and density, the membership grades for speed is $\mu_L(V)=0.85$, and for density is $\mu_M(K)=0.36$. Using the AND connective the intersection *t-norms* can be derived using the *MIN* operation:

$$\mu_M(CL)=\mu_L(V)\cap\mu_M(K)=MIN(\mu_L(V),\mu_M(K))=MIN(0.85,0.36)=0.36 \quad (5.30)$$

5.5.3 Evaluation of congestion level using actual data

The following section demonstrates the evaluation of congestion level using data on segment 80007762, Oct. 9th, 2003. Speed and density are derived from the field data collected in 5-min. granularity and are plotted in Figure 5.16.

Figure 5.16: Speed and density profiles

Since data in the HDB are point measurements, they do not permit direct calculation of speed and density. Speeds used in this analysis were space mean speeds ($\bar{U}_s$) that are estimated from the measured time mean speeds ($\bar{U}_t$) using Equation (5.31):

$$\bar{U}_s \approx 1.026 \times \bar{U}_t - 1.89 \quad (5.31)$$

if the time mean speed is less than 70 km/h (HCM, 1997). It should be noted that Equation (5.31) is used to approximate the space mean speeds from time mean speeds in absence of data for the regression of the model under the local condition. For the applications that require precise description of the relationship between the two types of speed, the relationship should be calibrated with field data.

For a time mean speed that is greater than 70 km/h, space mean speed is approximated as 2% less than the time mean speed (Taylor et al., 2000). From the

estimated space mean speed, density is calculated as the ratio of flow rate V (vehicles/lane/h) to the space mean speed $\bar{U}_s$ (km/h).

$$K = \frac{V}{\bar{U}_s} \quad (5.32)$$

To illustrate what have been discussed in Section 5.2.1 “the use of two out of three interdependent macroscopic traffic variables (volume, speed, density) is the necessary and sufficient conditions to represent the status of a traffic stream”, the congestion level estimated using both speed and density $CL(v*k)$ is compared with hypothetical congestion levels using a single quantity, namely speed ($CL(v)$) and density ($CL(k)$), respectively. Three methods use the same sets of membership functions proposed in Figures 5.11 to 5.13.

The membership grades for every fuzzy set and the congestion levels from the FIS system are listed in Table 5.8. The FIS is of Mamdani type that uses *MIN* for implication, *MAX* for aggregation, and *Centroid* defuzzification technique. Using speed as the only input variable, the input-output mapping of $CL(v)$ is implemented by a simple set of rules:

- *If speed is very low then congestion level is heavy*
- *If speed is low then congestion level is moderate*
- *If speed is medium then congestion level is light*
- *If speed is high then congestion level is pre-congestion*
- *If speed is very high then congestion level is free-flow.*

Table 5.8: Evaluation of congestion levels by three methods

(Notes: VL: Very low; L: Low; M: Medium; H: High; VH: Very high)

Time	Speed	Density	Membership Grade										CL(v*k)	CL(v)	CL(k)
			Speed_CL					Den_CL							
			VL	L	M	H	VH	VL	L	M	H	VH			
5:13:20	97	11	0	0	0	0	1	0.75	0.68	0	0	0	1.53	1.33	2.28
5:18:20	96	16	0	0	0	0	1	0.56	0.76	0.18	0	0	1.58	1.35	2.79
5:23:19	87	10	0	0	0	0.29	0.81	0.78	0.65	0	0	0	1.69	1.79	2.19
5:28:19	85	18	0	0	0	0.32	0.78	0.52	0.92	0.15	0	0	1.81	1.97	3.01
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
8:43:19	74	27	0	0	0.08	0.95	0.19	0.1	0.93	0.87	0	0	2.44	2.98	3.95
8:48:21	70	34	0	0	0.25	1	0.16	0	0.4	0.97	0.15	0	3.58	3.29	4.77
8:53:24	75	40	0	0	0.1	0.89	0.23	0.07	0.86	0.48	0	0	3.99	2.90	5.64
8:58:26	78	42	0	0	0	0.73	0.41	0.05	0.83	0.51	0	0	4.10	2.66	5.87

The membership functions of the $CL(v)$ is illustrated in Figure 5.17.

Figure 5.17: The rule viewer for $CL(v)$

Similarly, the $CL(k)$ are obtained using the simple set of rules:

- If density is very low then congestion level is free-flow
- If density is low then congestion level is pre-congestion

- *If density is medium then congestion level is light*
- *If density is high then congestion level is moderate*
- *If density is very high then congestion level is heavy.*

The membership functions of the $CL(k)$ is illustrated in Figure 5.18.

Figure 5.18: The rule viewer for $CL(k)$

The evaluated congestion levels are plotted in Figure 5.19. The data profiles in the figure show that although the general relationship between speed and density basically follows what is formally known as stated in Section 5.2.3, it is not unique: speed is not very sensitive to density in free-flow condition, and a small range of speed is sometimes associated with a wide range of density, and vice-versa. Figure 5.19 shows that there are large differences between the congestion levels evaluated by $CL(v)$ and $CL(k)$ under free-flow condition. In principle, if $CL(v)$ and $CL(k)$ were good indicators of congestion level, they would be sufficiently close since they used the same set of data. This proves that it would be inadequate to represent the congestion level using a single variable of speed or density. The congestion level using both speed and density will provide better representation of traffic conditions.

Figure 5.19: Evaluation of congestion levels by different methods

Figure 5.20 plots the association between the state and control variables for $CL(v)$ and $CL(k)$ respectively, represented by lines. By contrast, the association between speed, density and congestion level in the case of $CL(v*k)$ is represented by a surface knowledge (Figure 5.8).

a) V-CL(v)

b) K-CL(k)

Figure 5.20: State-control fuzzy relations using a single variable

5.6 SUMMARY

This chapter presents a comprehensive appraisal of congestion situation, including congestion level, congestion mobility and congestion status. The congestion level evaluates the severity of congestion using speed and density that is derived from field data in the HDB, being principal quantifiable traffic variables. Various methods of fuzzy partition of state and control variables, including the granularity and line style have been proposed. The fuzzy relations between state variables as well as between fuzzy values in different variables have been explored. The establishment of fuzzy relations is important in dealing with the problem of missing data. In addition, the well-defined relationships between speed and density in traffic engineering are extended to fuzzy logic rules from the fuzzy logic angle.

The quantification of the current traffic condition also addresses another issue of congestion, expressed by the terms congestion mobility, which estimates the dynamics of traffic streams under heavy congestion category. Fuzzy relations are explored to get insights into understanding complex terminologies, including SM-HC and MM-HC. The degree of fulfilment of each fuzzy relation depends on the existence of traffic speed in the corresponding heavy congestion subset.

Finally, the congestion status determines the spatial magnitude of congestion, given the number of vehicles in a queue. Like congestion mobility, congestion status considers the length of a queue in the existence of heavy congestion category. The congestion status is classified into *SQ-HC*, *MQ-HC*, and *LQ-HC* fuzzy terms. Different control measures will be issued in response to different congestion status.

CHAPTER 6 ANTICIPATION OF INCIDENT TRAFFIC CONDITIONS

6.1 SCHEMATIC RULE FLOW

6.1.1 Overview

The anticipation of incident-related traffic conditions is the 2nd and the intermediate stage of the MS-FLC. The stage is essential in an attempt to see how incident-related traffic conditions evolve so that suitable solutions for proactive traffic control can be recommended. The anticipation of traffic conditions under incident situations is a complex multivariate process that involves short-term forecast of traffic variables such as volume, speed, and travel time. Owing to the fact that traffic conditions during incidents are characterised by random fluctuations, imposing more challenges on traffic forecast, further considerations and adjustments to the predicted result by a risk factor is necessary. The use of the risk factor is to cater for unknown and unexpected impacts from the traffic environment that may reduce the prediction accuracy.

In the prediction of traffic variables, the SVM technique will be deployed. The rationale behind the application of the SVM technique in conjunction with fuzzy logic has been explained in Chapter 3. Particularly, SVMs possess advantages of machine learning in prediction of traffic patterns, dealing efficiently with incident conditions, which are considered to be the most challenging issue in the state-of-the-art short-term traffic forecasting. The merits from the applications of the technique will be presented in Section 6.2.

Figure 6.1 describes the schematic process of the 2nd stage of the MS-FLC. The stage consists of the estimation of the risk factor, the prediction and adjustment of traffic variables, and the anticipation of the evolution of traffic conditions.

Figure 6.1: Schematic rule flow for the 2nd stage

6.1.2 Risk factor

The risk factor is an adjustment to cater for external risks that exist exogenously with the prediction being made. The traffic forecasting techniques are typically data-driven, relying purely on data collected within a time window in the past. In other words, the prediction modules has no reasoning instrument to take account of random fluctuations influencing the traffic environment in future, which may be particularly high under incident situations. The application of the risk factor is an attempt to take advantage of reasoning capabilities in fuzzy logic to compensate for shortcomings of data-driven traffic forecast in general and of the SVM prediction technique in particular. Other than the congestion level produced in the first stage that is known as the internal input, the estimation of the risk factors involves considering several external variables, ranging from the effectiveness of the control strategy being implemented, incident type, incident severity, the time of day, the weather condition, etc. Since the consideration of all influencing factors is not possible, and the number of rules increases exponentially with the number of inputs, it may be more suitable to consider the most influencing variables, including the effectiveness of the control strategy, the incident severity, and the time of day.

(i) Effectiveness of the control strategy

Since the magnitudes of traffic parameters and the control actions being implemented are mutually dependent, it is desirable to take into account the effectiveness of the control actions, which can be represented by the degree the

drivers conform to the control directives, such as the proportion of drivers following route-diversion or lane-changing messages. The level of conformity depends on a number of factors, such as the strength and content of VMS messages, driver behaviour, network attributes, incident severity, and traffic congestion level (Section 2.1.2). Equation (6.1) describes the level of conformity by a set of three linguistic terms:

$$Conform_lev = \{Weak, Medium, Strong\} \quad (6.1)$$

Under a diversion scheme, the diversion flow can be split into deterministic and random components. The deterministic component is the default factor, being the proportion of traffic following the diversion route in absence of the diversion scheme. This component is usually estimated from historical time-dependent OD demands. The second component expresses the additional amount of traffic following the diversion command, the level of which may be low, medium or high depending on the aforementioned factors. If historical time-dependent OD demands properly represent the deterministic component, the magnitude of the random component will govern the reliability of the volume forecast. On the other hand, if the historical time-dependent OD demands do not properly reflect the deterministic component, the prediction of volume will be less reliable. While it is desirable to prescribe the risk based on the random component, it is difficult to split the two components since the diversion volume is usually collected as a whole from the traffic stream.

The membership functions for the conformity level are therefore apparently site-specific. A set of membership function templates for conformity level adhered to the diversion rate as a whole is shown in Figure 6.2. The parameters (a, b, c) need to be calibrated for specific diversion locations.

Figure 6.2: A conformity level template

(ii) Incident severity

The incident severity, in this context, is represented by the percentage of capacity reduction (Figure 6.3), which is estimated from the number of lane closure in comparison to the total number of available lanes. Without loss of generality, it is assumed that the higher the capacity reduction, the higher the risk the prediction encounters. In other words, the traffic prediction is highly error-prone under severe incidents. In Equation (6.2), the capacity reduction is represented by three fuzzy sets:

$$CapR = \{Slight, Medium, Severe\} \quad (6.2)$$

Figure 6.3: Membership functions for capacity reduction

(iii) The time of day

Although traffic forecasting modules utilises the real-time data in the rolling horizon to project future flows (Section 6.2.2), the rolling horizon pertains to the immediate (local) historical trend, while the time of day (TOD) reflects the global tendency of background traffic prevailing for a longer period, and the question of

whether the local trend always matches the global tendency is not warranted. For example, the local historic trend may incur random fluctuation (local uptrend or downtrend) while the peak rises are periods of intense use of traffic, characterized by a steady rise (always uptrend) of traffic flow. Therefore the TOD explicitly takes account of potential deviations of the local trend from the global tendency. In this regard, the TOD should be an important factor in the adjustment in the sense that with the same prediction error it may be highly risky in peak hours, but less risky in off-peak or night times since the traffic demands during these periods are lower, and the road still has available capacity to accommodate the error surcharge. The fuzzy sets for TOD differentiate the peak from the off-peak (day time) and the nighttimes:

$$TOD = \{Peak, OffPeak, NightTime\} \quad (6.3)$$

Figure 6.4 presents the concept of learning membership parameters for TOD from traffic volumes on the stretch of segment (Figure 4.4) on PIE. The peak rises have similar patterns from day to day (weekdays) but started at different time points. For example, the start times of the morning peaks typically range from 5.30 to 6.00 AM, while the end times range in between 8.00 and 8.30 AM.

Figure 6.4: Detection of decision points for TOD membership functions

(S_{AM} , E_{AM} , S_{PM} , and E_{PM} : start and end times of AM and PM peaks, respectively)

Figure 6.5 plots the membership functions for the TOD for that segment.

Figure 6.5: Membership functions for TOD

It should be noted that the use of the TOD is to cater for external risks, taking into account the fact that the prediction may be error prone due to potential random fluctuations, not to supersede the temporal variations of traffic volume that is explicitly captured in the data in the prediction horizon.

(iv) Risk factor

Being the function of traffic environment variables accounting for exogenous effects, the risk factor is a collection of linguistic terms that represent various risky levels, from *no risk* to *extream risk*. In Equation (6.4) the risk factor is labelled by a collection of 4 fuzzy sets:

$$Risk = \{Low, Medium, High, Very_high\} \quad (6.4)$$

Figure 6.6 plots the fuzzy sets for the risk factor, equally spaced in the scale of 10.

Figure 6.6: Membership functions for the risk factor

The risks factors are evaluated based on three state variables: the level of conformity (*Conform_lev*), the capacity reduction (*CapR*), and the time of day (*TOD*). The mapping between the state variables and the control variable (*Risk*) is made so that it forms an elegant transition from one fuzzy value to another, where the fuzzy values cover the whole of the output space: if the state variables indicate that traffic is in favourable conditions (strong conformity level, low capacity reduction, off-peak period), the risk will be evaluated as “Low”. By contrast, when state variables indicate critical conditions, the risk will be evaluated as “High” or “Very_high”. The mapping between the input variables and the risk factor is summarized in Table 6.1. In the table, each row signifies one rule. For instance, row 1 is expressed as:

If Conform_lev is Weak and CapR is Severe and TOD is Peak then Risk is Very_high.

Since the input variables are independent of each other, the number of rules in the rule set is the maximum combination of scenarios ($3 \times 3 \times 3 = 27$ rules). The rule set is consistent that there are no different outputs from the same set of inputs. Nevertheless, since the number of independent input combinations (27) is larger than the number of output sets (4), there are cases that different input combinations produce the same output, thus rules can be combined. For example, rules 6 and 9 can be merged into one rule using *AND/OR* composite operator:

If Conform_lev is Weak and CapR is (Medium OR Slight) and TOD is NightTime then Risk is Low.

Similarly, rules 7 and 8, rules 26 and 27, etc. can be formulated in the same fashion. However, the combination of rules in this manner does not yield significant reduction in the number of rules, while causing operational complications for the rule inference. Therefore, the rules should remain separate for simplicity.

Table 6.1: Decision table for the risk factor

Rule	If		Then	
	<i>Conform_lev</i>	<i>CapR</i>	<i>TOD</i>	<i>Risk</i>
1	Weak	Severe	Peak	Very_high
2	Weak	Severe	OffPeak	High
3	Weak	Severe	NightTime	Medium
4	Weak	Medium	Peak	High
5	Weak	Medium	OffPeak	Medium
6	Weak	Medium	NightTime	Low
7	Weak	Slight	Peak	Medium
8	Weak	Slight	OffPeak	Medium
9	Weak	Slight	NightTime	Low
10	Medium	Severe	Peak	High
11	Medium	Severe	OffPeak	Medium
12	Medium	Severe	NightTime	Low
13	Medium	Medium	Peak	Medium
14	Medium	Medium	OffPeak	Low
15	Medium	Medium	NightTime	Low
16	Medium	Slight	Peak	Medium
17	Medium	Slight	OffPeak	Low
18	Medium	Slight	NightTime	Low
19	Strong	Severe	Peak	High
20	Strong	Severe	OffPeak	Medium
21	Strong	Severe	NightTime	Low
22	Strong	Medium	Peak	Medium
23	Strong	Medium	OffPeak	Low
24	Strong	Medium	NightTime	Low
25	Strong	Slight	Peak	Low
26	Strong	Slight	OffPeak	Low
27	Strong	Slight	NightTime	Low

Since the 2nd stage deals primarily with the anticipation of incident-related traffic conditions, this Chapter reserves an important part for the SVM technique. In the following, the investigation of SVM application potentials in forecasting of traffic variables is presented in Sections 6.2 and 6.3.

6.2 SHORT-TERM PREDICTION OF TRAFFIC VOLUME

6.2.1 Introduction

Accurate short-term prediction of traffic parameters is an essential ingredient for any proactive traffic control system in ATMSs. The need for accurately predicted traffic flow has been recognized, and reliable traffic flow prediction has been a subject of intense research. Traffic flow prediction techniques have their sources from various disciplines, ranging from historical profiling (Smith and Demetsky, 1997; Oswald et al., 2001), Kalman filtering (Okutani and Stephanedes, 1984; Whittaker et al., 1997), nonparametric statistical methods (Davis and Nihan, 1991; Smith et al., 2000), sequential learning (Chen and Grant-Muller, 2001), exponential smoothing (Williams et al., 1997, Oswald et al., 2001), and time-series ARIMA models (Williams et al., 1997; Chang and Miaou, 1999; Williams, 2001). Each of the afore-mentioned approaches has its predictive advantages and limits, and results exhibit different degrees of success (Chen and Grant- Muller, 2001).

In an attempt to look for a more reliable technique, interests since the mid-90s have focused on the application of neural networks (Dia, 2001; Smith and Demetsky, 1994). The neural network is naturally a methodological candidate for forecasting with multiple inputs and outputs. As the traffic process is non-linear, neural networks bring a coherent framework of non-linear regression models to the problem (Van Arem et al., 1997). The Neural Network approach, with its parallel structure and learning capability, is suitable for solving complex problems like prediction of traffic parameters (Vanajakshi and Rilett, 2004). However, the method requires extremely long learning time, is prone to bad generalization and poor performance on previously unseen data during the test phase (Kecman, 2001). A

comparison between neural networks and standard statistical models made by Van Arem et al. (1997) showed that “neural networks performed almost as well as time-series models of the ARIMA type for simple sequences of links but not so well as special purpose pattern recognition techniques”. Kirby et al. (1997) made a comparison between neural networks and time-series ARIMA model and concluded that neural networks perform slightly worse than the traditional time-series ARIMA. Smith and Demetsky (1997) in a comparison among historical average, neural network, and nonparametric regression models for Woodrow Wilson Bridge, found that the neural network model suffered significantly higher errors.

Support Vector Machines, introduced by Vapnik (1995) is a family of learning algorithms, which is currently considered to be one of the most efficient methods for solving pattern recognition and regression problems in many applications. It possesses a good generalization capability, computational efficiency, and is very robust in high dimensions (Kecman, 2001). Due to its ability to outperform most other learning algorithms (Vert, 2001), it has been successfully applied to many applications, ranging from biology (Furey et al., 2000; Minh, 2004) to financial time-series analysis (Kim, 2003), etc. However, in traffic engineering in particular, with a few exceptions (Vanajakshi (2004), Vanajakshi and Rilett (2004) on prediction of travel time; Yuan and Cheu (2003), Jiangtao et al. (2002) on incident detection), to the best of this author’s knowledge at the time of writing, this could be the first attempt to investigate application potentials of the technique in traffic volume forecast.

6.2.2 Basic notations

Some basic notations are used in this study:

- *Prediction interval* (Δ): the time window from the current time, at the end of which prediction is made. This study explores a wide spectrum of prediction intervals: $\Delta = 5, 10, 15, 20, 25, 30$ and 60 minutes.
- *Rolling horizon* (Ω): the look-back time from the current time, within which data is used for prediction. In the first stage of this study $\Omega = 30$ minutes is used. To

study the effect of rolling horizon on prediction accuracy Ω is subsequently extended to 45 and 60 minutes. We denote $\Omega \rightarrow \Delta$ as a prediction with rolling horizon Ω and prediction interval Δ , for example $30 \rightarrow 5$, $30 \rightarrow 10$, etc.

- *Time-series prediction*: from the definitions of Δ and Ω , the time-series traffic prediction can be generalized as follows: Given the time window T , be it 24 hours or any time span in which prediction is considered, decompose into sub-windows Ω_s . Let δ denote the length of a time interval, $V(t)$ denote traffic volume at time t , and n denote the number of intervals in Ω . The objective of the time-series prediction is to find a function $f : R^{n+1} \rightarrow R$ such that

$$f(V(t-\Omega), \dots, V(t-\delta), V(t)) = V(t+\Delta); \forall t \in \{0, T-\Omega\} \quad (6.5)$$

6.2.3 Data and baseline predictors

The data used for prediction involves actual traffic volumes obtained on a 4-lane segment (80007766) with the length of 400 meters, between Adam and Kheam Hock roads, along the PIE in Singapore (Figure 6.7).

Figure 6.7: The study segment on PIE

As investigated from the HDB, the segment represents one of the most congested segments: the annual average daily traffic (AADT) amounts to 110,000 vehicles, while flow rates (veh/h) vary widely - from a few hundreds at nights to nearly ten thousands in peak periods. From the HDB, traffic volumes for the whole month of October 2003 are used for training and testing. For practical applications, the study investigates prediction performances over a wide range of prediction intervals, namely 5, 10, 15, 20, 25, 30, and 60 minutes. Data in a day are represented by a vector of 288 intervals of 5 minutes.

The study involves employing several data sets. Let S_1 denote the first set of traffic volumes for the whole month, exclusively classified into the learning (training) data S_1^L and test set S_1^T . S_1^L includes 24 hour-traffic volumes for 21 days from 1st to 26th, and S_1^T includes 5 separate test sets, one for each day, from 27th to 31st of October.

SVM and baseline predictors

(i) SVM

In this study, the SVR regression package LIBSVM 2.71 introduced by Chang and Lin (2001) is used for prediction. To avoid numerical difficulties during the calculation, the prediction starts with data scaling in both training and testing sets from large numeric ranges into smaller numeric ranges of $[+1, -1]$. Given the fact that traffic volume patterns are highly non-linear, the RBF kernel model is used for training since it nonlinearly maps samples into a higher dimensional space. In order to find the model parameters (C and γ) to optimise the training, the 5-fold cross-validation is used to avoid overfitting. The training set is subdivided into 5 subsets of equal size. Sequentially, one subset is tested using the classifier trained on the remaining subsets. The best parameters (C^* and γ^*) are used to train the whole training set, and is subsequently used to predict the test sets. The volume predicted by SVM method at time $t + \Delta$ is denoted as $V_{SVM}^p(t, \Delta)$.

(ii) Historical Mean Predictor (HMP)

The historical mean profiling is simply the series of the means of historical volumes in successive intervals. Since the test set includes data on weekdays, and there exist high correlation coefficients of traffic counts among weekdays in the segment (Section 4.3.1), the historical average is calculated from traffic volumes exclusively on weekdays in S_1^L . Let S_{1-H}^L denote the learning set for historical profiling method, D_H denote the number of days in S_{1-H}^L , and $\overline{V}_H^p(t, \Delta | S_{1-H}^L)$ denote the historical volume at time $t + \Delta$ given S_{1-H}^L . The HMP can be represented as Equation (6.6).

$$\overline{V}_H^p(t, \Delta | S_{1-H}^L) = \frac{1}{D_H} \sum_{d \in S_{1-H}^L} V(t, d) \quad (6.6)$$

(iii) Current-Time based Predictor (CTP)

The CTP predicts traffic volumes by simple projection of traffic volume at the current time t . Let $V_R^p(t, \Delta)$ denote the predicted volume at time $t + \Delta$, it follows:

$$V_R^p(t, \Delta) = V(t) \quad (6.7)$$

Since the CTP simply projects the current traffic volume to the future flow, it is likely that the CTP incurs high errors for long prediction intervals, especially under dynamic conditions. However, for this comparative study, the method is introduced to see the temporal effects on prediction performances.

(iv) Double Exponential Smoothing Predictor (ESP)

The ESP method weights past observations using exponentially decreasing weights: recent observations are given higher weights in forecasting than the older observations (Makridakis et al., 1983).

$$V_{SM}^p(t+1) = 2S'(t) - S''(t) + \left(\frac{\alpha}{1-\alpha} \right) (S'(t) - S''(t)) \quad (6.8)$$

$$S'(t) = \alpha V(t) + (1-\alpha)S'(t-1)$$

$$S''(t) = \alpha S'(t) + (1-\alpha)S''(t-1)$$

where: $V_{SM}^p(t+1)$: Predicted volume at the next interval

$V(t)$: Volume at the current time t

$S'(t)$: Single exponential smooth value at time t

$S''(t)$: Double exponential smooth at time t

α : Smoothing coefficient, $0 < \alpha < 1$.

Note that in Equation (6.8), t indicates the *index* of the current interval, not the calendar time like in the other predictors. For a prediction interval Δ equals m intervals, the prediction is made recursively m times, starting from $t+1$ to $t+m$.

Performance measures

Root Mean Square Error (RMSE) and Mean Absolute Percentage of Error (MAPE) are two statistics commonly used as principal performance measures. Let $V^a(t, \Delta)$ denote the actual volume, and $V_{(i)}^p(t, \Delta)$ denote the predicted volume at time $t + \Delta$, i.e. $V_{SVM}^p(t, \Delta)$, $\overline{V_H^p(t, \Delta)}$, $V_R^p(t, \Delta)$ or $V_{SM}^p(t, \Delta)$. By definition:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (V^a(t, \Delta) - V_{(i)}^p(t, \Delta))^2} \quad (6.9)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|V^a(t, \Delta) - V_{(i)}^p(t, \Delta)|}{V^a(t, \Delta)} \times 100 \quad (6.10)$$

where N is the number of observations. Since RMSE has a unit and varies in a large scale, it is difficult to quantify the magnitude of errors. On the other hand, MAPE, being unit-free and measured in relative term, is more convenient as a basis for comparison. Therefore, in this study, MAPE is used as the principal measure of prediction performance, whereas RMSE is used as a reference. MAPE is calculated for each day separately, and the parameter of interest is the mean of the MAPEs (or $\overline{MAPE}$) for all days in test sets. The MAPE (or MAPEs) used in the following sections are actually the $\overline{MAPE}$ values.

6.2.4 Overall predictive performance

The prediction with training set S_1^L uses the rolling horizon $\Omega = 30$ minutes to predict volumes for prediction intervals $\Delta = 5, 10, 15, 20, 25, 30$, and 60 minutes, respectively. Figure 6.8 presents the prediction errors with S_1^L .

Figure 6.8: Errors of the overall prediction by different methods

Figure 6.8 shows the general uptrend of the predictive errors as Δ increases. This may be due to the fact that the longer the prediction interval the less relevant the current volume is in reflecting the future condition, hence the predictability reduces with accumulated errors. The SVM outperforms the baseline predictors. The MAPEs by SVM for Δ within 15 minutes are less than 10%. Given the fact that traffic volumes normally fluctuate over time, this indicates that the SVM method performs satisfactorily for most short-term applications such as traffic control. On the other hand, the CTP works reasonably well for small Δ s, but deteriorates faster than the SVM and ESP for $\Delta > 15$ minutes. The ESP prediction errors are acceptable for Δ less than 10 minutes, but are worse than the HMP as Δ is greater than 25 minutes. As a whole, HMP is the worst performer, whose errors may be too high to be accepted for practical traffic control.

The prediction profiles by SVM with intervals $\Delta = 5$ and 30 minutes are presented in Figure 6.9. The SVM-5 is virtually indistinguishable with the actual pattern. Nevertheless, the prediction accuracy deteriorates with noticeable lags for $\Delta = 30$ minutes.

Figure 6.9: SVM-based prediction in comparison with actual data

Table 6.2 shows the prediction errors by SVM for individual test sets. The errors are bounded in small ranges, especially for $\Delta < 25$ minutes. Although the test days are all working days, Tuesday 28th seemed to have traffic patterns that varied widely, and MAPEs for this day are considerably and consistently higher than the others.

Table 6.2: MAPEs (%) by SVM for different test days

Δ	27-Oct Mon.	28-Oct Tue.	29-Oct Wed.	30-Oct Thu.	31-Oct Fri.	Average
5	2.39	2.45	2.37	2.49	2.24	2.39
10	4.75	5.29	5.24	4.93	4.79	5.00
15	9.05	9.18	8.86	8.44	10.85	9.28
20	10.09	12.24	11.37	10.88	10.14	10.94
25	11.11	13.93	12.67	11.98	11.43	12.23
30	12.30	15.62	13.87	12.61	12.39	13.36
60	17.89	22.95	20.15	17.95	17.79	19.34

6.2.5 Temporal behaviour

The prediction presented in Section 6.2.4 used the 24-hour data set, where traffic encountered different conditions. To explore how the predictors work under recurring congestion the analysis is carried out for peak periods. The data set S_2 for training and testing is the same as S_1 , ($S_2 \equiv S_1$). An inspection of data revealed that dates 1st, 2nd and 3rd experienced different congestion levels, hence these three days were selected for the second test sets S_2^T , and the remaining days of the month for the learning set S_2^L .

Figure 6.10 shows the prediction errors for AM peak, from 6 AM to 8 AM, based on S_2^L and S_2^T . The SVM predictor significantly outperforms its counterparts: By exploring a large searching space, the SVM algorithm allows a proper representation of repetitive traffic patterns to enhance its predictive power, whereas its counterparts do not possess this learning capability. In particular, the SVM even slightly outperforms the SVM prediction using S_1^L and S_1^T (Figure 6.8). The better performance of SVM during the peaks may be the result of a high level of deterministic component in the uptrend flow. The CTP and ESP performances deteriorate rapidly as Δ increases, though the rate of increase of the errors by ESP is less. The serious impairment of CTP can be conceptually explained: the CTP predicts future flows by simple projection of the current flow rate. It relies purely on the current traffic information. Therefore, under rapid growth of demand, the method inevitably exposes its poor behaviour: the higher the Δ , the higher error the CTP suffers. Similarly, although the ESP model employs two components to handle the trend (S' and S'' in Equation 6.8), it seems to perform less satisfactorily unless the trend is linear.

Figure 6.10: MAPEs of predictors on AM peak

The capability to follow the uptrend is reflected in Figure 6.11. The SVM follows the actual profile well, while the CTP and ESP lag behind the actual profile by some intervals.

Figure 6.11: Predicted traffic profiles in AM peak

6.2.6 Effect of rolling horizon

The predictive performances of different methods presented in Sections 6.2.4 and 6.2.5 are associated with rolling horizon of $\Omega = 30$ minutes. The SVM predictor proves its superiority over the other methods. It would be, however, useful to learn how SVM prediction performances change with Ω . To explore this, the first data set S_1^L and S_1^T is used: prediction made by SVM is conducted for the same set of $\Delta = 5, 10, 15, 20, 25, 30$, and 60 minutes, while Ω subsequently increases from 30 to 45 and 60 minutes.

Figure 6.12 plots the prediction errors for three values of Ω s over different values of Δ s. The figure shows the monotonous uptrend of MAPEs as Δ increases: as the prediction interval increases the prediction errors normally increase since for the higher Δ the future traffic patterns deviate farther away from the historical patterns in the rolling horizon that are used to predict future conditions. It can be seen that the rates of changes of MAPEs are higher for $\Delta < 15$ minutes, significantly decrease for the range $15 \leq \Delta \leq 20$, and are then remarkably lower in the range $20 \leq \Delta \leq 60$ minutes. The differences in the rate of increase in errors for various Δ s could be the result of random fluctuations in the data. For another data set, this phenomenon may be different.

Figure 6.12: Effect of rolling horizon on prediction accuracy

An important feature that can be observed from Figure 6.12 is that MAPEs decrease as Ω increases. The improvements are marginal for small Δs , but increased considerably for $\Delta = 30$ and 60 minutes (most beneficial for $\Delta = \Omega$). This finding is, however, in an opposite direction found by Ishak and Al-Deek (2002): in a study of the effect of rolling horizon on the errors of travel time prediction for the rolling horizon less than 30 minutes, the authors found that MAPEs tended to rise with Ω , and “the longer horizon may have adverse impact on prediction performance”. A closer look at the SVM theory may help explain why SVM prediction is improved in high dimensional predictions: SVM is a learning technique with the theoretical foundation from statistical learning theory and structural risk minimization. The predicted error by SVM method known as generalization error E_{gen} includes approximation error E_{app} and estimation error E_{est} (Kecman, 2001).

$$E_{gen} = E_{app} + E_{est} \quad (6.11)$$

The magnitudes of the two components depend on model complexity. A simple model (low Ω) may not have enough representational power, thus it may be biased with resulting higher E_{app} . However, such a model will be less dependent on training data, consequently it incurs lower E_{est} . By contrast, a higher order model (larger Ω) has a higher power to classify data more accurately, hence lower E_{app} but higher E_{est} . In other words, SVMs possess powerful learning capabilities in high dimensions, and with a reasonable training size, the higher dimension the prediction is, the more accurate the model in representing the data, at the expense of larger E_{est} . Therefore, there is a trade off between the two error components. In this experiment, when Ω increases from 30 to 45 and 60 minutes the model dimension increases from 7 to 10 and 13 accordingly, there is one point at which the decrease in E_{app} exceeded the increase in E_{est} , giving a rise in prediction accuracy. Given this trade off, there may be an optimal dimension (Ω^*) for a given training set.

6.2.7 Nearest Neighbour method for SVM training

Results from the experiments in the previous sections show that SVM performed better than the baseline predictors in most of prediction intervals and traffic conditions. However, the use of the complete training set would be computationally expensive. Critical issues regarding practical use of SVM involve the improvement of training speed for online applications.

A possible solution is that the data can be used to train the model offline, and then the trained set is used for online prediction. This sounds attractive, but the offline-trained prediction may have poor online performance if the training set has only few patterns representing the real-time condition. A crucial requirement is that for online application, the prediction should be flexible to respond efficiently and effectively to the current situation. As stated in Section 2.4.1, in the input data only *support vectors* give shape to the construction of hyperplanes. Training points with inactive constraints do not contribute to the solution. If these training points were removed, the same solution would have been obtained. This feature allows resolving the difficulty in handling large data sets by sparse training.

Efforts have been made in exploring various methods for the reduction of training size. Zhang et al. (2006) proposed a hybrid algorithm of SVM and k-Nearest Neighbour (kNN) method for visual category recognition. The basic idea of the algorithm is to find close neighbours to a query sample for training of a local support vector machine. The results showed that the hybrid SVM-kNN “has reasonable computational complexity both in training and at run time, and yields excellent results in practice”; Li et al. (2007) proposed a method that combines SVM and kNN, labelled as SVM-kNN, to construct a solar flare forecasting model. The test results indicated that the rate of correct predictions from the SVM-kNN is higher than that from the SVM and Neural Networks-based algorithms. Similar methods used or the improvement of SVM training can be found in Blanzieri and Bryl (2007), Sun and Cho (2006), and Wang and Xu (2004).

The aforementioned works have the same objective that minimizes the training time while maintain the prediction quality. The algorithms were differed from

application to application given that in classification problems (SVM) data are scattered in the searching space without any pattern, while in regression problems (SVR) data are stored in the form of time-series patterns. There are a number of ways to reduce training data, however NN could be well suited for locating similar time-series traffic patterns. Motivated by this, an empirical Nearest Neighbour method, denoted as NN-SVM, is introduced to improve the SVM training. The procedure presented below is for offline training. Issues concerning online training will be addressed in the subsequent discussion.

The k-Nearest Neighbour (k-NN) algorithm is a non-parametric technique that measures the closeness between members in the learning set and the test set by Euclidean distance (Oswald et al., 2001; Kim, 2003). The Euclidean distance between samples x and y is calculated as:

$$D(x, y) = \sqrt{\sum_{t=1}^N (f(x, t) - f(y, t))^2} \quad (6.12)$$

where $f(., t)$ represents value of attribute f at time t in sample x or y ; N is the number of instances in each sample. Since the Euclidean distance considers the square of the deviation, it tends to be biased towards extreme values. Given that in this study MAPE is the target measure of effectiveness, the relative distance metric by an empirical NN measure that has the same form as MAPE is proposed. Let d^L denote a day in the learning set, d^T denote a day in the test set, $V(t, d^L)$ denote volume at time t in day d^L , $V(t, d^T)$ denote volume at time t in day d^T , and $D(d^T, d^L)$ denote the average distance between d^T and d^L . The distance metric $D(d^T, d^L)$ is defined as follows:

$$D(d^T, d^L) = \frac{1}{N_{d^T}} \sum_{\substack{d^T \in S^T \\ d^L \in S^L}} \frac{|V(t, d^T) - V(t, d^L)|}{V(t, d^T)} \quad (6.13)$$

where N_{d^T} denotes the number of time intervals in day d^T .

It should be noted that to investigate the effectiveness of using NN method for SVM training Formulae (6.13) is only used to estimate the average distances between the vectors of traffic volumes in the historical data and the vector in the test data (one day constitutes one vector). This is a pre-processing step that locates similar vectors that will be used in the (NN) training, and is conducted *independently* before the (NN) training. Data in the (NN) training set and the test set are *mutually exclusive* as in the case the whole data set is used for the training. Given these, the independence between the training data and the test data is maintained.

To see the effectiveness of this method, the 15-minute aggregated data for the whole month of October is used, named as data set S_3 , which is decomposed into learning set and test sets in the same way as S_1 : the learning data S_3^L includes data from October 1st to October 26th, and the test data S_3^T includes 5 test sets for 5 days, from October 27th to October 31st. Let S_{3-all}^L denote the training set from the whole training data, S_{3-5NN}^L denote the 5 closest matches, and S_{3-5FN}^L denote the 5 farthest matches in the training set S_3^L . The three learning sets were trained and tested with the same test sets in S_3^T . Results are presented in Figure 6.13.

As can be seen from the figure, the prediction with S_{3-5NN}^L performs closely to the standard SVM using S_{3-all}^L while it allows a substantial reduction of training size (5 days as compared to 21 days). The average $\overline{MAPEs}$ over the five days are 11.78% and 12.12% for S_{3-all}^L and S_{3-5NN}^L , respectively. By contrast, the prediction made by S_{3-5FN}^L is considerably worse than its counterparts. This implies that a substantial decrease in training size in a “supervised” learning does not necessarily lead to a decline in prediction accuracy. Given a reasonable training size, it is the similarity of patterns between the training and testing data the crucial influencing factor that determines the prediction quality. If the training set contains sufficient amount of good data (support vectors) to construct hyperplanes and represent the input properly, the other data can be discarded (Vanajakshi and Rilett, 2004). Since SVM is a black-box system, for an arbitrary training set, the question of whether the

training set has sufficient amount of good data may be difficult to address precisely. Nevertheless, it is true that the number of support vectors increases if the similarity of patterns between the training and testing data increases, and this can be determined empirically by the distance between the two data sets through the Nearest Neighbour method.

Figure 6.13: MAPEs of NN-SVM with 60 → 15 prediction

The features of NN method are explored by addressing the question: whether the NN-SVM is workable under more dynamic and unexpected conditions like incidents and whether the similarity effect helps to enhance prediction under incidents? Ideally, these questions should be verified with actual data. However, since actual incident data is not rich enough and the quality of LTA's incident data does not warrant a prediction that requires high data resolution and complete incident attributes, the prediction is extended to extensive simulated incident conditions. The simulated segment has geometrical similarity to the actual site (segment 80007766), except for the length that is extended from 400 m to 1,000 m. Three levels of traffic demand corresponding to the night time, day time and AM peak, determined from a typical day in the HDB are created in $S_{traffic}$:

$$S_{traffic} = \{low_volume, medium_volume, high_volume\} \quad (6.14)$$

and three incident scenarios:

$$S_{incident} = \{no_incident, one_lane_closure, two_lane_closure\} \quad (6.15)$$

Given sets $S_{traffic}$ and $S_{incident}$, there are a combination of 9 scenarios simulated for 180 minutes each. The simulation time is decomposed into:

- Warm-up period: 60 minutes
- From 61st min. to 120th min.: no incident
- From 121st min. to 150th min.: incident (for the test scenario, from the time the incident starts to the time normal traffic is restored)
- From 151st min. to 180th min.: no incident

Each scenario is run 5 times with 5 different random number seeds until convergence is reach. Strictly speaking, the necessary number of repetitions should be estimated through an iterative process. However, due to the limited time for the high number of scenarios, the number of repetitions is empirically determined as 5 runs for each scenario. The random number seeds for the repetitions are consecutively assigned as 20, 30, 40, 50, and 60.

To examine the similarity effect of the NN-SVM method, data from the scenario (*high_volume-two_lane_closure*) are selected. Data from one random number is used for the test set, and the other 4 random numbers together with the remaining 8 scenarios are used for training. The characteristics of these 5 random numbers are that they have the same the traffic demand and incident scenario but vehicles are released randomly by the simulation generator, thus traffic patterns are similar but not identical. Data on traffic volumes is collected every 2 minutes. The parameters of prediction model are set as $\Omega = 10$ minutes, $\Delta = 2, 6$, and 10 minutes.

The concept of the aforementioned distance metric is applied to define the closeness between the test data and the training data, sorted by distances, then classified into three training sets: *Train-all* for the whole training set, *Train-NN* for the smallest distances (6 data sets), and *Train-FN* for the remaining 6 data sets. As can be

expected, cases of 4 random numbers in the training are included in *Train-NN* with relative distances from 0.04 to 0.06. The other scenarios have higher distance values, ranging from 0.27 to 0.52. Figure 6.14 shows the result of the NN-SVM prediction. Again, the prediction profile with *Train-NN* is close to that with *Train-all* for all prediction intervals. In reverse, the prediction with *Train-FN* incurs high errors, even with $\Delta = 2$ minutes.

Figure 6.14: NN-SVM with simulated incident data

Figure 6.15 plots the volume profiles for different training sets with $\Delta = 6$ minutes. The outcomes from *Train-NN* and *Train-all* are close together, while both of them exhibited some degree of smoothness compared to the simulated profile. On the other hand, the deviation of *Train-FN* is noticeably large. It seems that the *Train-FN* profile has no relevance to the simulated profile. The lack of similar patterns in *Train-FN* is probably the reason for its poor performance.

Figure 6.15: Predicted profiles by different training sets

Discussion: through using both actual and simulated data, this section has presented the merits of applying the empirical NN method to train and improve SVM prediction by locating days in the learning set with patterns closest to the day in the test set. The results are appealing since training with NN filter can be performed at a high speed. Let *speed-up factor* denote the ratio between the training time using all data and training time using the NN method, in the prediction $60 \rightarrow 15$ ($\Omega = 60$, $\Delta = 15$ minutes) the speed-up factor is about 60 (3 sec. versus 3 min.)¹ in an Intel Pentium 4 CPU 2.66 GHz for S_{3-5NN}^L (1,125 instances) versus S_{3-all}^L (4,725 instances). For a larger training set, this factor is probably greater, since the training time seems to increase exponentially with its size. The pre-processing time is short (typically within several tens of seconds), thus if it is included the total time is still considerably less than the training time of the complete training set. This is a good feature for online implementation, since it allows time saving without compromising the prediction quality.

Considering the trade-off between distance, training time and accuracy, it would be desirable to establish a *distance threshold* for a reduced training set. If the distance

¹ Approximated

threshold is too high then only a small portion of data is cut off, the prediction accuracy may improve marginally, but the advantage of fast training may not be significant. In contrast, too low a distance threshold enables fast training, but the sparse data provides a low representation power, leading to poorer performance. There is no universally established rule for the selection of distance threshold, since this depends on specific data and application.

The empirical NN method presented above is for offline training, and is applied to 24 hours. For online application, the method should be modified to adapt to real-time requirements. Recall that Ω denotes the rolling horizon, the NN method aims at retrieving patterns that are most similar to the data in Ω . Let $\vec{Z}^T(t, \Omega)$ and $\vec{Z}^L(t, \Omega)$ respectively denote vectors of volumes in the test data and in the learning data within the time window $(t, t - \Omega)$, δ denote the length of the rolling step, n denote the number of intervals in Ω , $V_{Z^T}(t)$ denote the traffic volume at time t in the test set, $V_{Z^L}(t)$ denote the traffic volume in the test data at time t in the learning set, and $D(\vec{Z}^T(t, \Omega), \vec{Z}^L(t, \Omega))$ denote the weighted sum of the relative differences in traffic volumes between the test set and the learning set. The distance metric between the test and the learning vectors can be expressed in Equation (6.16):

$$D(\vec{Z}^T(t, \Omega), \vec{Z}^L(t, \Omega)) = \frac{1}{n} \sum_{i=0}^{n-1} w_i \frac{|V_{Z^T}(t - \delta \times i) - V_{Z^L}(t - \delta \times i)|}{V_{Z^T}(t - \delta \times i)} \quad (6.16)$$

where w_i indicates the relative “weight” from each interval i in the rolling horizon, attributed to the total distance metric. Since the farther to the current time an instance in Ω is, the less relevant the traffic data in that interval is in reflecting the current and future traffic condition, the weights should be strictly decreased in Ω . In Rice and Zwet (2004), the distribution of weights is fitted by the Gaussian probability density function with mean zero and a variance σ^2 , which is specified by the model’s users.

The search for similar vectors for (NN) online training is a pre-processing step in each prediction interval, and is conducted in an *incremental* manner during which

the data are updated to the current time. The process moves forward with the rolling step in the rolling horizon procedure described in Peeta and Mahmassani (1995). As in case of offline training, this pre-processing step is *independent* from (NN) training, and data in the (NN) training set and the test set are mutually exclusive.

The concept of NN distance presented in Equations (6.13) and (6.16) is empirical and heuristic in nature. It mimics the form of MAPE to target the lowest prediction error. Alternative techniques of dissimilarity metric commonly used in the literature include Euclidean distance, square distance (Oswald et al., 2001), absolute distance and Windowed Nearest Neighbour (Rice and Zwet, 2001). The terminology “similarity”, being the inverse of “distance” can also be used. In essence, all of these techniques attempt to learn relevant information from a historic time window prior to the current time, and each has its own pros and cons.

6.2.8 Conclusion

This section presents an investigation of SVM applications for short-term traffic volume forecasting. The experiment conducted with actual data has been reinforced with simulated data. From the results and analyses, the following conclusions can be drawn given the described data and the road segment:

- (i) The SVM-based predictor in general outperforms the baseline predictors in almost all prediction intervals. It has good performance particularly for small Δs . For $\Delta = 60$ minutes, however, the use of historical data may be preferred (see Figure 6.8).
- (ii) The SVM significantly outperforms the baseline predictors under recurring congestion. Prediction under highly stochastic nature of non-recurring congestion can be improved if similar patterns can be found in the training set.
- (iii) The similarity of patterns between training and testing sets is of particular importance. SVM requires sufficient amount of similar patterns to represent the training data properly. On the other hand, the training size is not crucial,

providing that the learning set has enough support vectors to construct hyperplanes.

- (iv) The rolling horizon Ω has a positive effect on SVM's prediction accuracy. For short-term traffic prediction, the higher Ω the more accurate the prediction is, due to the capability of SVM in solving complex classification problems in a high dimensional space. It should be noted that this finding is obtained for the explored rolling horizon Ω in the range 30-60 minutes. It has not been verified for $\Omega > 60$ minutes.
- (v) SVM allows employing the NN method as a pre-processing step to reduce the size of large data sets to accelerate training, while maintaining prediction quality. On this basis, the expansion of the searching space by the NN method to enhance prediction accuracy becomes feasible. The method can also be employed to improve the SVM prediction in incident situations, as presented in Section 6.2.7. This is a distinct feature of SVM technique in knowledge-based system's applications in traffic flow forecasting.

6.3 SHORT-TERM PREDICTION OF TRAVEL TIME

6.3.1 Introduction

Reliable prediction of travel time is essential for successful implementation of ITS, especially for ATISs. The systems require real-time and predicted data to evaluate and monitor traffic and to disseminate advisory information to the travellers. For this reason, there has been extensive research on short-term travel time forecasting. Methodologies that have been commonly deployed for travel time forecasting include Kalman Filter (Chen and Chien, 2001; Chu et al., 2005), time-series ARIMA models (Williams, 2001), time-series nonlinear models (Al-Deek et al., 1998; Ishak and Al-Deek, 2002), time-varying coefficient (Zhang and Rice, 2003; Rice and Zwet, 2004), and neural network (Rilett and Park, 1999; Fu and Rilett, 2000; Lint et al., 2002). In the context of data input, the previous studies used either historical profiling or real-time data to predict future travel time (Chien and Kuchipudi, 2002;

Chun et al., 2003; Schrader et al., 2004). In the context of computing methods, prediction may be performed directly using travel time data obtained from traffic surveillance devices (Chien and Kuchipudi, 2003) or indirectly by imputing from other traffic variables (Nam and Drew, 1996; Roden, 1996; Al-Deek et al., 1998; Kwon et al., 2000). In brief, the previous prediction models have different levels of predictability and limits, and a great deal of effort is needed to look for an efficient instrument that achieves higher forecast accuracies.

In travel time prediction, however, only a few attempts were made in investigating the SVM technique: Vanajakshi (2004) was probably the first attempt that provides an initiative in investigation of the technique in this area. The author's results are encouraging despite the fact that the study used only a few simple baseline predictors for a limited data set. Particularly, more effort needs to be devoted to exploring the SVM behaviour in critical incident conditions that are most needed by ATISs. Motivated by this, this section investigates potential applications of SVM for time-series travel time forecasting through an extensive experiment with focus on the prediction performances under various traffic conditions and incident scenarios.

6.3.2 The study section

The study focuses on an extended segment of PIE of approximately 5.2 km long, consisting of 4 sub-segments with IDs 80007758, 80007762, 80007766, and 80007770, from the Bukit-Timah Expressway to Mount Pleasant road - an arterial road in Singapore (Figures 4.4 and 6.16). The analysis of traffic data in the year 2003 found that although the segment is equivalent to only 4% of the total lane-km, it accounted for approximately 20% the total delay of the expressway network. From the HDB, data on traffic speeds for the whole month of October 2003 is selected to estimate travel times, used for training and testing.

Since the studied segment consists of 4 sub-segments, travel time considered in this study will be path travel time. Let $x = 1, 2, 3, 4$ denote indices of sub-segments 80007758, 80007762, 80007766 and 80007770 respectively, from the upstream to

downstream. Let $L(x)$ denote the length of sub-segment x , $v(d, t, x)$ denote the speed at sub-segment x for time interval t on day d , and $T^*(d, t)$ denote the instantaneous path travel time of the trips departing the segment at time interval t on day d . The instantaneous path travel time $T^*(d, t)$ is estimated as follows:

$$T^*(d, t) = \sum_{x=1}^4 \frac{L(x)}{v(d, t, x)} \quad (6.17)$$

Figure 6.16: The study area

It is desirable to work with actual travel time that is estimated by tracking vehicle trajectories in space-time dimensions. However, the 15-minute aggregated data do not permit the transformation from instantaneous travel time to actual travel time. Therefore, in this section, the comparison of prediction is made on the basis of instantaneous travel time. The term “instantaneous travel time” is hereafter denoted as “travel time” for short.

Using Equation (6.17), the path travel time is computed for every time interval on every day. Let's denote S the set of travel times used in this experiment. Since the travel time variation is a key indicator of the reliability of a transportation network,

and is an important influencing factor of predictive performance, before going to select important statistical variables used in predictors, it is necessary to discuss characteristics of day-to-day travel time variation in the studied section.

6.3.3 Day-to-day travel time variation in the studied section

In statistics, with normal distribution, the arithmetic mean is commonly used to measure the central tendency, and the variance is commonly used to measure the spread of a distribution. In traffic engineering, most of the previous studies also used historical means to “predict” future travel times. However, if the distribution is highly skewed, the median may be preferred, and standard deviation should be supplemented in those cases by the semi-interquartile range.

Figure 6.17 plots travel time profiles of all days in data set *S*. The figure shows that the travel time distributions in most of intervals are highly positively skew with long tail in the right, especially during the peaks. Lint et al. (2005), in their study of one-year travel time distribution on the A20 freeway in The Netherlands also found that the day-to-day variation is highly skew.

Figure 6.17: Day-to-day variation of travel times in the studied segment

The deviation from symmetrical distribution is measured by the coefficient of *skewness* (Taylor et al., 2000) for each instance t in set S , defined by Equation (6.18):

$$\zeta_{sk}(t) = \frac{1}{N_d \times s^3(t)} \sum_{d \in S} \left(T^*(d, t) - \bar{T}^*(t) \right)^3 \quad (6.18)$$

where $\bar{T}^*(t)$ is the sample mean of historical travel times in instance t ; N_d is the number of days in set S , and s is the sample standard deviation. $\zeta_{sk}(t)$ takes a value of zero when the distribution is completely symmetrical (Figure 6.18a). A positive value indicates a skew with long tail in the right (Figure 6.18b).

Figure 6.18: Day-to-day travel time distribution pattern

In addition, the *spread* of the distribution is defined by the following empirical measure:

$$\psi_{spr}(t) = \frac{T_{85^{th}}(t) - T_{15^{th}}(t)}{T_{50^{th}}(t)} \quad (6.19)$$

where $T_{50^{th}}(t) = T_{median}(t)$ indicates the median value of travel times in each instance t . Equation (6.19) indicates the ratio of 70% of the distribution around the median, and the median travel time. The large value of $\psi_{spr}(t)$ indicates that the distribution is widely spread, travel time prediction will be more difficult and less reliable, and vice versa.

Figure 6.19 plots the values of the two aforementioned coefficients aggregated for every hour. The figure shows that the distribution of travel time is positively

skewed for all time periods. High skew values are notably observed for nighttime and early morning. This implies that during those periods, traffic is mostly in free-flow condition, though with some disturbances. On the other hand, a high value of $\psi_{spr}(t)$ represents a wide spread distribution with a significantly high frequency of heavy congestion. Intuitively, the highest spread values are observed during AM and PM peaks.

Figure 6.19: Skewness and spread of the travel time distribution

From this observation and analysis, it can be concluded that the use of historical mean as a travel time predictor may be highly biased since the mean may be strongly affected by extreme values (heavy congestion). Instead, the historical median that is less sensitive to extreme values than the mean and this makes the median a better measure of the central tendency for positively skew distributions. Some authors (Kwon et al., 2000; Lint et al, 2005) also used the historical median instead of the mean. This issue will be addressed in Sections 6.3.4 and 6.3.7.

An investigation of the HDB showed that traffic on October 1st and 3rd was in normal conditions, while October 2nd had a severe incident. To evaluate the effect of incident on prediction, the three days are used for three test sets (S_1^T, S_2^T , and S_3^T), and the remaining days of the month are used for the training set S^L .

6.3.4 Baseline predictors

(i) Historical Mean Predictor (HMP)

Since the day-to-day travel time distribution is highly skew, the use of historical mean for prediction of future travel time may be highly biased. However, to see the advantages of the alternative options, the historical mean is used as one of the baseline predictors. A comparison of performances among the Nearest Neighbour method, the mean, and the median will be made in Section 6.3.7.

The historical mean profiling is simply the series of the means of the historical travel times from individual intervals. Let D_H denote the number of days in S^L and $\bar{T}_H^p(t, \Delta | S^L)$ denote the average of historical travel times at time $t + \Delta$, given S^L .

$$\bar{T}_H^p(t, \Delta | S^L) = \frac{1}{D_H} \sum_{d \in S^L} T(t, d) \quad (6.20)$$

(ii) Current-Time based Predictor (CTP)

The CTP predicts the future travel time by projection of the current travel time. Let $T_C^p(t, \Delta)$ denote the predicted travel time for the prediction interval Δ by this method, it follows that:

$$T_C^p(t, \Delta) = T(t) \quad (6.21)$$

(iii) Time-Varying Coefficient (TVC)

In traffic engineering, the TVC model has been employed by Zhang and Rice (2003) as well as Rice and Zwet (2004) for travel time prediction. From practical observations, the authors proposed a linear relationship between the current instantaneous travel time $T^*(t)$ at time t and the future travel time $T(t, \Delta)$ at time $t + \Delta$:

$$T(t, \Delta) = \alpha(t, \Delta) + \beta(t, \Delta) \times T^*(t) + \varepsilon \quad (6.22)$$

where $\alpha(t, \Delta)$ and $\beta(t, \Delta)$ are coefficients being smoothed in t and Δ , and ε is the error term. The coefficients are estimated by minimizing the sum of the weighted square of errors within the rolling horizon Ω .

Considering the time window $(t, t - \Omega)$. Let δ denote the length of a time interval, $T(t)$ denote travel time at time t , and n denote the number of intervals in Ω . The sum of the weighted square of errors within the rolling horizon Ω can be defined as the Weighted Least Square minimization problem with the following objective function:

$$\sum_{i=0}^n [T(t - \delta \times i + \Delta) - \alpha(t, \Delta) - \beta(t, \Delta) \times T^*(t - \delta \times i)]^2 \times w(t - \delta \times i) \quad (6.23)$$

where $w(\cdot)$ denote the weight each interval contributes to the minimization problem. The weights may be fitted by either the uniform or exponential distributions. In this study, $w(\cdot)$ is a discrete distribution function, strictly decreasing in Ω due to the fact that the farther an interval in Ω is, the less relevant the travel time in this interval is in reflecting the current and future conditions. Let's denote the estimated values of coefficients from Equation (6.23) as $\hat{\alpha}(t, \Delta)$ and $\hat{\beta}(t, \Delta)$. Hence, the predicted travel time at time $t + \Delta$ is:

$$\hat{T}(t, \Delta) = \hat{\alpha}(t, \Delta) + \hat{\beta}(t, \Delta) \times T^*(t) \quad (6.24)$$

Zhang and Rice (2003) used the historical mean $\bar{T}_H^p(t, \Delta | S^L)$ as an independent variable to regress the predicted value. In their model, the coefficient $\alpha(t, \Delta)$ in Equation (6.24) is modified to capture the historical component:

$$\alpha(t, \Delta) = \alpha'(t, \Delta) + \bar{T}_H^p(t, \Delta | S^L) \quad (6.25)$$

Like $\alpha(t, \Delta)$, $\alpha'(t, \Delta)$ is a regression coefficient being smoothed in t and Δ .

Equation (6.22) then becomes:

$$T(t, \Delta) = \bar{T}_H^p(t, \Delta | S^L) + \alpha'(t, \Delta) + \beta(t, \Delta) \times T^*(t, \Delta) + \varepsilon \quad (6.26)$$

The purpose of inserting the historical mean into the regression problem in Zhang and Rice (2003) is to give a reference value as an estimator of the predicted travel time to regress the current travel time $T^*(t)$. If the reference value is an unbiased estimator of the predicted travel time, it provides a perfect regression. On the other hand, if the reference value deviates far away from the “would-be” actual observation, the regression suffers high errors. In Figure 6.17 for example, it seems that the historical mean profile does not provide a good estimate for any particular day. This implies that the historical mean is a poor estimator of the actual values. Considering the form of day-to-day travel time distribution, as discussed in Section 6.3.3, either the median value or the NN value should be in place of the historical mean in Equation (6.26). The median can be determined as directly as the mean. The concept of NN used in this section was introduced earlier in Section 6.2.7. It should be noted that the recommended use of the median instead of the mean does not imply that the median is an unbiased estimator of the predicted travel time, but it could be a better measure of the central tendency than the mean for day-to-day travel time distributions, which are positively skew.

6.3.5 Overall predictive performance

Figure 6.20 plots the prediction results from SVM and baseline predictors for prediction intervals $\Delta = 15, 30, 45$, and 60 minutes, respectively. The general upward trend shows that the prediction errors increase with Δ . As the prediction interval increases, errors accumulate from an interval to another, giving a rise in prediction error. The flatness of the curves implies that unlike traffic volume, the prediction errors do not change rapidly with time. Among the four predictors, SVM consistently provided the best performance, followed by TVC and CTP. The HMP performed the worst with MAPEs by about 11.1%, represented by a straight line since the sets of historical data for different Δ values are nearly the same.

Figure 6.20: Overall performances of predictors

Table 6.3 shows the prediction errors of all predictors for each test set. As stated in Section 6.3.3, the 1st and 3rd days are incident-free, while the 2nd day has an incident. To evaluate the predictive performance under normal condition, the average of errors from the 1st and 3rd days is computed, whereas the 2nd day is shown for comparison. The table shows a strong effect of incident congestion on prediction errors. There is a notable discrepancy between averages of errors of the normal days and the incident day, with respect to MAPE. Regarding RMSEs, the gaps between the normal days and the incident day are even larger. Since the RMSE considers the square of errors, it may be highly biased towards extreme values. This implies that the effect of incidents may become too strong if the RMSE is used.

Table 6.3: Prediction results from different test sets

(MAPE in %; RMSE in sec.)

Δ	Test set	SVM		TVC		CTP		HMP	
		MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
$\Delta = 15$ min.	S_1^T	5.16	21.56	6.31	24.73	6.98	30.54	10.56	32.85
	S_2^T	7.38	73.90	7.96	76.04	7.79	75.14	12.53	75.20
	S_3^T	5.20	20.53	6.36	20.59	6.79	20.50	10.53	29.17
	Average S_1^T & S_2^T	5.18	21.04	6.33	22.66	6.89	25.52	10.54	31.01
$\Delta = 30$ min.	S_1^T	5.39	23.27	6.95	26.27	7.36	29.50	10.56	32.85
	S_2^T	8.24	75.89	8.75	78.72	9.25	81.02	12.53	75.20
	S_3^T	5.71	19.38	6.40	20.35	7.21	22.67	10.53	29.17
	Average S_1^T & S_2^T	5.55	21.32	6.68	23.31	7.29	26.08	10.54	31.01
$\Delta = 45$ min.	S_1^T	5.95	25.43	7.58	27.50	8.06	30.63	10.56	32.85
	S_2^T	9.05	78.16	9.38	83.12	11.21	87.54	12.53	75.20
	S_3^T	6.11	19.32	7.34	23.05	7.91	24.08	10.53	29.17
	Average S_1^T & S_2^T	6.03	22.37	7.46	25.27	7.99	27.36	10.54	31.01
$\Delta = 60$ min.	S_1^T	6.88	26.05	8.43	26.00	9.79	31.03	10.56	32.85
	S_2^T	9.85	81.06	10.72	85.94	13.45	94.71	12.53	75.20
	S_3^T	6.48	19.36	8.05	26.52	8.65	26.20	10.53	29.17
	Average S_1^T & S_2^T	6.68	22.71	8.24	26.26	9.22	28.61	10.54	31.01

Table 6.4 shows a comparison of prediction made by the historical mean versus the historical median from S^L . The result shows significant improvements in prediction accuracies if the median is used instead of the mean in representing historical data.

Table 6.4: Comparison between the historical mean and median

Test set	MAPE		RMSE	
	Median	Mean	Median	Mean
S_1^T	6.55	10.56	26.54	32.85
S_2^T	10.09	12.53	55.92	75.20
S_3^T	7.06	10.53	21.75	29.17
Avg. S_1^T & S_2^T	6.81	10.54	24.15	31.01

Figure 6.21 presents the predicted profiles for S_1^T , with $\Delta = 15$ minutes. Although no incident is reported on the 1st October, there are several disturbances at 5:15 AM and 3:30 PM. The figure shows that SVM follows the actual profile relatively

closely, especially under “stationary” condition. The TVC seems to deviate a little farther away. On appearance, the CTP has a perfect shape like the actual profile, but subjected to a time lag of 15 minutes, leading to high errors. The HMP, except for free-flow condition, incurs unacceptably high errors during most of the daytime.

Figure 6.21: Profiles of predicted travel times from different predictors

To see how the prediction errors vary with changes in travel time, in Figure 6.22 the MAPEs are plotted with the actual travel time profile. It can be seen that the shapes of the errors and the observed travel times have a high degree of resemblance: the predictors performed reasonably well as the travel time change less, and encountered high errors as travel time fluctuates rapidly and randomly.

The distribution of average percentage of errors (APE) from all test sets is presented in Figure 6.23. The figure shows that with three candidates - SVM, TVC and CTP, the proportions of APE 0-5% are greater than 50%. The SVM works better with proportion of error 0-5% amounted to nearly 70%, whereas proportion of errors greater than 10% is as low as 14%. At the other extreme, 50% of the prediction intervals by the HMP have errors greater than 10%. This statistics show that albeit high errors locally, SVM generally provides good performances.

Figure 6.22: MAPEs in association with the observed travel time

Figure 6.23: Proportion of average percentages of errors

6.3.6 Prediction behaviour in congested conditions

To explore how the predictors work under recurring congestion, a detailed analysis of prediction results for the test set S_3^T (which has a recurring congestion) with $\Delta =$

15 minutes is carried out. Figure 6.24 shows the predicted profile for the period from 4:30 PM to 8:00 PM. It can be seen that the alternative predictors all performed satisfactorily with MAPEs of 3.73%, 4.11%, 5.77%, and 6.73% for SVM, CTP, TVC and HMP respectively. This indicates that the prediction performances do not necessarily deteriorate at peak times, for this case in particular. This is probably true when the upward trend corresponding to a growth of traffic demand from free flow into a steady state that is represented with a high proportion of deterministic component. This should not be generalized for severe congestions whose travel times usually fluctuate exceptionally, thus the predictability will certainly be impaired.

Figure 6.24: Predicted profiles for recurring congestion

As can be seen from Figure 6.17, random fluctuations may be the primary cause of highly erroneous prediction. This is explored by zooming in an accident in S_2^T . The accident occurred in link 80007766 from 1:00 PM and was cleared at 2:10 PM. The one-lane closure greatly reduced the road capacity, leading to a tremendous increase in travel time during that period. Since the historical profile consists of the means of travel times in the past that have no relevance to the current incident condition (data in the test and training sets are mutually exclusive), the historical profile is not a

reliable tool for predicting travel time under incident conditions, hence the HMP is excluded from the prediction described below:

Figure 6.25 plots the predicted profiles by SVM, CTP and TVC for $\Delta = 15$ minutes. The figure shows that all predicted profiles significantly lag behind the actual profile. The CTP seems to have the same shape as the actual data, unfortunately was shifted with time delay of 15 minutes, but the SVM profile lags even longer in the incident period. The MAPEs for the period from 1.00 PM to 3.00 PM (post-incident inclusive) are 25.47%, 24.13%, and 25.52% for SVM, CTP and TVC respectively. Clearly, these indicate that all predictors failed to respond effectively to the incident.

Figure 6.25: Prediction under incident condition

The reason could primarily be due to the high level of data aggregation. SVM and TVC predict the future state based on data from a short time window of rolling horizon, whereas a high level of data aggregation may not be responsive enough to signify evidences of random fluctuations as soon as incidents start. Consequently, the predicted patterns lagged far behind the actual pattern, and the predictability seriously deteriorated. A better data resolution may help resolve this problem by updating data of the most current time in shorter intervals so that part of the data reflect incident occurrence. From this consideration, the question “whether a higher

data resolution is more helpful” will be addressed by investigating travel time under simulated incident conditions from the simulation presented in Section 6.2.7.

To explore how the similar patterns in the training affect the SVM prediction, the scenario: (*high_volume-two_lane_closure*) simulated with 5 random numbers was selected. Data from one random number is used as the test set, and the other four random numbers, together with the remaining 8 scenarios are used for the training. Simulated travel time is collected every 2 minutes. The parameters of prediction model are set as $\Omega = 10$ minutes, $\Delta = 2$ minutes.

Figure 6.26 illustrates the SVM predicted profile versus the simulated travel time. The figure shows that SVM outcome follows the simulated data closely, even at the incident period. The MAPE is as low as 3.83%. This high level of accuracy raises some questions: whether this improvement is due to the higher data resolution or due to similar patterns in the training data, and whether the training size or data quality is the determinant factor of the SVM’s prediction quality? These issues are clarified in Section 6.3.7.

Figure 6.26: SVM prediction in the simulated incident

6.3.7 Nearest Neighbour method for SVM training

As in the case of traffic flow forecasting, in travel time prediction an important practical concern is how to sparsely control the training data of large size without reducing generalization performance. This issue is addressed by applying the NN-SVM concept presented in Section 6.2.7 with simulated training set presented in Section 6.3.6. As described, the data from 8 scenarios and 4 random numbers in the 9th scenario forms the training data with 12 samples, while data from the 5th random number in the 9th scenario forms the test sample. Equation (6.13) is used as a general form, where traffic volume is replaced by travel time, to compute average distances between the test sample and each train sample. The same as traffic flow forecasting, this is a pre-processing step to identify close patterns for the (NN) training. The step is conducted *independently* from the (NN) training, and the data in the (NN) training set and the test set are *mutually exclusive*. The introduction of the NN method in this section is to investigate its merit for SVM (offline) training. For online applications, the process is conducted in an *incremental* manner during which data are updated to the current time in each prediction interval.

Let $D(T, L(i))$ denote the average distance between the test sample T and a train sample $L(i)$. The train samples i are decomposed into three training sets corresponding to values called *distance thresholds*: The first training set used all data, named as *SVM-Train- All*, whose result is presented in Section 6.3.6. The 2nd training set, *SVM-D1*, involves samples in the range $0 \leq D(T, L(i)) \leq 0.2$, and the 3rd training set *SVM-D2*, in the range $0.2 < D(T, L(i)) \leq 0.4$. The values of 0.2 and 0.4 are selected so that the training data are equally divided for the training sets *SVM-D1* and *SVM-D2* (the value of 0.4 is not really a threshold since it is greater than the maximum distance of 0.38). As can be expected, 4 random numbers in the 9th scenario were included in *SVM-D1* with the average distances range between 0.05 and 0.16, while those in *SVM-D2* range between 0.25 and 0.38. All three sets are trained and tested with the same test set.

Predicted profiles are plotted in Figure 6.27. The figure shows that *SVM-D1* profile is close to the simulated profile. The MAPEs for *SVM-Train All*, *SVM-D1*, and

SVM-D2 are 3.83%, 3.7% and 12.1%, respectively. This indicates that both *SVM-Train All* and *SVM-D1* perform well, and *SVM-D1* was even slightly better than *SVM-Train All*. The reason could be due to the fact that *SVM-D1* employed solely “good” data, (see Section 6.2.7) so the system constructed hyperplanes more properly. By contrast, *SVM-Train All* used both “good” and “bad” data. Since the data are relatively sparse, the number of support vectors is not sufficient to represent the test data, while the generalization power can be impaired by the bad data. This implies that the training size is not as crucial as data quality.

Figure 6.27: Effects of distance threshold on training

Discussion: the above example presents the advantages of the NN method as a pre-processing step to curtail the size of training data to accelerate training speed. The speed-up factor typically ranges from 10 to 60, depending on their sizes. This is appealing since it allows time saving without compromising the prediction accuracy.

Refer to the discussion about the effectiveness of the NN approach on prediction quality in Section 6.3.4: the TVC prediction is extended by alternatively using the mean and the median values instead of the NN data. The outcomes are presented in Table 6.5.

Table 6.5: TVC prediction using NN, mean, and median values

Δ	S_1^T			S_2^T			S_3^T		
	NN	Median	Mean	NN	Median	Mean	NN	Median	Mean
$\Delta = 15$ min.	6.31	7.86	11.59	7.96	10.63	13.25	6.36	7.25	11.26
$\Delta = 30$ min.	6.95	8.41	11.84	8.75	10.85	13.07	6.40	7.54	11.08
$\Delta = 45$ min.	7.58	9.02	12.16	9.38	11.28	13.26	7.34	7.89	11.23
$\Delta = 60$ min.	8.43	9.47	12.68	10.72	11.54	13.52	8.05	9.34	11.47

The table shows consistent improvements of using NN values over using historical mean as adopted by previous studies in the literature. The median also proved its superiority to the mean in representing the historical data for the TVC regression. This could, however, be case specific, and no unique conclusion should be drawn.

As can be seen in Figure 6.28, the profiles indicate that most regions of three sets lay more closely to the median than the mean, hence the prediction accuracy with the median improved. In other cases, the use of the historical mean may be better. From this observation it can be concluded that although the median provides a better indicator than the mean for most of historical profiles in travel time prediction, the use of either the mean or the median depends on the actual travel time profile of the day being considered. In either case, the NN method provides better performance compared to the mean and the median. These findings are, however, drawn from the prediction with aforementioned prediction parameters and the given traffic data. They have not been verified for other prediction parameters or data sets.

Figure 6.28: Profiles of the data in test sets in relation to the mean and median

6.3.8 Conclusion

Section 6.3 presents an investigation of Support Vector Regression (SVR) technique for travel time forecasting. The prediction performances by different techniques with actual data have shown that SVM-based predictor significantly outperforms the baseline predictors in various traffic conditions and intervals. SVM provides good performance especially for small Δ s. It also performs well for peak periods. Through simulation, it is found that if data resolution is reasonably high (Section 6.3.7), and similar patterns can be found in the training data, the “unpredictable” nature of incidents can be partly resolved to achieve reasonable prediction accuracies.

It should be noted that there exist no traffic patterns that are identical. The similarity between two patterns is evaluated by the “closeness” between corresponding members in the patterns using the relative distance metric by the empirical Nearest Neighbour (Section 6.2.7, Equation 6.13). Two patterns are considered similar if the distance between them is small. In addition, it is desirable to work with a high level of data resolution since the higher resolution the more promptly the actual data are

captured. However, high-resolution data impose technological challenges to the data collection/transmission and incident detection systems. The minimum level of resolution to achieve a specific accuracy may be site-specific, but can be determined empirically depending on incident severity, background traffic, distance to the upstream detector, and traffic control method, etc.

The section also presented the benefit of the empirical NN method in retrieving similar patterns for reduced data set to accelerate the training while maintaining prediction quality, adaptable for online implementation. Particularly, the use of the NN value instead of the mean in the TVC method provides significant improvements in prediction accuracies. By the comparison between the NN method, the mean and the median, it is found that in this experiment, for travel time prediction the historical median is generally better than the mean in representing historical data (Tables 6.4 and 6.5).

From an algorithm perspective, SVM possesses good properties to resolve critical problems encountered in neural networks such as over-fitting, curse of dimensionality and local minima. Therefore, although SVM and neural networks have many resemblances with respect to theoretical foundation, SVM typically outperforms the traditional neural nets, as is reviewed in literature. In addition, the use of SVM requires no assumption about the underlying distribution patterns of the data being manipulated.

6.4 ANTICIPATION OF TREND OF TRAFFIC CONDITIONS

Having the traffic parameters been predicted by SVM-based methods, the MS-FLC anticipates the evolution of traffic conditions using the congestion level estimated in the first stage, the risk factor, and the predicted traffic demand. The evolution of traffic trend depends heavily on the balance between traffic demand and supply, represented by the ratio of the traffic volume (V) upstream and the capacity remaining (C^*) at the incident location, which is the available road capacity in

normal condition minuses the capacity reduction ($CapR$). Therefore, the V/C^* ratio is used as a principal parameter for the anticipation.

The V/C^* (or *adjusted* V/C^*) ratio is represented by four fuzzy sets (Figure 6.29):

$$V/C^* = \{Low, Medium, High, Very_high\} \quad (6.27)$$

Figure 6.29: Membership functions for (adjusted) V/C^* ratio

The process to infer the anticipated congestion level involves two sub-stages: First, the predicted V/C^* ratio and the *Risk* are used to estimate the *adjusted* V/C^* , then the *adjusted* V/C^* and the current congestion level is used to infer the predicted congestion level. The division of the process into sub-stages has advantages that the rule base is simplified and the number of rules can be cut off. For example, in the case of the single stage, where all three inputs (congestion level, the risk factor, and the predicted V/C^* ratio) are used, the number of rules will be $4 \times 4 \times 4 = 64$ rules (the heavy congestion category is not included in this stage). By dividing into sub-stages, the number of rules is reduced to $4 \times 4 + 4 \times 4 = 32$ rules. Although the processing time in a sequential manner is marginally higher than the processing time in parallel for the same number of rules, the reduction of the number of rules is significant in saving the computational time, and makes the rule flow more comprehensible.

Figure 6.30 describes the FIS architecture for the 1st sub-stage in the reasoning process that derives the predicted congestion level. The risk factor and (predicted) V/C^* ratio are the two inputs, used to make adjustment for the V/C^* ratio. The inference engine is of Mamdani type.

Figure 6.30: FIS architecture for the 1st sub-stage

Given the fuzzy sets of the two input variables in Figures 6.6 and 6.29, a template set of 16 rules for the 1st sub-stage is proposed, as shown in Table 6.6.

Table 6.6: Rule templates for the 1st sub-stage

Rule No.	Rule condition		Rule conclusion
	predicted V/C^*	Risk	adjusted V/C^*
1	Low	Low	Low
2	Medium	Low	Low
3	High	Low	Medium
4	Very_high	Low	High
5	Low	Medium	Low
6	Medium	Medium	Medium
7	High	Medium	High
8	Very_high	Medium	Very_high
9	Low	High	Medium
10	Medium	High	High
11	High	High	Very_high
12	Very_high	High	Very_high
13	Low	Very_high	High
14	Medium	Very_high	Very_high
15	High	Very_high	Very_high
16	Very_high	Very_high	Very_high

Figure 6.31 describes the architecture of the FIS for the 2nd sub-stage. The congestion level supplied by the 1st stage of the MS-FLC and the adjusted V/C^* ratio obtained from the above rules are the two inputs used to infer the anticipated congestion level. The inference engine is also of Mamdani type.

Figure 6.31: FIS architecture for the 2nd sub-stage

Given the fuzzy sets of the two input variables in Figures 5.13 and 6.29, a collection of 16 template rules for the 2nd sub-stage is proposed in the FIS system (Table 6.7). Note that in this sub-stage, the variable *CongestionLevel* indicates the prevailing current congestion level, which does not include the *Heavy* congestion level since it is tracked directly from the 1st into the 3rd stage of the MS-FLC.

Table 6.7: Rule templates for the 2nd sub-stage

Rule No.	Rule condition		Rule conclusion
	adjusted V/C^*	CongestionLevel	predicted-CongestionLevel
1	Low	Free-flow	Free-flow
2	Medium	Free-flow	Free-flow
3	High	Free-flow	Pre-con
4	Very_high	Free-flow	Light
5	Low	Pre-con	Free-flow
6	Medium	Pre-con	Pre-con
7	High	Pre-con	Light
8	Very_high	Pre-con	Moderate
9	Low	Light	Pre-con
10	Medium	Light	Light
11	High	Light	Moderate
12	Very_high	Light	Heavy
13	Low	Moderate	Light
14	Medium	Moderate	Moderate
15	High	Moderate	Heavy
16	Very_high	Moderate	Heavy

Discussion: The above section presents the rule formulation in the two sub-stages. In the first sub-stage both quantitative measurement (V/C^*) and qualitative estimation (*Risk* factor) are employed, where the *Risk* factor is used to adjust the

predicted V/C^* ratio. The *Risk* factor accommodates potential uncertain elements in the traffic environment and is used as a scale. It is classified into 4 predicates with the basic idea: *Low* risk levels adjust the V/C^* ratio down; the *Medium* keeps the ratio the same, and the *High* and *Very_high* risks scale the V/C^* up. As presented earlier, the use of the risk factor is to cater for external risks that exist exogenously with the prediction being conducted. If data measurement and prediction were highly accurate, and if every parameter under consideration were known *a priori*, the use of quantitative measurements would suffice, and there is no need for the adjustment.

In the 2nd sub-stage, the *adjusted* V/C^* and the current congestion level are used to infer the anticipated congestion level for the next control step. Obviously, the anticipated traffic condition depends not only on the current condition, but also on the trend and the rate of change of the state, and the prediction interval. The current congestion level is used as the baseline for reference, whereas the *adjusted* V/C^* is used as the determinant that orientates the *trend* toward which the current traffic condition evolves as well as the *rate of change* of traffic state. The rate of change in this case does not refer to a quantifiable measure, but a fairly abstract attribute, whose magnitude depends on both the V/C^* value as well as the baseline condition. It should be reminded that the volume V indicates the traffic demand (that can be measured far upstream) but not the measured flow rate in the immediate upstream of the incident. Therefore, the ratio does not incur the double effect as in the fundamental *measured* speed-volume relationship (for example, a low value of flow rate indicates either free-flow traffic or heavy congestion). It is likely that *Low* V/C^* is associated with *Free-flow*; *Medium* V/C^* is associated with *Pre-congestion*; *High* V/C^* is associated with *Light* or *Moderate*; and *Heavy* congestion level can be expected at *Very_high* V/C^* .

In mapping of two state variables ($\mathfrak{R}^2 \rightarrow \mathfrak{R}^1$ mapping), there are some overlaps of rule conclusions (for example rules 1 and 2 in both Tables 6.6 and 6.7). However the overlaps do not destroy the consistent principle of rule mapping: different inputs may produce the same output, but there is no case that different outputs are produced from the same set of inputs. In addition, it should be noted that the same fuzzy output in different conclusions does not deliver the same fuzzy membership values. For example, the value "Low" in the conclusions of rules 1 and 2 in Table 6.6 produces different values of membership functions in the two rules.

The two aforementioned rule templates are proposed in line with the above concepts with a short prediction interval and control step. They can be varied: with short prediction intervals the anticipated congestion level will be closer to the current congestion level; by contrast, with long prediction intervals the anticipated congestion level will be highly governed by the V/C^* ratio.

Although the rule formulation seems to be highly intuitive, it reflects attempt to confront with uncertain problems in daily control practices to foresee what might happen in future, since there exists no mathematical equation for such a complex and multivariable problem.

Alternatively, the anticipated congestion level can be inferred from predicted traffic speed and density, in the same manner as is used to evaluate the current congestion level. However, as traffic during incident changes unexpectedly, this direct inference without referring to the prevailing current congestion may be highly risky.

Figure 6.32 illustrates the surfaces of knowledge for the two sub-stages, which are obtained using the *centroid* defuzzification method. The figure represents the projection of a hyperspace of knowledge into one three-dimensional space.

a) The 1st sub-stage

b) The 2nd sub-stage

Figure 6.32: Knowledge surfaces for two sub-stages

6.5 SUMMARY

This chapter presents the second stage that involves the anticipation of short-term traffic and incident conditions. Since traffic under incident is less predictable, this chapter addresses a challenging issue in traffic forecasting.

In the prediction of traffic volume, one-month traffic data on a segment of PIE is used for training and testing. Three baseline predictors are employed for comparison, including the HMP, the CTP and the ESP. The results show that SVM in general outperforms the baseline predictors for a number of prediction intervals. The SVM is also the best performer in recurrent congestion. In the investigation of the effects of rolling horizon on prediction accuracy, the results show a consistent improvement as rolling horizon increases within time windows 30-60 minutes. In particular, the learning capability of SVM in pattern recognition induces the application of the empirical NN method that retrieve the most similar traffic patterns for the model training. The method facilitates a substantial reduction of learning size to accelerate training without compromising prediction accuracy. This feature is not available in the baseline predictors.

In the prediction of travel time, the baseline predictors include the HMP, the CTP, and the TVC. The results show that SVM in general outperforms the baseline predictors over a wide range of prediction intervals. The empirical NN method is

also introduced to retrieve patterns that are close to the test vector for SVM training. As in the case of traffic volume forecast, the results show that the method allows a large reduction of training size to accelerate online training. In particular, the NN method allows an expansion of the searching space to improve the prediction accuracy for incident situation, as presented in Sections 6.2.7 and 6.3.6.

The predicted traffic data are used in the MS-FLC for prediction of congestion trend. Results from the automated forecast are subjected to further treatment inducing a risk factor to cater for unknown and unexpected impacts from traffic environment, including the effectiveness of the control scheme being implemented, incident severity, and the time of day.

Although promising results have been achieved from the SVM prediction, there are a number of issues that should be pointed out:

Firstly, the out-performance of the SVM predictor was achieved for the given data sets and sites. Although one-month data may be of medium size, the experiment is conducted only for a single location (400-m segment in traffic volume prediction and 5.2-km segment in travel time forecast). In other words, the superiority of SVM over its counterparts has not been verified for other datasets and locations.

Secondly, the concept of similarity between traffic patterns was reinforced with simulated incident data. The pattern similarity found in the NN method with simulated incident data used for SVM training (Sections 6.2.7 and 6.3.7) may represent an ideal condition of the similarity between traffic patterns. The concept of “pattern recognition” may be more applicable to events of recognisable patterns such as recurring congestion. Incident events are random in nature, thus this functionality of SVM will decline if the incident data are poor. In reality, it is hard to find a good collection of incident data, thus it is hard to find sufficient amount of close patterns that represent an incident condition. The possible solution to overcome this could be the use of high-resolution data so that the data in the rolling horizon become more responsive in capturing the incident occurrence, as is presented in Section 6.3.6.

Thirdly, an important issue that should be discussed regarding the requirement of data: SVM can explore a high amount of data that are not only in the immediate past, but also in the long historical storage. This data-intensive orientation could be a strong point but could also be a drawback because an adequate historical storage is required to provide a proper representation of training data.

Lastly, the Kernel parameters (C^* and γ^*) for non-linear mapping are normally empirically determined by the user, and only sub-optimal results are obtained.

CHAPTER 7 CONTROL ACTIONS

7.1 INTRODUCTION

The recommendation of control strategies and control actions is the last stage of the MS-FLC. The stage receives the evaluated and predicted traffic conditions from previous stages, and other traffic and incident information to supply recommended solutions. Since the traffic management and control under incident situations is very complex, dynamic and critical, the timely provision of appropriate recommended solutions is essential for efficient expressway operation management with effective allocation and utilization of available resources.

The expressway operation management during incidents undertakes important tasks, including the dissemination of prevailing information to motorists, the regulation of ramp access, the control of route diversion, and the management of queues. The tasks employ appropriate control measures to target the following primary objectives:

- Minimize mainline congestion
- Maximize throughput
- Minimize the total travel time (TTT) and total time spent (TTS) in the network.

Subjected to constraints:

- Prevent excessive ramp queues
- Prevent secondary queues at ramp merges
- Prevent severe congestion on the alternative routes due to traffic diversion.

It is possible that the above objectives are conflicting and constraints cannot be all satisfied. For example, the objective to minimize mainline congestion may induce restrictive metering rates and formation of ramp queues, which may result in an increase in TTS in the network, violating the third objective concerning the TTS. In

an analogous way, the constraint that prevents secondary queues during heavy mainline congestion entails a strict regulation of ramp flow. Therefore, there should be a reasonable balance between control objectives providing that the constraints are strictly observed.

Traffic control during incidents represents a complex and multivariable problem. Ideally, the control program should be established to target optimum solutions. However, as commented in Ben-Akiva et al. (2003), “*the reliability of analytical solutions for most traffic engineering problems is almost always questionable due to a large number of assumptions made in the problem formulation*”. The analytical approach, apart from its complexities in formulating problem solutions, can hardly represent properly the actual traffic congestion due to the uncertainty in evaluating the current situation as well as the lack of knowledge in prediction of traffic variables. Attempts to optimise the problem for very precise solutions in the absence of precise input data and information is something like “*penny wise but pound foolish*”.

Under such circumstances, a sub-optimal solution is often considered a good-enough solution, and hence the development of sophisticated optimised control programs lies outside the scope of this research. This research attempts to provide a more realistic local ramp control procedure following fuzzy logic approach with some extension to the coordinated and integrated controls at the strategic level.

The control decisions in the 3rd stage are made on the basis of prevailing evaluated information from stage 1 and short-term predicted information from stage 2. Strategies associating with these two types of information are known as *reactive control*, and *proactive control*, respectively. Naturally, the proactive control is preferred to the reactive control for a number of reasons: In the first place, there is a time lag from data collection and analysis to control implementation. Therefore, the reactive control reacts to outdated traffic information. By the time the decision is made the traffic states may have already changed and new problems may have arisen, hence the efficiency of recommended solutions declined. Second, reactive control with current information may induce side effects such as overreaction in alternative diversion routes. The proactive control, on the other hand, reacts to near-

term anticipated conditions that are more relevant to the prevailing conditions. From the user's perspective, the anticipated information provides the drivers opportunities to select or switch paths before reaching the incident location. From the control perspective, the anticipated information allows sufficient time envelopes for setting or resetting control parameters. Hence, the full benefits of ITSs can only be utilized with accurate anticipated traffic information.

For these reasons, proactive control is both theoretically attractive and practically advantageous than reactive control. Nevertheless, the benefits of proactive control may not be fully realized due to potential inaccuracies of traffic prediction. Therefore, the two control strategies should be complementarily implemented. Reactive strategy can be used for setting initial control parameters as soon as the incident is verified, or resetting of control parameters to adapt to changes in events. In particular, reactive control is applicable if critical situations such as heavy congestion, excessive queues, and over-saturation, prevail. Proactive control, on the other hand, governs route and access controls with anticipated information such as travel demand, travel time, and travel delay. It should be the primary mode of operation during the whole control process, especially when traffic conditions are not yet critical.

7.2 DECISION-MAKING LOGIC

7.2.1 Overview

Figure 7.1 illustrates the decision-making process in the last stage of the MS-FLC. The stage consists of three blocks, including intervention level, control strategy, and control action.

The *intervention level* indicates how strong the control needs to be, classified into 5 fuzzy sets: *No*, *Slight*, *Moderate*, *Strong*, and *Very_strong*. *No intervention* means no control action needs to be carried out since the congestion effect can be negligible. More precisely, the traffic control is implemented as usual since the incident creates no significant congestion effect. This associates with favourable

conditions (light traffic demand, no capacity reduction at the incident location, etc.) At the other extreme, *Very_strong* intervention level requires the maximum utilization of available resources to mitigate congestion. This intervention level is associated with critical congestion (heavy congestion, severe capacity reduction, high traffic demand).

Figure 7.1: Schematic flow of the 3rd stage

The employment of the intervention level attempts to "quantify" the level of necessary countermeasures in response to the current conditions. Since the fuzzy KBS is designed to assist the operator, the transformation from numerical inputs (input data, congestion level, congestion status, etc.) into these fuzzy terms facilitate the operator's understanding of the situation to formulate control decisions. The strength of intervention is represented by a nominal numerical domain to facilitate intuitive quantification. The inputs for the block include the traffic mobility and status of traffic congestion of heavy category evaluated from stage 1, and the predicted congestion level from stage 2 in case the current congestion is not heavy.

The *control strategy* represents the necessary countermeasures to deal with the incident problems. The strategy stands for the *supply side* that utilizes available resources in response to the control requirements having known the intervention level. It provides a broad methodological outlook in facing the control problems before concrete control actions are implemented. Strategies to be considered include several options, namely the isolated, coordinated, or integrated control, as presented in Section 2.2.1. The selection of control strategy should be made so that available resources are allocated in a cost-effective manner. Apart from the intervention level, the decision of which control strategy is suitable depends on the layout of the

expressway (Section 7.2.2), the locations of control devices, and the traffic conditions on street network.

Finally, the *control action* is a collection of specific control settings, given the selected control strategy. Examples of control actions include the setting of ramp metering, the management of queues, the determination of diversion split, and the dissemination of VMS messages.

7.2.2 Intervention level

The intervention-level block uses three input variables, including congestion level, congestion mobility, and congestion status. Since the input variables encompasses both prevailing and forecast information, the block is a blend of two control modes described in Section 7.1, *i.e.* reactive control and proactive control. The former is associated with critical problems such as heavy congestion, and the latter is applicable to non-heavy congestion.

Input variables

As in the case of the current congestion level, the predicted congestion level Pre_CL is characterised by 4 fuzzy sets:

$$Pre_CL = \{Free_flow, Light, Moderate, Heavy\} \quad (7.1)$$

while the congestion mobility C_Mob and congestion status C_Stat are described by two and three fuzzy sets, respectively (Sections 5.3 and 5.4):

$$C_Mob = \{SL-HC, MV-HC\} \quad (7.2)$$

$$C_Stat = \{SQ-HC, MQ-HC, LQ-HC\} \quad (7.3)$$

Output variable

The intervention level is *imprecisely* evaluated based on the input variables, and is scaled into a nominal numerical domain [1, 10] with 5 fuzzy sets (Figure 7.2).

$$Int_Lev = \{No, Slight, Moderate, Strong, Very_strong\} \quad (7.4)$$

Figure 7.2: Fuzzy sets of intervention level

Since the input variables independently coexist, the formation of rules in this block is straightforward: each condition of the rules is the union of several fuzzy sets, connected with an *OR* operator. The following describes 5 tentative rules for the intervention level:

R₁: If Pre_CL is Free_flow then Int_Lev is No

R₂: If Pre_CL is Light then Int_Lev is Slight

R₃: If Pre_CL is Moderate OR C_Stat is SQ-HC then Int_Lev is Moderate

R₄: If Pre_CL is Heavy OR C_Mob is MM-HC OR C_Stat is MQ-HC then Int_Lev is Strong

R₅: If C_Mob is SM-HC OR C_Stat is LQ-HC then Int_Lev is Very_strong

Rules *R₁* and *R₂* define the first sub-domains of the intervention level, where predicted congestion level is free flow or slightly congested. The canonical format of the rules implies that given such forecasted states of traffic there would be no prevailing moderate, heavy congestion, or queue. Rules *R₃*, *R₄*, *R₅*, on the other hand, implicitly indicate that drastic control measures need to be implemented given the severe current or predicted congestion level, or the existence of queues on the expressway. The *OR* operator in the conditions of the rules indicates that the predicates are equivalently considered.

7.2.3 Control strategy

Intuitively, if a control uses the local measurements and has a local impact on the traffic, it is known as the local control. The other control strategies are associated with area-wide control, or more precisely in this study - the corridor control. The determination of whether a corridor control should be coordinated or integrated is not straightforward. More specifically, the determination of the strategy for corridor control depends not only on the traffic and incident conditions, but also on the network topology, on the queue management policy, and on the background congestion level on the street network. The following section presents the methodology to determine control strategy following corridor control with the aid of Figure 7.3.

Figure 7.3: Layout of an expressway section

Figure 7.3 illustrates a basic hypothetical expressway segment (layout of expressways in Singapore) with an incident blocking one lane downstream. The segment is divided into sub-segments with traffic sensors. There are two on-ramps (ramp 1 and ramp 2) and an alternative route for diversion. To alleviate incident congestion, the control scheme considers three possible strategies. Let strategy 1 (ST1) denote the local control implemented by ramp 1, strategy 2 (ST2) denote the coordinated control implemented by the *coordination* of ramp 1 and ramp 2, and strategy 3 (ST3) denote the *integration* of ramp 1 and the route diversion. In addition, the integrated-control terminology can be extended to the combined effect of ramp 1, ramp 2, and the route diversion, etc., denoted as ST3-ext. Assume that traffic and incident problems require corridor control. The question is to determine which strategy (ST2 or ST3) is more suitable for control?

In order to determine a viable control option, the knowledge on time-varying OD demands is required. In Figure 7.3, O_1 , O_2 , and O_3 denote the origins where traffic is generated from ramp 1, ramp 2, and the mainline, respectively; D_1 , D_2 denote destinations. In Figure 7.4a, $\beta_{ij}(k)$ (%) indicates the proportion of traffic generated from zone i and destined to zone j during interval k . In principle, the decision of control strategy (ST2 or ST3) should be made by considering the $\beta_{ij}(k)$ values. Assume that $\beta_{ij}(k)$ values are represented by two fuzzy sets *Low* and *High* (Figure 7.4b).

Figure 7.4: OD assignment and membership functions

There are two possible situations that are associated with traffic demand upstream of ramp 1. If traffic demand upstream of ramp 1 is low, the implementation of corridor control is not justified. Hence, the assumption that the corridor approach is required associates with high upstream demand. The high demand upstream is the result of either high demand from ramp 2 (O_2) or/and from the mainline (O_3), denoted as $\beta_{21}(k)$ and $\beta_{31}(k)$ respectively. To alleviate the heavy congestion at the incident place, the arrivals upstream of ramp 1 should be managed by either regulating the entering flow at ramp 2 or/and diverging of traffic, which is willing to take risk by proceeding to the incident place on the mainline. These concepts can be restated by three simple rules:

If $\beta_{21}(k)$ is high then ST2

If $\beta_{31}(k)$ is high then ST3

If $\beta_{21}(k)$ is high AND $\beta_{31}(k)$ is high then ST3-ext

Alternatively, the decision can be made by considering the ratio of traffic demands from O_2 and O_3 that goes to D_1 . Let $\xi_1(k) = \frac{V_{31}(k)}{V_{21}(k)}$ denote the ratio of flow rates from zone 3 and zone 2 to zone 1, respectively, during interval k . If the ratio is low the demand from ramp 2 is relatively high, the coordinated control will be more effective. By contrast, if the ratio is high the demand on the mainline upstream of ramp 2 is relatively high, the integrated control should be considered. Please note that the terms *low* and *high* in this context are imprecisely defined (refer to Figure 7.4b). The above concept can be extended to an expressway system with n ramps and m diversion routes. However, it has been reviewed in literature that the control facilities far upstream have only little impact on the control scheme being considered.

Figure 7.5 illustrates a more detailed version of the decision-making sequence in deriving the control strategy. The process involves several steps, starting from traffic condition to intervention level, control scale, control strategy and control action. From the systematic points of view the traffic condition is the control input, control action is the output, while the other blocks are intermediate components that subordinate decision making at the strategic level. With an exception of control intervention, there will be no membership functions defined for the intermediate components. Instead, the decisions made in selecting control scale and control strategy rely on knowledge of traffic operators accumulated from daily control experiences in consideration of practical network layout and prevailing traffic conditions.

The *traffic condition* includes the evaluated and predicted traffic conditions from stages 1 and 2. The traffic condition is translated into intervention level, classified from *no control* to *very strong* level. In cases of *slight* and *moderate* intervention level, the control magnitude may be of local scale, whereas the corridor control is considered for *strong* and *very strong* level. Note that the control scale may be shifted by the operators during the control process subjected to changes in traffic trends within a certain period. Likewise, the control strategy is selected on the basis

of the recommended control scale and a number of traffic parameters and network attributes, including the V/C^* ratio, the ramp queue, and the congestion level of diversion routes. Like the control scale, the strategy may change during the control process subjected to the development of queues on ramps and the evolution of traffic conditions. Finally, given the selected control strategy, the system decides which control devices will likely be utilised with their locations known, and then relevant traffic conditions in the first block are used to calculate specific control settings, as presented in Section 7.3.

Figure 7.5: Decision tree for control strategy

Note that the labels associating with the intermediate components are not standard specifications but only suggestions. In the KB-DSS, the suggestions are presented to the traffic operators for their selection through interactive menus, discussed in Chapter 8.

7.3 FORMATION OF RULES

The above section describes the decision-making logic for control intervention and control strategy. This reasoning sequence is necessary to support decisions made by the operators at strategic levels. Given the selected control strategy, the system starts the operation at the implementation level. To allow better observation and tuning of the parameters and to facilitate the evaluation of the MS-FLC, in the following section the process of rule formation is presented with focus on the local ramp control. It should also be mentioned that the rule design in this section attempts to consider a complete control rule set, therefore the linguistic predicates of input variables are extended and slightly different from what have been suggested in Figure 7.5 for local ramp control.

7.3.1 Input variables

The input for control action includes the traffic conditions in Figure 7.5. The traffic conditions is a generic term characterised by a number of traffic and incident parameters, including the congestion level upstream of the incident, the traffic demand upstream of the ramp, the ramp queue, and the other incident attributes. Particularly, the change in traffic state also needs to be considered for proper switching in control settings.

(i) Traffic conditions upstream of the incident (downstream of the ramp)

Figure 7.6 illustrates the layout of a local ramp control. The traffic conditions associating with the local ramp control include the predicted congestion level, and short queue-heavy congestion SQ-HC.

$$Upstr_Traffic = \{Predicted\ CL, SQ-HC\} \quad (7.5)$$

where $Predicted\ CL = \{Free_flow, Light, Moderate\}$ (7.6)

Figure 7.6: Layout of local ramp control

(ii) The volume/capacity ratio

Traffic demand rate is defined as the number of vehicles wishing to pass a section in a given period of time. Traffic demand considered in this part does not include latent demand, but only the demand that currently uses the facilities. If the demand rate does not exceed the corresponding road capacity, all vehicles that wish to pass can be accommodated, and the measured flow rate in the field reflects the demand rate. Higher capacity induces higher demand, but demand is always equal or greater than observed throughput. If the demand rate exceeds the road capacity, the measured flow rate represents the number of vehicles that can be handled, not the number of vehicles that wish to be handled. In this case traffic demand rate is an unknown quantity. An indication of this imbalance is a queue upstream of the incident location due to traffic breakdown.

In addition to the traffic condition downstream of the ramp (upstream of the incident location), the volume/capacity ratio is another important factor in determining the ramp flow. The ratio indicates the relation between the demand of traffic that wishes to transverse the mainline and the reduced capacity due to the incident. This volume (V) does not include the ramp flow and can be determined from the measurement immediate *upstream of the ramp*. Prior to the beginning of the control scheme, such a measurement provides an approximation of actual *informed demand* in absence of the control intervention where the incident is partly known to a fraction of motorists. During the control, however, the demand is

affected since part of the traffic has been diverted. The notation (V) then reflects the actual traffic arrivals upstream of the ramp, but not a reasonable proxy of demand. However, these actual arrivals directly affect the amount of ramp traffic to be metered.

Figure 6.29 represents the volume/capacity relationship. From the V/C^* ratio, the level of the ramp traffic to be added to the expressway can be inferred. If the ratio is low, the level of ramp flow will be high. By reverse, if the ratio is high, the ramp rate should be restricted to avoid mainline congestion.

(iii) Ramp queue

The queues on ramps can be observed by means of traffic sensors or by queue detectors installed at the check-in point and at the ramp entrance. The magnitude of queues on the ramps exhibits high demand of street traffic that wishes to enter the expressway. Prevention of excessive ramp queues besides the alleviation of mainline congestion is one of the principal objectives of an incident management scheme. In terms of traffic management, maintaining reasonable ramp queues is important since it is difficult to dissipate the queues without causing mainline congestion if the queues are too long. With respect to concerns of social equity, although preventing excessive ramp queue does not always improve system-wide TTS, it helps avoid ramp traffic facing too long delays.

For evaluation of the concept *excessive ramp queue*, subjective judgments are required. Figure 7.7 illustrates a template of the membership functions for ramp queue, classified into three fuzzy sets *short*, *medium*, and *long* queue in association with the percentage of the maximum physical ramp storage $Q_{R-\max}$, proposed for ramps with medium lengths.

$$Q_R = \{Short, Medium, Long\} \quad (7.7)$$

It should be noted that the parameters of membership functions for ramp queue are site-specific since the queue management strategy aims not only to prevent excessive queue formation, but also to minimize the system-wide TTS. Therefore,

the parameters a , b , c should be calibrated through a sensitivity (parametric) analysis: Given a network (mainline, ramp) and a control system, the parameters (a , b , c) are altered while the network geometries are kept unchanged. The system is run in different iterations to observe how the objective function (e.g: TTT, or TTS) change with the parameters. The parameters should be a combination that optimises (sub-optimises) the objective function.

Figure 7.7: Membership functions for ramp queue

(iv) Other input variables

In addition to the three aforementioned input variables, the FLC needs to consider the other complementary factors such as incident attributes. These variables are needed to see whether the control strategy being implemented is adequate, or a corridor control should be adopted. The key factor of interest may be the *remaining time*, being the period from the current time to the expected end time of the incident. It is derived from the predicted incident duration minuses the elapse time since the incident starts. Although the predicted incident duration is a random variable and is hard to obtain precisely, it is reasonable to estimate it by traffic operator or by incident response team in consideration of a number of factors such as incident type, incident severity, the expected clearance time, the time of day, and the background traffic. It should be noted that there is an additional lag from when the incident actually occurs to the time it is detected. The issue of time lag has been discussed in Section 4.1.3.

The remaining time sets a threshold beyond which the current control strategy is considered to be inadequate and too costly, and should be changed to avoid severe congestion. The parameters of the membership functions for the remaining time are

subject to various control settings, depending on the time period in a day (e.g. peak/off-peak) and the congestion level of traffic environment. Figure 7.8 illustrates a possible form of membership functions for the time remaining for an incident occurring in the AM peak, with light-medium congestion level. The remaining time domain is labelled by three fuzzy sets:

$$\text{Remaining_Time} = \{\text{Short}, \text{Medium}, \text{Long}\} \quad (7.8)$$

Figure 7.8: Tentative membership functions for the remaining time

In Figure 7.8, the values a , b , c are bounded parameters, which could be established based on the operator's control experience.

7.3.2 Output variable

The output of the MS-FLC for the local ramp control is the flow rate of the ramp traffic that is allowed to enter the expressway in each time interval. To obtain a single numerical value for control, the recommended ramp flow is obtained by defuzzifying the control output that is represented by several fuzzy values (see Section 2.3.2 (iii)). The domain of ramp flow rate encompasses from *very low* to *very high* levels, or from zero to C_R where C_R is the metering capacity of the ramp.

Figure 7.9 plots the membership functions for ramp flow rate with respect to C_R .

The ramp flow rate domain $\left(\frac{V_R}{C_R}\right)$ is represented by 5 linguistic predicates:

$$\text{Ramp_Flow} = \{\text{Very_low}, \text{Low}, \text{Medium}, \text{High}, \text{Very_high}\} \quad (7.9)$$

Figure 7.9: Membership functions for ramp flow

Rule formation for control actions

With the above sets of state and control variables, the input-output mapping of the MS-FLC can be made using traffic engineering knowledge. The formation of rules is conducted such that the principal objectives of the control strategy - the amelioration of the mainline congestion and prevention of excessive ramp queues - are observed. Since these two objectives are often conflicting to each other, rules should be designed to compromise them at a balance point.

If the first three input variables are considered, the maximum number of rules is the combination of all scenarios, that is $4 \times 4 \times 3 = 48$ rules. However, some scenarios are not possible, meaningless and conflicting. Furthermore, several rules implicitly trigger others. Consequently, the number of meaningful rules is substantially lower.

Table 7.1 summarizes the decision rules for the local ramp control strategy. Each rule is a mapping between two (three) predicates in the rule conditions and one predicate in the rule conclusion. The rule conditions are joined with *AND* connectives. In correspondence to the control objectives, the conditions of the rules consider the traffic condition (congestion level, CL) upstream of the incident (*downstream of the ramp*), the traffic demand (indicated by the V/C^* ratio) *upstream of the ramp*, and the ramp queue (see Figure 7.6). For scenarios such that the traffic condition upstream of the incident and the V/C^* upstream of the ramp favour high ramp flows, the rules can be generated regardless of the ramp queue status. For example, a number of rules (rules 1, 2, 6, 7, 8, etc.) in Table 7.1 consider

Table 7.1: Decision table for rules with local ramp control

Rule	Rule condition		Rule conclusion		Key Control objective
	Congestion level (CL) upstr. of the incident	V/C^* upstr. of the ramp	Ramp Queue	Ramp Flow	
1	Free-flow	Low	-----	Very_high	Maximize mainline utilization
2	Free-flow	Medium	-----	High	Maximize mainline utilization
3	Free-flow	High	Short	Low	Prevent mainline congestion
4	Free-flow	High	Medium	Medium	Maintain acceptable ramp queue
5	Free-flow	High	Long	High	Prevent excessive ramp queue
6	Free-flow	Very_high	-----	Low	Prevent mainline congestion
7	Light	Low	-----	High	Maximize mainline utilization
8	Light	Medium	-----	Medium	Balance between objectives
9	Light	High	Short	Low	Prevent mainline congestion
10	Light	High	Medium	Medium	Prevent mainline congestion
11	Light	High	Long	Medium	Prevent excessive ramp queue
12	Light	Very_high	-----	Very_low	Prevent secondary queue
13	Moderate	Low	-----	Medium	Balance between objectives
14	Moderate	Medium	-----	Medium	Balance between objectives
15	Moderate	High	Short	Low	Prevent secondary queue
16	Moderate	High	Medium	Low	Prevent secondary queue
17	Moderate	High	Long	Medium	Prevent secondary queue
18	Moderate	Very_high	-----	Very_low	Prevent mainline congestion
19	SQ-HC	Low	-----	Medium	Balance between objectives
20	SQ-HC	Medium	Short	Low	Prevent mainline congestion
21	SQ-HC	Medium	Medium	Medium	Prevent excessive ramp queue
22	SQ-HC	Medium	Long	Medium	Prevent excessive ramp queue
23	SQ-HC	High	-----	Low	Prevent mainline congestion
24	SQ-HC	Very_high	-----	Very_low	Prevent mainline congestion

only two inputs: if the predicted CL *downstream* of the ramp is *Free-flow* (or *Light*), and the demand (V/C^*) *upstream* of the ramp is *Low*, then most of the ramp demand should be released.

R1: If CL is Free_flow and V/C^ is Low then Ramp_Flow is Very_high*

R2: If CL is Free_flow and V/C^ is Medium then Ramp_Flow is High*

The inputs are combined in such a way that predicates are scaled gradually over the input domains, and the outputs are translated elegantly from one fuzzy value to another. For example, in the following rules the V/C^* changes gradually from *Low* to *High*, whereas the *Ramp_Flow* is set from *High* to *Low*.

R7: If CL is Light and V/C^ is Low then Ramp_Flow is High*

R8: If CL is Light and V/C^ is Medium then Ramp_Flow is Medium*

R9: If CL is Light and V/C^ is High and Ramp Queue is short then Ramp_Flow is Low.*

Nevertheless, as in the case of rule formulation in Section 6.4, there are cases where different inputs produce the same output. For example, rules 13 and 14 propose a *Medium* level of ramp flow associated with different levels of traffic demand V/C^* :

R13: If CL is Moderate and V/C^ is Low then Ramp_Flow is Medium*

R14: If CL is Moderate and V/C^ is Medium then Ramp_Flow is Medium*

The rules regulate the ramp rate subjected to the demand upstream of the ramp (V/C^*) so as to prevent the mainline congestion:

R6: If CL is Free-Flow and V/C^ is Very_high then Ramp_Flow is Low*

or to prevent a secondary congestion at the ramp metering point:

R12: If CL is Light and V/C^ is Very_high then Ramp_Flow is Very_low*

R15: If CL is Moderate and V/C^ is High and Ramp_queue is Short then Ramp_Flow is Low.*

The reason for this restriction is that when the mainline is congested, the ramp traffic will hardly find an acceptable gap to join the mainline, so a secondary queue of the metered vehicles may form spontaneously. If a secondary queue persists, ramp metering is not beneficial. At the extreme, vehicles in the secondary queue may try to encroach the mainline, breaking down traffic upstream of the ramp and creating safety risk. Therefore, in the presence of a secondary queue, it is imperative that the vehicles be stored on the ramp to wait for an opportunity in the next period rather than being metered.

In addition to traffic upstream of the incident and V/C^* ratio, the ramp-queuing status needs to be considered. The rules are designed to achieve a proper balance between alleviating mainline congestion and adjusting the queue length. For example, with the same level of traffic demand upstream, the ramp flow is set to less restrictive rates if the ramp queue develops:

R3: If CL is Free-Flow and V/C^ is High and Ramp_queue is Short then Ramp_Flow is Low*

R4: If CL is Free-Flow and V/C^ is High and Ramp_queue is Medium then Ramp_Flow is Medium*

R5: If CL is Free-Flow and V/C^ is High and Ramp_queue is Long then Ramp_Flow is High.*

Most of the rules aim at a certain principal objective that could be set by the control operator or traffic manager. The objectives are the elaboration of overall goals of ramp control described in Chapter 2. If the traffic condition upstream of the incident is *Free-flow* or *Light* and traffic demand is *Low/Medium*, the ramp flow is set to *High/Very_high* level so as to *maximize mainline utilization* (rules 1, 2, 7). In

contrast, if the traffic demand (V/C^* ratio) upstream is *High/Very_high* the ramp flow is set to *Low/Very_low* levels to *prevent mainline congestion* (rules 3, 6, 9, 10, 18, 20, 23, 24). In addition, the ramp flow is adjusted according to the ramp queue status so as to *maintain acceptable ramp queue* (rule 4), to *prevent excessive ramp queue* (rules 5, 11, 21, 22), or to *maintain a balance between objectives* (rules 8, 13, 14, 19). Finally, if the traffic on the mainline is congested, the restriction of the ramp flow is to target *preventing a secondary ramp queue* at the ramp merge (rules 12, 15, 16, 17) for the reasons explained earlier.

Figure 7.10 describes the FIS interface of the rule base for ramp control with the set of 24 rules presented in Table 7.1. For the given data inputs, most of the rules are active, while rules 15, 17, and 18 are inactive since none of the predicates in the rule conditions satisfies. The FIS uses the *MIN* implication method (Section 2.3.2 (ii)) to obtain the output in each rule.

For example, rule 8 defines:

If Predicted CL is Light and V/C^ is Medium then Ramp_Flow is Medium*

For the given inputs, the output of rule 8 is inferred as:

$$\begin{aligned}\mu_{Medium}(Ramp_flow) &= \min\left[\mu_{Light}(predictedCL = 2.97), \mu_{Medium}\left(\frac{V_d}{C^*}\right) = 0.575\right] \\ &= \min[0.83, 0.33] = 0.33\end{aligned}$$

The FIS uses the MAX aggregation method and the centroid defuzzification to obtain the final output of ramp flow of 61.8%, namely $0.618 \times C_R$. Given the recommended ramp flow, the green time in each signal cycle can be inferred.

The readers may refer to Section 2.3.2 (ii) for theoretical explanation and Figure 2.13 for illustration of the FIS system's operation.

Figure 7.10: FIS for ramp control using MIN implication method

7.4 SUMMARY

This chapter presents the decision-making process of the last stage of the MS-FLC for control implementation. The process is decomposed into three blocks, including intervention level, control strategy, and control action. The intervention level quantifies the extent to which the control scheme should impose, while the control strategy reflects appropriate countermeasures to confronting with incident problems

given the prevailing traffic conditions, network topology as well as available control facilities, and the control action translates the selected strategy into concrete control settings.

The rules for the intervention level, control scale, and control strategy are formulated at strategic level. In the KB-DSS, these rules are presented to the traffic operators for their selection through a GUI interface. A case study in approaching the control strategy following corridor control has been presented in Section 7.2.3.

While the rules at strategic level are recommended for all control scales, rules at operational level focus on local ramp control to facilitate tuning parameters of membership functions and evaluating the MS-FLC. In formulating the control rules, all input variables that have significant effects on the control objectives have been employed to calculate ramp flows. The rules are combined so that they allow smooth transitions of settings from one state to another. Some of the control rules target a single objective, while the others attempt to keep balance between multiple objectives.

CHAPTER 8 MODEL VALIDATION AND EVALUATION

8.1 OVERVIEW

This chapter presents the development and validation of a Traffic Simulator and Control (TSC) model and the evaluation of the MS-FLC framework presented in previous chapters. The TSC model is developed in SIMULINK, (MATLAB 2002).

The TSC consists of two main components: the car-following model (CFM), and the traffic controller (TC). Although the simulation keeps track of individual vehicles, the TSC behaves more like a macroscopic traffic simulation model since only aggregated traffic variables, which are parameters of interest, are produced. The dynamic longitudinal interactions between vehicles, namely the car-following behaviours, are simulated by embedding the CFM component in the model. The aggregated outputs are sent to the TC for traffic control purposes. For simplicity in the model calibration and validation, the traffic in the multi-lane expressway where the data was obtained is represented as an equivalent single-lane system.

A standard microscopic traffic simulation package considers both car-following and lane-changing behaviour in a multi-lane expressway. SIMULINK, unfortunately, has no capability to capture lane-changing behaviours. Although in the congested condition there are little lane-changing opportunities, in free-flow conditions, lane-changing manoeuvres may have a significant effect on speeds and travel times of vehicles on the traffic stream. However, the parameters of interests are the overall macroscopic traffic variables that are averaged across lanes and hence may not be very sensitive to changing of vehicles from one lane to another, and a traffic control for non-recurring congestion normally focuses on congested conditions. For these reasons, the CFM developed in this thesis implicitly incorporates lane-changing impacts through the adjustment of its parameters. An iterative process of calibration simultaneously refines the model's parameters so that the model represents sufficiently closely to the real system behaviour. This was followed by a uncertainty analysis to identify the influential factors. The validation of the TSC was performed

by comparing the model's output with actual data collected from a segment on PIE in various traffic conditions. For the evaluation of the MS-FLC (in this Chapter this is called FLC for short), the developed FLC was embedded in the TC component. In the evaluation, the FLC performance was compared with the *No-control* scenario and *ALINEA* ramp controller over different traffic conditions and incident scenarios.

8.2 DEVELOPMENT OF THE TSC

8.2.1 Conceptual model

Figure 8.1 presents the conceptual model of the traffic simulator and control, the TSC. The CFM receives input data and simulates the traffic system based on car-following behaviour, and provides aggregated traffic data for the TC, which monitors the simulated traffic using either one of the control algorithms to be presented in Section 8.4.2. It requires external input data from files as well as internal input data from the CFM. Since the control decisions implemented by the TC alter the traffic system, the output from the TC is fed-back to the CFM that adjusts the gaps and acceleration/deceleration of vehicles. For this working mechanism, the TSC behaves as a close-loop control system.

Figure 8.1: Conceptual model of the TSC

8.2.2 Implementation model

(i) Introduction

The conceptual model of the TSC in Figure 8.1 is elaborated in the implementation model in SIMULINK. Like the conceptual model, the implementation model has two main components: the CFM and the TC, Figure 8.2.

Figure 8.2: Architecture of the TSC implementation model

Since the CFM is essential in the TSC model, the following section describes the key derivational backgrounds for the CFM development.

(ii) Car-following model

The car-following theory is applied to simulate how vehicles follow each other. Figure 8.3 illustrates a system of two vehicles running from left to right.

Figure 8.3: Car-following diagram

Notations and definitions in the Car-Following Theory

$n - 1$	: Lead vehicle	
n	: Following vehicle	
L_{n-1}	: Physical length of the lead vehicle	
L_n	: Physical length of the following vehicle	
L_{n-1}^{eff}	: Effective length of the lead vehicle, being the sum of the physical length and the minimum distance between stationary vehicles	
L_n^{eff}	: Effective length of the following vehicle	
$x_{n-1}(t)$	: Position of the lead vehicle	
$x_n(t)$	: Position of the following vehicle	
$\dot{x}_{n-1}(t)$	: Speed of the lead vehicle	
$\dot{x}_n(t)$	: Speed of the following vehicle	
$\ddot{x}_n(t + T)$	: Acceleration rate (deceleration rate) of the following vehicle	
T	: Reaction time	
Spacing between vehicles:	$S_n(t) = x_{n-1}(t) - x_n(t)$	(8.1)
Vehicle following distance:	$d_n(t) = (x_{n-1}(t) - L_{n-1}^{eff}) - x_n(t)$	(8.2)
Relative speed:	$\Delta \dot{x}_n(t) = \dot{x}_{n-1}(t) - \dot{x}_n(t)$	(8.3)
Time gap:	$g_n(t) = \frac{x_{n-1}(t) - x_n(t)}{\dot{x}_n(t)}$	(8.4)

The acceleration (or deceleration) of the following vehicle ($\ddot{x}_n$) is considered to occur at time $t + T$. If the relative speed is positive the lead vehicle has a higher speed and the spacing is increasing. By reverse, if the relative speed is negative the following vehicle has a higher speed and the spacing is decreasing. Analogously, if the $\ddot{x}_n(t + T)$ value is positive the following vehicle is accelerating, and if the $\ddot{x}_n(t + T)$ value is negative the following vehicle is decelerating. Equation (8.5) indicates that the acceleration of the following vehicle is a function of the relative movements of the two vehicles, including positions and speeds:

$$\ddot{x}_n(t+T) = f(x_{n-1}(t), x_n(t), \dot{x}_{n-1}(t), \dot{x}_n(t)) \quad (8.5)$$

Figure 8.4 illustrates the key elements of the acceleration (or deceleration) rate. The first element, e_{-1} , is the response of the following vehicle to the speed and spacing relative to its front vehicle. If the relative speed is small and the spacing is large the value of e_{-1} may be negligible, and the following vehicle adjusts its acceleration to attain *its desired speed*. This explains the reason why under *free-flow regime* the vehicle's speeds are relatively insensitive to the speeds and positions of the vehicles in front. In macroscopic quantities, speed is also relatively insensitive to flow rate in free-flow conditions.

Figure 8.4: The block diagram for acceleration/deceleration

In the *car-following regime*, the acceleration (or deceleration) rate constitutes two components:

$$\ddot{x}_n(t+T) = K_1(e_1) \times e_1 + K_g(e_g) \times e_g \quad (8.6)$$

where $K_1(e_1)$ and $K_g(e_g)$ are the gain parameters reflecting driver behaviour and aggressiveness. Ranges of calibrated values of $K_1(e_1)$ and $K_g(e_g)$ are shown in Table 8.1.

The CFM is developed in SIMULINK and is presented in Figure 8.5. The speed profile of the first vehicle is provided, while the time gaps (g) between vehicles at the origin are implicitly provided through flow rates: SIMULINK reads HDB data from volume and speed reference matrices (*volume1.mat* and *speed1.mat* in Sub-block 1.1.1, Appendix C), provided in 5-minute intervals. Since 5-minute interval is too large for the generation of vehicles, it is divided into shorter intervals (1 second, 5 seconds, or 10 seconds, etc., specified by the user). The generation of the number of vehicles in each short interval considers two cases:

- In *free-flow regime* the number of the vehicles generated in each short interval (counting distribution) is approximated by Poisson distribution:

$$P_n(t) = \frac{(\chi t)^n}{n!} e^{-\chi t} \quad (8.7)$$

where χ denotes the mean rate of the arrivals calculated from the field data in the HDB (in 5-minute interval); t is the time of short interval; n is the number of arrivals in each short interval. In SIMULINK, in the *free-flow regime*, vehicles were generated using the Poisson Integer Generator.

- In *congested conditions*, vehicles are more regularly distributed along the highway, the mean headways are calculated directly using the flow rates given by the reference volume matrix from the HDB.

Figure 8.5: Car-following model

The time gaps are continually adjusted with changes in the ramp flows: the traffic controller recommends ramp flows using control algorithms (*FLC*, *ALINEA* and *ALINEA* $\backslash Q$, see Section 8.4.2 and Appendix D). In the network, vehicles' speeds are adjusted according to the time gaps with the vehicles immediately in front. In the CFM the acceleration and speed integrators are used to construct vehicle trajectories, which is progressively updated every 0.1 second during the simulation. Appendix C provides more details and better visualization of the CFM.

(iii) Traffic controller

The second component of the TSC is the traffic controller (TC), described in Appendix D. The TC implements the control algorithms and calculates traffic parameters in different sections for the model evaluation. In the FLC control method the FLC controller is embedded in the TSC for the execution the fuzzy rules. The FLC consists of 7 rule blocks, illustrated in Figure 8.6.

Figure 8.6: The FLC in SIMULINK

The state and control variables of the *FLC* controller have been explained in the previous chapters. The *ALINEA* (*ALINEA\Q*) algorithms will be described in Section 8.4.2 and Appendix D.

8.3 MODEL VALIDATION

8.3.1 Calibration of the CFM

The calibration of the CFM targets identifying and tuning the parameters that affect the model accuracy by comparing the model with the real system. The acceleration/deceleration behaviour is the kernel of the CFM model. In the real world, this behaviour depends on many factors, including driver's aggressiveness and awareness, network geometry, obstructions on the road, vehicle type, etc. In short, the behaviour depends on *driver*, *vehicle*, and *driving environment* elements. Therefore, the calibration is a complex and iterative process of comparing the

model with the reality, locating model deficiencies and making adjustments to the parameters until the model is judged to be sufficiently accurate.

In this Chapter, the calibration is conducted by fitting the model with actual data on a number of segments of PIE in the case study presented in Chapter 4. The historical profiles of November 2003 on segments 80007758, 80007762, 80007766, and 80007770 were used for the model calibration, while the data on segment 80007774 were used for the model validation. The calibration process targets the minimization of the following objectives:

$$Z_v = \sqrt{\sum_i \sum_d (v_{actual} - v_{sim})^2} \quad (8.8)$$

$$Z_V = \sum_i \sum_d \left(\frac{V_{actual} - V_{sim}}{V_{actual}} \right) \quad (8.9)$$

where i indicates the segment in the case study; d indicates the day in the calibrated period; v_{actual} and V_{actual} denote the observed historical speed and volume; v_{sim} and V_{sim} denote the simulated speed and volume, respectively.

Constrained by the time and complexity, the calibration process has been conducted manually with trial-and-error iterations. Table 8.1 introduces the most important parameters identified through the calibration process.

Table 8.1: The most influencing parameters in the CFM

(Note: (*): variable. The lower limit corresponds to lower speed, and vice versa; (**): value depends on human factors and vehicle type)

Index	Parameter	Unit	Value
1	Desirable gap G^{**}	sec.	1.25 to 3.5
2	Gain factor for acceleration $K_{1,a}^*$	m/s	0.7 to 1.0
3	Gain factor for acceleration $K_{g,a}^*$	m/s ³	-1.5 to -1.0
4	Gain factor for deceleration $K_{1,d}^*$	m/s	0.7 to 1.0
5	Gain factor for deceleration $K_{g,d}^*$	m/s ³	-1.2 to -1.0
6	Maximum acceleration a_{max}	m/s ²	3.0
7	Maximum deceleration d_{max}	m/s ²	-3.0
8	Speed limit	km/h	(*)
9	Reaction time T^*	sec.	1.0 to 1.5

The following section presents car-following behaviours in an n-vehicle system over a simulated expressway segment with the length of 1.5 km, simulation time of 3,600 seconds, and the desired gap of 3.5 seconds. The other model parameters are taken in the range listed in Table 8.1. The simulated expressway segment is extended before its start point by a dummy link whose length is not limited. In the dummy link, the vehicles behave in the same manner as in the simulated segment. Therefore, stable conditions can be expected as the vehicles enter the simulated network. When joining the platoon in the dummy link, the last vehicle immediately attains its speed, which is equal to the speed of the vehicle in front, thus time and distance for physical acceleration from speed zero is not required.

Figure 8.7 makes a zoom for the first two hundred seconds of the vehicles' positions and speeds from the simulation profiles. In the figure (upper part) the trajectory of each vehicle is demonstrated by a curve (diagonal) that consists of a set of colours where each colour is associated with the order of the vehicle in the traffic stream: the first vehicle is represented by the yellow segment, the second vehicle is represented by the magenta segment, etc. At any time instance t , the platoon is lead

by one first vehicle. When a lead vehicle reaches the end of the simulated boundary, it is removed from the system and immediately joins the platoon in the dummy link (see Block 1.1: VEHICLES IN RANGE, Appendix C). At the same time, the order of the vehicles is changed: the second vehicle becomes the first; the third vehicle becomes the second, and so on. The vertical lines in the upper part of the Figure indicate the shift in the orders of vehicles in the platoon whenever a lead vehicle reaches the end point of the network.

Figure 8.7: A zoom in the simulation profiles of vehicles' positions and speeds

The speeds of vehicles are described in the lower part of Figure 8.7. The speeds change in the range 40-81 km/h. When a vehicle joins the platoon in the dummy link, its speed is started the same as the speed of the front vehicle. Subsequently, the

speeds are adjusted to the gaps available: the vehicles accelerate if the actual gap is larger than the desirable gap, and decelerate if the gap is smaller.

Figure 8.8 depicts the vehicles spacing, time gap, and acceleration/deceleration from the simulation. The figure shows that the spacing between vehicles typically change in the range 40-80 m, where the larger range is associated with vehicles in front of the platoon, and smaller values are associated with vehicles in the end of

Figure 8.8: Vehicle's spacing, time gap, and acceleration (n-vehicle system)

the platoon. The figure shows that, excepting for approximately the first 300 seconds as the "warm-up" period, traffic up to the 2,500th second was in stable condition, which is represented by fairly constant spacing, time gaps and low values of acceleration/deceleration. The time gaps fluctuated around the desired gap of 3.5

seconds, and the acceleration/deceleration moved around -0.12 m/s^2 to $+0.3 \text{ m/s}^2$. From the 2,500th second there was a disturbance of traffic condition, and the spacing, time gap and acceleration fluctuate more widely: the spacing between vehicles change in the range 30 - 88 m, the time gaps deviated far away from the desired gap, from 2.4 to 4.6 seconds, leading to the change in the acceleration/deceleration, from -2 to $+2 \text{ m/s}^2$.

8.3.2 Validation of the TSC model

I. The site

The site selected for the model validation is the expressway section 80007774 along the PIE, from Mount Pleasant road to Jalan Toa Payoh (Figures 4.4 and 8.9). The section has three lanes in each direction, consisting of the upstream link 103066282 with the length of 292 m, and the downstream link 102058526 with the length of 337 m. These segments are probably relatively short compared to the average link segment of 418 m of the expressway network, but may be prevailing in the built-up areas in the urban expressway system of Singapore. In addition, given the complexity in designing the TSC and the limited time for this study, the site represents a simplified network for the experimental purpose.

Figure 8.9: Actual site for model validation

There is an on-ramp from Mount Pleasant road. The data used for validation is the historical OD demands in December 2003 in the HDB, aggregated in 5-minute intervals. Since the network has only two OD pairs (upstream link - downstream link, and ramp - downstream link, in Figure 8.9), the link counts directly reflect the OD demands.

Three typical traffic conditions collected in December 2003 were selected for the validation: free flow, medium congestion, and heavy congestion during peak, off-peak and nighttime periods. There is no control at the ramp. Traffic demand from the ramp ranges between 200 and 800 veh/h, depending on the time of day.

Since the CFM converts multi-lane traffic into a single equivalent lane flow, and the traffic composition varies considerably between lanes and with time of day, the equivalent lane flow is computed as the weighted average flow, while speed is the harmonic mean speed of the three lanes, taking into account the lateral distribution among lanes in different periods.

$$q_{eq}(t) = \frac{1}{3} \sum_{i=1}^3 \xi_j p_{ij}(t) q_i(t) \quad (8.10)$$

where $q_{eq}(t)$ denotes the equivalent lane flow rate in Passenger Car Unit (PCU); ξ_j denotes the Passenger Car Equivalent factor (PCE) of vehicle type j converted into PCU; $p_{ij}(t)$ denotes the proportion of vehicle type j in lane i during interval t ; $q_i(t)$ denotes the flow rate in lane i during interval t (veh/h).

Similarly, the equivalent lane speed $v_{eq}(t)$ is computed as:

$$v_{eq}(t) = \frac{\sum_i v_i(t) \times q_i(t)}{\sum_i q_i(t)} \quad (8.11)$$

where $v_i(t)$ and $q_i(t)$ are speed and flow rate of lane i during interval t , respectively. This assumes that platoons of vehicles in each lane travel with the same speeds in each time interval.

The values of ξ_j for the Pan Island Expressway (PIE) in Singapore was first studied by Fan (1990), as shown in Table 8.2. It should be noted that the PCE values by Fan (1990) were estimated for capacity flow conditions for level terrain that is prevailing in Singapore expressways. Under free-flow conditions, the PCE values for heavy vehicles may be lower: the effect of heavy vehicles on traffic stream is a function of the interaction between them and smaller vehicles in the traffic stream, and is mainly attributed to two important factors: physical dimensions (larger dimensions) and performance (inferior acceleration performance and lower maximum speeds on steep and long upgrades). Under congested conditions, the effect increases due to the greater interaction between heavy vehicles and smaller vehicles in the traffic mix, namely greater interaction between various types of vehicles that have different performances. Under free-flow conditions, it can be expected that larger vehicles would have a smaller effect than under congested conditions (Al-Kaisy et al., 2005) since the interaction between heavy vehicles and smaller vehicles is less intense, which leads to higher passing opportunities for the smaller vehicles to keep their desired speeds.

Table 8.2: PCE equivalent factors for Singapore expressways

(Source: Fan, 1990)

Vehicle type (j)	ξ_j
Cars and taxis	1.00
Light goods vehicles	1.30
Heavy trucks	2.60
Buses	2.70
Motorcycles	0.40

Given the limited source of reference and the fact that terrain is among the most influencing factors of PCE and has been considered by Fan (1990), the values are proposed to use as a rough estimate for converting various types of vehicles into Passenger Car Unit.

The proportion of vehicles $p_{ij}(t)$ is not only lane-specific but also time-dependent. Spatially, the fastest lane is represented by higher proportions of cars and vans, while heavy goods vehicle presents mainly on the slow lane. Temporally, during peak periods, commuter vehicles may be the predominant component of traffic while truck accounts for a significant proportion during off-peak and nighttime.

Since the collection of data for every segment and time interval is very time consuming and cost expensive, it would be reasonable to conduct sampling surveys for typical time periods, being peak, off-peak and nighttime, at representative locations. Table 8.3 shows the result of video-based surveys on segment 80007770 of PIE that has similar geometrical characteristics (lane width, number of lanes) with the simulated site (segment 80007774, see Figure 4.4), conducted on Thursday 5th and Tuesday 10th, December 2002. Since they are two subsequent segments, the traffic composition and lane distributions may be similar to each other. The survey periods ranged from 45 minutes (AM peak) to 75 minutes (off peak) that experienced all free-flow, medium, and congested conditions.

Table 8.3: Lateral distributions of traffic on PIE

Traffic lane	Lower value (%)	Upper value (%)
Lane 1	39.47	42.1
Lane 2	31.60	35.94
Lane 3	24.62	26.30

The traffic composition was also obtained from the same survey. Table 8.4 shows ranges of different vehicle types for AM peaks. The values have been used to calibrate parameters in the CFM and to convert the $q_{eq}(t)$ in PCU values in the simulation. It should be noted that under free-flow conditions there might be bias of flow rates towards the fastest lane (lane 1), while under congested condition traffic tends to distribute more equally between lanes. Therefore, in selection of traffic composition between lanes, the use of either extreme values or the average depends on traffic conditions.

Table 8.4: Traffic composition on PIE in AM peaks

Vehicle type	Lower value (%)	Upper value (%)
Car	62.60	66.10
Van	6.22	7.55
Light truck	13.70	17.8
Heavy truck	3.20	7.32
Bus	0.50	1.52
Motorcycle	5.23	5.89

The simulated traffic speed and flow rate were compared to the observed data for the upstream and downstream links separately:

- Comparison of simulated counts with actual counts
- Comparison of simulated speeds with actual speeds

The settings of simulation for validation include: simulation time of 90 minutes, including 30 minutes of the warm-up period. The parameters of interest are one minute-aggregated speeds and densities, collected from the 31st to 90th minute.

II. Validation

(i) Speed

Since the actual and simulated speeds are collected in pairs, where each pair represents one-to-one correspondence, the paired t-test is used for analysing the differences between readings of speeds on each interval.

Let $(X_{11}, X_{21}), (X_{12}, X_{22}), \dots, (X_{1n}, X_{2n})$ be a set of n paired observations where the mean and variance of the population represented by X_1 are μ_1 and σ_1^2 , and the mean and variance of the population represented by X_2 are μ_2 and σ_2^2 . Define the differences between each pair of observations as $D_j = X_{1j} - X_{2j}$, $j = 1, 2, \dots, n$. The differences are assumed normally distributed with mean:

$$\mu_D = E(X_1 - X_2) = E(X_1) - E(X_2) = \mu_1 - \mu_2 \quad (8.12)$$

and variance σ_D^2 . The test statistic is:

$$T_0 = \frac{\bar{D} - \Delta_0}{S_D / \sqrt{n}} \quad (8.13)$$

where $\bar{D}$ is the sample average of the n differences $D_1, D_2, \dots, D_n$, and S_D is the sample standard deviation of these differences.

Table 8.5 shows the actual and simulated speeds (km/h) for upstream segment of the simulation under free-flow condition in the evaluation period.

The paired t-test for this condition is conducted through the 8-step procedure as follow:

- i. The parameter of interest is the difference in the means of actual and simulated speeds, μ_D . Assume $\mu_D = 0$.
- ii. Null hypothesis $H_0 : \mu_D = 0$
- iii. Alternative hypothesis $H_1 : \mu_D \neq 0$
- iv. Significance level $\alpha = 0.05$
- v. The test statistic is:

$$t_0 = \frac{\bar{d}}{s_D / \sqrt{n}} \quad (8.14)$$

- vi. Reject H_0 if $t_0 > t_{\alpha/2, n-1} = t_{0.025, 59} \approx 2$ or if $t_0 < t_{0.025, 59} \approx -2$.
- vii. Calculation: the sample mean and standard deviation of the differences are $\bar{d} = -0.114$, and $s_D = 2.268$. The resulting test statistic $t_0 = -0.387$.
- viii. Conclusion: Since $t_0 = -0.387 > -2$, the null hypothesis $\mu_D = 0$ cannot be rejected at the significance level $\alpha = 0.05$.

Table 8.5: Actual and simulated speeds on upstream segment (free-flow)

Minute	Actual speed	Simulated speed	Difference
31	71.76	74.43	-2.66
32	72.13	75.92	-3.79
33	74.19	76.39	-2.21
34	76.38	78.37	-2.00
35	78.98	76.47	2.51
36	80.81	79.52	1.29
37	79.37	81.77	-2.41
38	81.73	83.86	-2.13
39	78.91	82.83	-3.91
40	82.29	84.42	-2.13
41	80.85	82.29	-1.44
42	79.26	80.48	-1.22
43	78.73	77.15	1.58
44	80.59	81.80	-1.20
45	82.95	81.73	1.22
46	85.21	84.29	0.92
47	85.69	81.88	3.80
48	84.43	81.21	3.22
49	85.79	79.12	6.67
50	83.15	80.29	2.85
51	83.44	81.34	2.09
52	82.97	83.19	-0.22
53	81.88	81.58	0.30
54	82.88	80.30	2.57
55	79.80	82.59	-2.79
56	76.73	77.83	-1.10
57	74.22	74.33	-0.11
58	74.15	76.09	-1.94
59	77.12	74.24	2.87
60	81.03	78.36	2.67
61	83.58	83.37	0.21
62	86.38	86.40	-0.02
63	85.30	88.32	-3.03
64	86.17	86.36	-0.19
65	83.05	82.50	0.55
66	81.44	83.80	-2.36
67	80.30	81.93	-1.63
68	80.29	79.45	0.85
69	79.20	81.70	-2.50
70	80.59	79.06	1.53
71	80.89	79.47	1.43
72	79.81	78.17	1.64
73	82.67	77.37	5.30
74	81.53	83.12	-1.59
75	81.41	80.78	0.63
76	82.17	79.35	2.82
77	81.63	80.24	1.39
78	83.42	83.82	-0.40
79	83.33	83.79	-0.46
80	81.89	82.03	-0.14
81	81.36	82.05	-0.70
82	81.44	81.61	-0.18
83	79.92	79.42	0.50
84	81.89	84.56	-2.67
85	79.44	79.47	-0.03
86	77.55	81.28	-3.72
87	76.35	80.41	-4.05
88	74.99	76.87	-1.87
89	73.85	74.81	-0.96
90	73.74	74.03	-0.30

Table 8.6 summarises the results from the paired t-test for the upstream and downstream segments under free-low, medium congestion, and heavy congestion. The table shows that the simulated speed is not significantly different from the actual speed at the significance level $\alpha = 0.05$ for both segments under all three prevailing traffic conditions.

Table 8.6: Summary of paired t-test for speed analysis

Simulated regime	Segment	$\bar{d}$	S_D	n	t_0	Conclusion
Free flow	Upstream	-0.114	2.268	60	-0.387	Not significantly different
	Downstream	-0.122	2.389	60	-0.396	Not significantly different
Medium congestion	Upstream	0.160	2.599	60	0.470	Not significantly different
	Downstream	-0.114	1.717	60	-0.517	Not significantly different
Heavy congestion	Upstream	-0.242	1.120	60	-1.674	Not significantly different
	Downstream	0.321	1.452	60	1.712	Not significantly different

(ii) Flow rate

The accumulated simulated counts are plotted against the actual counts for upstream and downstream links in Figures 8.10 to 8.12 under free-flow, medium congestion, and heavy congestion, respectively. The figures show that although the simulated data are slightly undercounted under the medium and heavy congestion, the data pairs stick relatively closely to the perfect lines, represented by diagonals.

a) Upstream link

b) Downstream link

Figure 8.10: Simulated versus observed count under free-flow condition

a) Upstream link

b) Downstream link

Figure 8.11: Simulated versus observed count under medium congestion

a) Upstream link

b) Downstream link

Figure 8.12: Simulated versus observed count under heavy congestion

Since the magnitude of absolute errors for traffic volume is large, it may be difficult to envisage using RMSE. Therefore, the Mean Absolute Percentage of Error (MAPE) is used instead:

$$MAPE(\%) = \frac{\sum_{i=1}^N \left(\frac{|V_i^a - V_i^s|}{V_i^a} \right) \times 100}{N} \quad (8.15)$$

where V_i^a and V_i^s denote the actual and simulated flow rates, respectively, in interval i ; N is the number of intervals ($N = 60$).

Table 8.7 lists the MAPEs for simulated traffic counts where the errors vary from 2.21% to 3.15%. It should be noted that the comparison of simulated flow is made on the basis of PCU values, which have been converted from various types of vehicles. Given that the PCE values in Fan (1990) was estimated for congested condition, the errors for free flow and medium congestion may be higher than those reported in Table 8.7.

Table 8.7: Errors of simulated flow rates (MAPE, %)

Simulated regime	Upstream link	Downstream link
Free flow	2.21	2.30
Medium congestion	2.54	2.85
Heavy congestion	2.85	3.15

An inspection of results from the simulation shows that there are higher local deviations of the simulated volume from the actual volume. Figure 8.13 gives an example of the absolute percentage of errors from all intervals for the upstream segment, under heavy congestion scenario. The Figure shows that although the MAPE for this scenario is 2.85%, errors of greater than 5% can be seen in 8 out of 60 intervals.

Figure 8.13: Absolute percentage of errors on upstream segment
(Heavy congestion)

8.4 MODEL EVALUATION

8.4.1 General issues

I. The network

For the model evaluation, it would ideally be to use observed data with a real network to investigate the model behaviour under different conditions. However, it is technically difficult, time consuming, and expensive to obtain data from actual sites. By contrast, in simulation traffic can be replicated from one run to another, therefore the outcomes among scenarios can be compared easily. Apart from that, the simulation-based evaluation using a generic network is a viable alternative that offers greater flexibility in considering various traffic conditions and incident scenarios. With simulation-based evaluation, the criteria used to evaluate the effectiveness of control algorithms can be easily and uniquely obtained.

In these regards, this section presents the evaluation of the FLC control algorithm using a simulated network illustrated in Figure 8.14. The network is modelled after

that described in Section 8.3.2. It has three links: the link upstream of the ramp, the link upstream of the incident (downstream of the ramp), and the link downstream of the incident. For local ramp control, most of the measurements are made in the vicinities of the incident, namely the upstream and downstream links. The lengths of the links used in this experiment are $L_{upstr} = 1000\text{ m}$, $L_{downstr} = 500\text{ m}$. The lane-blocking incident reduces the capacity of the expressway, and the local ramp control is implemented to regulate traffic demand from the ramp to prevent or ameliorate the mainline congestion.

Figure 8.14: Layout of the simulated network

(Not to scale)

The inputs of the TSC involve 2 pairs of time-dependent OD demands, speed profile of the first vehicles, and time-varying splits at the diversion route. The time varying splits are specifically considered in the rules in the FLC algorithm. As presented earlier, the multi-lane expressway is converted into a single-lane expressway that is calibrated so that lane-changing behaviours can be implicitly captured in the simulation model.

II. Simulation settings

The basic parameters of the simulation experiment are set as follows:

Simulation time: 90 minutes, including:

- From the 1st min. to 30th min.: normal traffic
- From the 31st min. to 60th min.: incident period
- From the 61st min. to 90th min.: normal traffic

- Evaluation interval: every 10 seconds
- Evaluation period: from the 16th min. to 90th min.

To achieve a high level of representation and accuracy, the vehicle's acceleration, speed and position are updated every 0.1 second.

III. Simulation scenarios

The experiment investigates a wide range of traffic conditions and incident situations. The traffic OD flows are loaded at Low, Medium, and High demand levels, the values of which are recommended in Section 8.4.3. It should be mentioned that the terms *Low*, *Medium*, *High*, *severe*, *very severe*, *etc.* for traffic demand and capacity reduction used in this experiment are *fuzzy-like* based on judgements in consideration of the reduced capacity (see Figure 6.29). They are not unique, but used to classify traffic status and incident severity.

In addition to traffic conditions on the expressway and on the ramp, the experiment investigates various incident scenarios, including capacity reduction and incident location. Although the capacity reduction is usually estimated on the basis of the number of lanes closed in association of the total number of available lanes, for instance 50% capacity reduction is estimated if one of the three lanes is closed, the experiment relaxed this custom for more flexible scenarios. The changes in capacity reduction (from medium to very severe levels) and incident location are considered in the *High demand* traffic scenario, described in Section 8.4.3, part (iii).

IV. Parameters of interest

The parameters of interest used for control and evaluation are aggregated variables including traffic flow rate $q_{(i)}(t)$, speed $v_{(i)}(t)$, and density $k_{(i)}(t)$ for every interval t , where the (i) denotes the location upstream and downstream of the incident. Apart from that, the queues on expressway and on the ramp are also collected.

Traffic demand

Traffic demand (q_d) immediate upstream of the ramp can be obtained from the time-dependent OD demand minus the diversion flow:

$$q_d(t + \tau_1) = q_{O_1}(t) - q_{div}(t) \quad (8.16)$$

where $q_{O_1}(t)$ denotes the flow rate generated from origin O_1 during time interval t , and τ_1 denotes the travel time from O_1 to the detector immediate upstream of the ramp. The diversion component is shown as in Appendix D, Block 2: TRAFFIC CONTROLLER - FLC control (Page D-2), and is elaborated in Sub-block 2.3: FLC Controller (Page D-8).

Speeds

The speeds at upstream link $v_{upstr}(t)$ and downstream link $v_{downstr}(t)$ are space mean speeds aggregated for each interval t . The method to obtain space mean speeds is illustrated in Appendix D, Block 2.1.

Density

Details on the calculation of aggregated flow rate, speed, density, and travel time are presented in Appendix D, Blocks 2.1 and 2.2, in which Block 2.1 presents an alternative method to calculate density during an interval as the ratio between the counted number of vehicles on the segment during that interval and the length of the segment.

Queues on the expressway and the ramp: $Q_{exp}(t), Q_r(t)$

Figure 8.15 describes the concept for deriving the queue from the arrival-departure curve. The queue at certain simulation time t equals the accumulated arrival minus the accumulated departure, assume the boundary condition $Q_{(t=t_0)} = 0$, where t_0 is the start time of the evaluation period (minute 16th onward).

Figure 8.15: Arrival-departure curve

$$Q_{\text{exp}}(t) = \sum_{t=t_0}^{t=T} (q_{\text{upstr}}(t) - q_{\text{downstr}}(t)) \times \Delta_t \quad (8.17)$$

where

$$q_{\text{downstr}}(t) = C^* \text{ if } Q_{\text{exp}} > 0 \quad (8.18)$$

$$q_{\text{downstr}}(t) = \min(q_{\text{upstr}}(t), C^*) \text{ if } Q_{\text{exp}} = 0 \quad (8.19)$$

where T denotes the simulation time; Δ_t denotes the time of simulation interval; C^* denotes the remaining capacity.

This logic is implemented with the aid of Switch 1 in Figure 8.16. Switch 1 has three inputs: inputs 1 and 3 are data inputs while input 2 is the control input. Data will be passed through input 1 when input 2 satisfies the selected criteria, otherwise data will be passed through input 3.

Figure 8.16: Calculation of queues on expressway

The criterion for passing through input 1 is $U_2 = Q_{\text{exp}} > 0$. Since Equations (8.17) to (8.19) are mutually dependent, the algorithm is implemented in SIMULINK using the Memory function that retrieves the data on the expressway queue from the previous simulation interval.

The same concept is used for calculating the queue on the ramp:

$$Q_r(t) = \sum_{t=t_0}^{t=T} (q_r^{\text{gen}}(t) - q_r(t)) \times \Delta_t \quad (8.20)$$

where $q_r^{\text{gen}}(t)$ denotes the flow rate of traffic entering the ramp; $q_r(t)$ denotes the actual ramp flow entering the expressway during time interval t .

$$q_r(t) = \min(C_r, q_r^{\text{model}}) \text{ if } Q_r(t) > 0 \quad (8.21)$$

$$q_r(t) = \min(C_r, q_r^{\text{model}}, q_r^{\text{gen}}(t)) \text{ if } Q_r(t) = 0. \quad (8.22)$$

where C_r stands for the ramp capacity, and q_r^{model} stands for the flow rate recommended by the control algorithms.

The control logic for the ramp queue is implemented in SIMULINK by the rule block described in Figure 8.17, with the aid of Switch 2. If the queue on the expressway propagates up to the ramp location, the ramp totally closes. This is implemented with control group in the dotted-boundary rectangle with relational operator ($<$): if the logical comparison holds true, the control group returns 1, otherwise it returns 0.

Figure 8.17: Calculation of ramp queue

V. Measures of effectiveness (MOEs)

The TSC uses the following measures of effectiveness as the evaluation criteria:

a) Total travel time on the expressway, TTT (veh.h)

The TTT is the sum of travel times of individual vehicles. In SIMULINK, the TTT is the sum of the number of vehicles in the expressway $N(t)$ over time in successive intervals:

$$TTT = \sum_{t=t_0}^{t=T} N(t) \times \Delta_t \quad (8.23)$$

The TTT is a principal evaluation criterion. The calculation of the TTT allows the comparison of the total time spent in the system (Part c). Alternatively, the average travel time can also be used, however, the average travel time can be inferred directly from the average speed (Part e) that is used as one of the major MOEs. The lower the TTT indicates the positive signal, providing that higher throughput and higher speed are also obtained. Nevertheless, if the lower TTT is the result of too restrictive a control method that produces a lower throughput, this “saving” is misinterpreted. Therefore, the TTT should be evaluated in accordance with the other MOEs.

b) Total waiting time on the ramp, TWT (veh.h)

The TWT is the accumulated waiting time of vehicles in the ramp queue due to the control regulation. Like TTT, TWT is the sum of the number of vehicles in ramp queue Q_r over time in successive intervals:

$$TWT = \sum_{t=t_0}^{t=T} Q_r(t) \times \Delta_t \quad (8.24)$$

Unlike TTT, TWT is a secondary criterion an incident management strategy normally sets a higher priority for the expressway than the ramp traffic.

c) Total time spent in the system, TTS (veh.h)

The TTS is the total time all vehicles spend in the system during the simulation period, being the sum of the TTT and TWT.

$$TTS = TTT + TWT \quad (8.25)$$

d) Total travel distance, TTD (veh.km)

The TTD is the sum of distances travelled by individual vehicles during the simulation. In SIMULINK, the TTD is calculated as the sum of the total of travel distances upstream and downstream sections in successive intervals.

$$TTD = \sum_{t=t_0}^{t=T} [N_{up}(t) \times \bar{V}_{up}(t) + N_{down}(t) \times \bar{V}_{down}(t)] \times \Delta_t \quad (8.26)$$

where $N_{up}(t)$ and $N_{down}(t)$ denote the number of vehicles in upstream and downstream links during interval t ; $\bar{V}_{up}(t)$ and $\bar{V}_{down}(t)$ are the space mean speeds during the same period.

Like TTT, TTD is a primary MOE since it indicates the level of “productivity” the expressway yields. It encompasses both the mainline throughput and average speed.

e) Average speed on expressway, MS (km/h)

The MS on the expressway is among the most important criteria since it represents the dynamics of a vehicle's motion. The average speed is calculated as the ratio of TTD and TTT.

$$MS = \frac{TTD}{TTT} \quad (8.27)$$

where TTD and TTT are associated with the same number of vehicles (see Block 2, Appendix D).

f) Mean density, MD (veh/km)

Like speed, MD is a primary indicator of congestion level. The mean density is the arithmetic mean of traffic densities $k(t)$ in the network in successive intervals.

$$MD = \frac{\sum_{t=0} k(t)}{N} \quad (8.28)$$

where N is the number of simulated intervals. Since the density is determined for upstream and downstream segments separately, the traffic density $k(t)$ in the network in an interval t is calculated as the weighted mean of densities on upstream and downstream segments:

$$k(t) = \frac{L_{upstr} \times k_{upstr}(t) + L_{downstr} \times k_{downstr}(t)}{L_{upstr} + L_{downstr}} \quad (8.29)$$

where L_{upstr} and $L_{downstr}$ are the lengths of upstream and downstream segments, respectively. In Section 8.4.3 the two segments respectively have the lengths of 1,000 m and 500 m, excepting for the Scenario "*High demand, Case 4*" where the incident location is assumed to move upstream, the length of the segments change, i.e. $L_{upstr} = 500$ m and $L_{downstr} = 1,000$ m.

Apart from the described measures, the simulation considers the maximum length of queues on the expressway Q_{exp} and on the ramp Q_r , explained earlier in Equations (8.17) and (8.20).

8.4.2 Control methods

I. No control

The *No-control* scenario indicates the situation when there is no control at the ramp merging. Simulated vehicles at the ramp accept or reject the gaps in the mainline. The *No-control* scenario is the base case for comparison of the effectiveness among algorithms. The TC of the *No-control* scenario is illustrated in Appendix D, Page D-5.

II. ALINEA

Since ALINEA is considered an efficient local-ramp control algorithm for monitoring the mainline traffic, in this experiment ALINEA is used to compare with FLC. ALINEA uses the measured occupancy at a loop detector downstream of the ramp, and regulates the ramp flow based on the difference between the measured occupancy and the optimal set point occupancy (Papageorgiou et al, 1991). The equation used to calculate the metering rate for time interval t is:

$$q_r(t) = q_r(t-1) + K_R [O_{opt} - O_{down}(t-1)] \quad (8.30)$$

where

$q_r(t)$ and $q_r(t-1)$: metering rates of the current and previous intervals, respectively.

O_{opt} : set point optimal occupancy, which is set to obtain optimal operation (Papageorgiou et al, 1991). O_{opt} is taken slightly lower than the critical occupancy O_{cr} , which can be determined from an inspection of the volume-occupancy diagram.

$O_{down}(t-1)$: occupancy downstream in the previous interval.

K_R : regulator parameter. Field experiment has shown that ALINEA has not been very sensitive to the choice of K_R , and the typical value of K_R is 70 veh/h (Papageorgiou et al., 1991).

Figure 8.18 plots the volume-occupancy relationship from 227 simulated records. From the figure, parameters of interest are obtained as capacity $C = 2,500$ veh/h; $O_{cr} = 26\%$. Therefore the O_{opt} used in this experiment is set at 24%, a little lower than the O_{cr} .

Figure 8.18: Determination of optimal set point occupancy

Since the standard ALINEA algorithm targets the optimal occupancy at the immediate detector downstream of the ramp, it uses the point measurement. The algorithm is not very useful for incident management since it does not actually consider the prevailing traffic condition on the upstream section of the incident. In case the incident location is far away from the ramp, the incident's impacts are not visible until the congestion propagates upstream to the detector. Therefore, in this experiment the average occupancy for the whole section from the incident location to the ramp, estimated from the average density, is recommended to capture the spatial effect of the incident. The average occupancy is inferred from density in Equation (8.31):

$$O_{down}(t) = (L + d) \times k(t) \quad (8.31)$$

where L is the average vehicle length, d is the length of detector. The average density $k(t)$ in each evaluation interval is calculated by the ratio between the

number of vehicles on the link and the length of the segment (see Block 2.1, Appendix D).

The average vehicle length is the arithmetic mean of lengths of vehicle types, which can be derived from the vehicle composition (Table 8.4). Equation (8.31) holds true when the vehicles have constant speeds. In congested condition (stop-and-go), this assumption is not valid, and Equation (8.32) will be used instead:

$$O_{down}(t) = \frac{\sum_i (L_i + d) / v_i}{T} \quad (8.32)$$

where L_i is the length of vehicle type i ; v_i is the vehicle speed; T is the period of measurement.

A drawback of the ALINEA control is that the algorithm solely aims at maintaining optimal occupancy downstream, regardless of the status of traffic on the ramp. In case high demands in both expressway and ramp are encountered, ALINEA incurs excessive long ramp queues. In the case of high ramp demand, the use of the control algorithm known as ALINEA/Q (Section 8.4.3) that incorporates mainline control with ramp queue management is recommended. In the evaluation, both ALINEA and ALINEA/Q control algorithms are used to compare with the FLC algorithm. The TC of the ALINEA and ALINEA/Q control algorithms are illustrated in Appendix D, Pages D-3 and D-4 respectively. The ALINEA and ALINEA/Q Controllers are presented in Appendix D, Pages D-9 and D-10, respectively.

III. FLC

For the evaluation, the FLC is embedded in the TSC model. The approach monitors the ramp flow by considering both the congestion level of the expressway and the ramp queues, with priority given to the mainline. Results from initial simulated scenarios were used to train the FLC before the actual evaluation: some of the membership functions need to be recalibrated to fit the single equivalent lane system. The TC of the control algorithm is illustrated in Appendix D, Page D-2. The FLC controller is elaborated in Appendix D, Page D-8.

8.4.3 Results and analysis

This section presents the results from the simulation experiment in different traffic and incident scenarios. In the scenarios, the terms *low*, *medium*, *high*, etc. in traffic demand are evaluated imprecisely based on the V/C^* ratio. Considering that a ramp has a limited length, in this experiment the ramp is assumed to have a storage capacity of 60 vehicles (see Appendix D, Block 2: TRAFFIC CONTROLLER). Once the ramp queue reaches this level, the urban traffic will not proceed to join the queue, but diverts to the urban network and enters the expressway through downstream ramps (see Appendix D, Page D-11). The availability of diversion alternatives encourages local traffic to utilize the parallel sub-network in case of critical mainline conditions.

Tables 8.8 to 8.14 show the values and percentile changes of the *MOEs*, which are described in Section 8.4.1. For the temporal *MOEs* (*TTT*, *TWT*, *TTS*), the negative sign of the percentile change indicates time saving, while in the spatial *MOEs* (*MD*, *max Q_exp*, *max Q_ramp*) the negative sign indicates the improvement, and in the remaining attributes (*TTD*, *MS*) the positive sign is a positive indication of the related parameter. Note that the evaluation does not include the warm-up time (the first 15 minutes), while the figures plot the results of the whole simulation period. In Figures 8.19 to 8.26, the horizontal axis indicates the simulation time (in second).

(i) Low-medium demand

The *low-medium* demand scenario is associated with the traffic demand of 700-800 veh/h on the mainline, and of $250 \pm 20\%$ veh/h on the ramp. The capacity remaining (C^*) at the incident location fluctuates in the range of 50-60% of the full capacity, which is approximately 2,500 veh/h from Section 8.4.2 (II).

The *MOEs* for this scenario is listed in Table 8.8. It appears from the table that all *MOEs* of the two control methods (*ALINEA* and *FLC*) are identical with those of *No control*. The reason is that since the mainline was in light traffic condition (the total traffic demand was low during the non-incident periods, and significantly below the remaining capacity (from 1,250 to 1,500 veh/h) during the incident), the ramp flows

recommended by the control algorithms were greater than the ramp demands (Figure 8.19), thus no ramp traffic was metered. There was no queue on the expressway and the ramp. From these, it could be inferred that under "low-medium" traffic condition, ramp control intervention might not be necessary.

Table 8.8: MOEs for low-medium demand

MOE	Unit	No control	ALINEA		FLC	
		value	value	% change	value	% change
TTT	veh.h	23.91	23.91	0.00	23.91	0.00
TWT	veh.h	0.00	0.00	-----	0.00	-----
TTS	veh.h	23.91	23.91	0.00	23.91	0.00
TTD	veh.km	1931.99	1931.99	0.00	1931.99	0.00
MS	km/h	80.80	80.80	0.00	80.80	0.00
MD	veh/km	12.40	12.40	0.00	12.40	0.00
max Q_exp	veh	0.00	0.00	-----	0.00	-----
max Q_ramp	veh	0.00	0.00	-----	0.00	-----

Figure 8.19 demonstrates some simulation results under *ALINEA*. The ramp demand fluctuated between 200-300 veh/h during the simulation. The occupancy upstream of the incident (downstream of the ramp) was low, from 8-9% (9-10% during the incident period), therefore the ramp rates recommended by *ALINEA* increased fast during the first two minutes from the initial rate of 250 veh/h to the ramp capacity (800 veh/h) to target the set point optimal occupancy O_{opt} , and this level sustained for the remaining time. The actual ramp rate that was set as the minimum between the recommended ramp rate and the ramp demand was equal to the ramp demand. There was no queue on the ramp.

Figure 8.19: Ramp control under *ALINEA*

(ii) Medium-high demand

The *medium-high* demand scenario is associated with the mainline traffic demand in the range 800-900 veh/h, the ramp demand in the range $350 \pm 10\%$ veh/h, and the remaining capacity $C^* = 40\text{-}50\%$. Table 8.9 summarises the results of the simulation for this scenario.

Table 8.9 shows that among the control alternatives *ALINEA* offered the most significant benefits for the mainline with a total travel time (*TTT*) saving of 7.31%, an increase in the mean speed (*MS*) of 7.89%, a decrease in the mean density (*MD*) of 7.31%, and a reduction in the maximum queue on the mainline ($\max Q_{exp}$) of 19.96%. The *FLC* method gained a *TTT* saving of 6.23%, an increase in the *MS* of 6.64%, a decrease in the *MD* of 6.23%, and a reduction in the $\max Q_{exp}$ of 18.9%.

Table 8.9: MOEs for medium-high demand

MOE	Unit	No control	ALINEA		FLC	
		value	value	% change	value	% change
TTT	veh.h	42.14	39.06	-7.31	39.52	-6.23
TWT	veh.h	0.00	4.29	-----	2.64	-----
TTS	veh.h	42.14	43.35	2.86	42.16	0.03
TTD	veh.km	2326.85	2326.85	0.00	2326.85	0.00
MS	km/h	55.21	59.57	7.89	58.88	6.64
MD	veh/km	21.73	20.15	-7.31	20.38	-6.23
max Q_exp	veh	89.16	71.37	-19.96	72.31	-18.90
max Q_ramp	veh	0.00	22.14	-----	15.00	-----

Nevertheless, *ALINEA* and *FLC* incurred 4.29 and 2.64 veh.h of ramp waiting time (*TWT*) respectively, which made the total times spent (*TTS*) slightly exceeded that of *No control*. Since the traffic states were similar across all control scenarios (there was no queue on expressway and ramp at the start (the 901st second) and end (the 5,400th second) of the evaluation period, the control methods have the same total vehicle mileage (*TTD*).

Figure 8.20 demonstrates the profiles of several key *MOEs* under *ALINEA* control. When the incident started at the 1,801st second, the queue formed and developed, leading to an increase in the mainline occupancy downstream of the ramp $O_{down}(t)$. As the occupancy was still less than the O_{opt} (Equation 8.30), the ramp rate at the maximum ramp capacity (800 veh/h) was recommended. When the occupancy was greater than the O_{opt} ($O_{down}(t-1) = O_{opt} = 24\%$ at the 2,850th second), the ramp rate proposed by *ALINEA* decreased accordingly.

The queue formed on the ramp at the 3,200th second when the recommended ramp rate became less than the ramp demand, and declined from the 3,600th second when the incident ended and the actual ramp rate raised up to the ramp capacity. From the 3,980th second there was no ramp queue, thus the actual ramp rate was set equal to the ramp demand.

Figure 8.20: MOEs for medium-high demand, ALINEA control

Figure 8.21 depicts profiles of aggregated traffic parameters under ALINEA control. During the incident, as the demand generally exceeded the remaining capacity, the queue on the mainline built up, the speed upstream gradually decreased since the upstream condition became worse while the speed downstream rapidly increased. At the same time, the density upstream increased gradually but the density downstream decreased considerably. Since the queue sustained on the mainline, the flow rate downstream was equal to the discharge capacity C^* (Equation 8.18). Once the incident was removed and the full capacity recovered (at the 3,600th second), the

flow rate increased shortly to the full capacity since the vehicles in the mainline queue were released.

Figure 8.21: Traffic parameters for medium-high demand, *ALINEA* control

(Yellow colour: upstream segment of the incident; red colour: downstream segment of the incident)

(iii) High demand

To investigate the properties of the control algorithms under various situations, the experiment was extended to the *high-demand* scenario. This scenario encompasses several cases in which the high level of mainline traffic demand is associated with various levels of ramp demand, capacity reduction, and incident location. More specifically, the following cases are investigated:

- Case 1: High expressway demand, medium ramp demand;
- Case 2: High expressway demand, high ramp demand;

Since the experiment focuses on congested conditions, Case 2 was extended to:

- Case 3 with more severe capacity reduction, and
- Case 4 with a change in the incident location.

Case 1: High expressway demand, medium ramp demand

Table 8.10 lists the results from the simulation where the mainline traffic demand changes in the range 1,000-1,100 veh/h, the ramp demand changes in the range $300 \pm 10\%$ veh/h, and the capacity remaining $C^* = 40\text{-}50\%$. The table shows that in general under both *ALINEA* and *FLC* more significant benefits were achieved compared to the *medium-high demand* scenario. *ALINEA* gained a *TTT* saving of 13.13%, an increase in the *MS* of 15.12%, and a reduction in the *MD* of 13.12%, compared to *No control*. The algorithm also enjoyed a substantial reduction in the *max Q_exp* of 32.38%. Nevertheless, *ALINEA* suffered considerable long *TWT* of 9.54 veh.h, and an excessive ramp queue (*max Q_ramp*) of approximately 46 vehicles.

Table 8.10: MOEs for high demand - Case 1

MOE	Unit	No control	ALINEA		FLC	
		value	value	% change	value	% change
TTT	veh.h	55.62	48.32	-13.13	48.36	-13.06
TWT	veh.h	0.00	9.54	-----	6.56	-----
TTS	veh.h	55.62	57.86	4.03	54.91	-1.28
TTD	veh.km	2541.43	2541.43	0.00	2541.43	0.00
MS	km/h	45.69	52.60	15.12	52.56	15.02
MD	veh/km	29.55	25.67	-13.12	25.69	-13.05
max Q_{exp}	veh	112.47	76.05	-32.38	77.62	-30.99
max Q_{ramp}	veh	0.00	46.54	-----	20.55	-----

The *FLC* obtained a compatible level of benefits: the improvements in the *TTT*, *MS*, and *MD* were 13.06%, 15.02%, and 13.05%, respectively. As compared to *ALINEA*, the *TWT* and *max Q_ramp* under *FLC* were less severe, which leads to a saving in *TTS* of 1.28% compared to a loss of 4.03% under *ALINEA*. Like the *medium-high demand* scenario, the *TTDs* were the same since the traffic states were similar across three control methods at the beginning and at the end of the evaluation period (there was no queue on the mainline and the ramp at these time points).

Simulation profiles of several key *MOEs* under *FLC* control are illustrated in Figure 8.22. It could be seen from the figure that there was an imbalance between the demand and capacity during the incident period: the flow rates upstream of the incident slightly declined due to a decrease in the controlled ramp traffic, but were greater than the flow rates downstream. This imbalance caused the queue that formed and grew on the mainline toward the end of the incident period (at the 3,600th second). The queue decreased and dissipated shortly as the expressway's capacity recovered. It could also be observed from the figure that during the incident the rate of increase in the *TTD* declined slightly while the rate of increase in the *TTT* rose remarkably, probably due to the decrease in the mainline speed.

Figure 8.22: Profiles of MOEs under FLC control - High demand, Case 1

(Flow rate: Yellow colour: upstream segment; red colour: downstream segment of the incident)

Case 2: High expressway demand, high ramp demand

To explore how the control algorithms work under critical conditions, the experiment was carried out with high demands on both expressway and ramp, where the mainline demand changes in the range 1,000-1,100 veh/h, the ramp demand fluctuates in the range $400 \pm 10\%$ veh/h, and the capacity remaining $C^* = 40\text{-}50\%$.

The results from Case 1 (Table 8.10) show that the standard *ALINEA* gained substantial benefits to the mainline, where the key *MOEs* such as *TTT*, *MS*, *MD* and *max Q_exp* were improved considerably. To some extent, *ALINEA* even slightly outperformed *FLC* control with respect to the mainline conditions (see Tables 8.8 and 8.9). Nevertheless, the *ALINEA* algorithm (described in Section 8.4.2 (II)) shows that the method merely targets benefits for the mainline without considering the status of the ramp traffic. In Tables 8.9 and 8.10, for example, the ramp conditions became worse with long ramp queue and ramp waiting time as the ramp demand increased. Under heavy ramp demands, the burden created by the standard *ALINEA* on the ramp traffic could be even intolerable.

The traffic control philosophy should be so that smooth expressway travel is achieved, and at the same time a reasonable ramp traffic status can be maintained. In incident management in particular, the control objectives should target efficient incident responses to the mainline without incurring excessive ramp queues. In this regard, the standard *ALINEA* should be improved so that the method can handle the ramp queue at an acceptable level.

ALINEA/Q (Smargdis and Papageorgiou, 2003) is an enhancement of the *ALINEA* algorithm. *ALINEA/Q* incorporates ramp control with ramp queue management by considering two metering rates. The first rate is calculated exactly the same as in the *ALINEA* algorithm. The second rate is calculated so as to maintain the ramp queue below a desirable queue length:

$$q'_r(t) = \frac{1}{T} [Q_r(t) - Q_r^{\max}] + q_r(t-1) \quad (8.33)$$

where $Q_r(t)$: Number of vehicles in the ramp queue in time period t .

$Q_r^{\max}$: Maximum desirable queue length on the ramp. In the subsequent Cases (Cases 2, 3, and 4), $Q_r^{\max}$ is assumed equivalent to 75% of the ramp storage capacity, i.e $Q_r^{\max} = 60 \times 0.75 = 45$ vehicles.

$q_r(t-1)$: Rate of traffic entering the ramp in the previous interval.

T : Time period over which the measurement is taken.

The control rate $q_r^*(t)$ is the maximum of the rates suggested in Equations (8.30) and (8.33).

$$q_r^*(t) = \max\{q_r'(t), q_r(t)\} \quad (8.34)$$

The *ALINEA/Q* control algorithm is presented in Appendix D, Pages D-4 and D-10.

Since Cases 2, 3, and 4 are associated with heavy ramp demand, the *ALINEA/Q* is proposed to be used instead of standard *ALINEA* for the comparative study.

Table 8.11 summarises the results of the simulation for Case 2 "*high mainline demand, high ramp demand*". The table shows that both *ALINEA/Q* and *FLC* control methods achieved considerable improvements: *ALINEA/Q* gained a *TTT* saving of 13.92%, an increase in the *MS* of 15.61%, and a decrease in the *MD* of 13.50%. In particular, *ALINEA/Q* handled the ramp queue better than *FLC* and slightly better than the standard *ALINEA* under the *medium ramp demand* Scenario (Table 8.10).

Table 8.11: MOEs for high demand - Case 2

MOE	Unit	No control	ALINEA/Q		FLC	
		value	value	% change	value	% change
TTT	veh.h	70.31	60.52	-13.92	55.14	-21.58
TWT	veh.h	12.91	22.62	75.24	24.71	91.40
TTS	veh.h	83.22	83.15	-0.09	79.85	-4.05
TTD	veh.km	2728.41	2715.12	-0.49	2719.53	-0.33
MS	km/h	38.80	44.86	15.61	49.32	27.10
MD	veh/km	37.42	32.36	-13.50	29.44	-16.70
max Q_exp	veh	153.87	135.22	-12.12	92.73	-39.73
max Q_ramp	veh	33.40	45.00	34.73	50.00	49.70

The *FLC* control alternative, with an exception of the ramp-related attributes, gained higher benefits than *ALINEA/Q*: the improvements in *TTT*, *MS*, and *MD* were 21.58%, 27.1%, and 16.7%, respectively. In particular, *FLC* also gained a reduction in the *TTS* of 4.05%.

Figure 8.23 gives an example that explains how the ramp flow is controlled by *ALINEA/Q* in response to the ramp queue Q_r . Under the below-capacity condition (pre-incident and post-incident periods), the recommended ramp flows were set to the ramp capacity. At the onset of the incident, the occupancy upstream of the incident started to increase.

Figure 8.23: Simulated profiles by *ALINEA/Q* - High demand, Case 2

The recommended ramp rate by standard *ALINEA* (q_r) decreased when the mainline occupancy exceeded the O_{opt} at the 2,200th second. The second element of

the *ALINEA\Q* ramp flow (q'_r , Equation 8.33) was effectuated when the ramp actual queue exceeded the maximum desirable ramp queue $Q_r^{\max}$ at the 2,850th second. The ramp rate proposed by *ALINEA\Q* (the 3rd panel from top) is the maximum of the two elements q_r and q'_r , while the actual ramp rate (the 4th panel from top) is the minimum of the proposed ramp rate, the ramp demand, and the ramp capacity (see Appendix D, Block "Derivation of actual ramp rate q_{r_r} ", Page D-4). As the proposed ramp rate was less than the ramp demand, queue built up on the ramp up to $Q_r^{\max}$. This level was maintained during the incident. The ramp queue was released when the mainline occupancy decreased and the actual ramp rate was greater than the ramp demand at the 4,250th second.

Case 3: High expressway and ramp demands, severe capacity reduction

Results from Case 1 and Case 2 show that there exist excessive long queues on the mainline. In Table 8.11 in particular, the expressway queues under *No control*, *ALINEA\Q*, and *FLC* were 153.87, 135.22, and 92.73 vehicles respectively. This is partially attributed to the implicit assumption that the ramp closes only when the mainline queue reaches the ramp. If the incident occurs far from the ramp, this passive type of ramp closure will tolerate a very severe mainline condition. It should be noted that if a long queue exists on the mainline, additional discharge from the ramp may not benefit the ramp traffic but aggravate the mainline conditions, thus a longer time will be required for the mainline traffic to dissipate. From the control perspective, an active action of ramp closure should be taken to avoid severe congestion. Therefore, in Case 3 under *ALINEA\Q* and *FLC* the ramp closure is set when the mainline queue reaches 50% of the length of the upstream-incident segment, while this feature of operation is not available under *No control*.

Table 8.12 lists the results from the simulation where the mainline demand changes in the range 1,000-1,100 veh/h, the ramp demand changes in the range $400 \pm 10\%$ veh/h. The incident is assumed to create a more severe capacity reduction (capacity remaining $C^* = 30\text{-}40\%$). The table shows that benefits of *ALINEA\Q* and *FLC* obtained for the mainline in this Case were, in general, higher than the previous Cases. Compared to *No control*, *ALINEA\Q* gained a *TTT* saving

of 22.14%, an increase in the *MS* of 26.82%, a reduction in the *MD* of 23.44%, and a cut down in the *max Q_exp* of 41.86%. The *FLC* benefits were even more profound with improvements in *TTT*, *MS*, *MD*, and *max Q_exp* of 23.13%, 27.98%, 23.11%, and 42.61%, respectively. The improvements of *ALINEA/Q* and *FLC* were certainly due to a strong regulation of the ramp traffic with active response to the mainline conditions. The results under *No control* also indicate that without strong control intervention, the system performances may deteriorate seriously. Despite that, with the early ramp closure subjected to the mainline queue, it is certain that *ALINEA/Q* and *FLC* impose more *TWT*, and more vehicles have to divert from entering the ramp.

Table 8.12: MOEs with high demand - Case 3

MOE	Unit	No control	ALINEA/Q		FLC	
		value	value	% change	value	% change
TTT	veh.h	71.00	55.28	-22.14	54.57	-23.13
TWT	veh.h	20.28	25.75	26.99	23.90	17.86
TTS	veh.h	91.28	81.04	-11.22	78.47	-14.03
TTD	veh.km	2543.77	2511.95	-1.25	2502.36	-1.63
MS	km/h	35.83	45.44	26.82	45.85	27.98
MD	veh/km	37.89	29.01	-23.44	29.14	-23.11
max <i>Q_exp</i>	veh	181.85	105.72	-41.86	104.37	-42.61
max <i>Q_ramp</i>	veh	60.00	60.00	0.00	60.00	0.00

Figure 8.24 plots some profiles of ramp control under *FLC*. During the pre-incident period, the *FLC* recommended ramp rates (typically in the range 650-750 veh/h) were greater than the ramp demands. Therefore, the actual ramp rates (the 3rd portion of the Figure) were equal to the ramp demands of $400 \pm 10\%$ (veh/h). During the incident the recommended ramp flow decreased toward the end of the incident period since the queue propagated gradually upstream and the mainline condition became worse. When the expressway queue reached the mid-point of the upstream-incident link (*max Q_exp* = 500m, at the 2,420th second), the ramp was closed to avoid further risk to the mainline. Given the ramp closure, the ramp queue developed fast to the maximum ramp storage (60 vehicles) at the 3,000th second. From the 2,420th to the 3,600th second the expressway queue continued to propagate

upstream due to the imbalance between demand and capacity but at a lower speed since the demand was not contributed by the ramp traffic. When the incident ended, the capacity recovered and the mainline queue decreased (from the 3,600th to 4,000th second) the *FLC* recommended ramp rates increased sharply to the ramp capacity of 800 veh/h. From the 4,200th to the 4,450th second the recommended ramp rates remained higher than the ramp demands to take account of additional discharge from the ramp queue. From the 4,450th second, when the ramp queue dissipated, the actual ramp rate reduced to the ramp demand of $400 \pm 10\%$ veh/h.

Figure 8.24: Profiles of ramp control under *FLC* - High demand, Case 3

As opposing to the active mode of ramp closure under *ALINE/Q* and *FLC*, the passive mode of ramp closure was applied to *No control*, illustrated in Figure 8.25. The ramp rate under *No control* is straightforward: without ramp control intervention, all ramp traffic merged the mainline until the expressway queue

reached the ramp at the 2,885th second (the corresponding queue length was approximately 153 vehicles). When the ramp was closed, the ramp queue formed and extended quickly to the maximum ramp storage capacity of 60 vehicles. When the incident was over and the queue on the expressway was shortened downstream of the ramp (the 3,670th second), the ramp traffic continued to contribute to the mainline. Since the sum of the mainline and ramp traffic was less than the expressway capacity, the ramp rate was set to the ramp capacity. The maximum ramp rate sustained until the ramp queue was dissipated at the 4,450th second, then was reduced to the ramp demand of $400 \pm 10\%$ veh/h.

Figure 8.25: Profiles of *MOEs* under *No control* - High demand, Case 3

Case 4: High expressway demand, high ramp demand, more severe capacity reduction, incidence location changed

Case 4 is associated with the mainline demand in the range 1,000-1,100 veh/h, the ramp demand in the range $400 \pm 10\%$ veh/h, and the capacity remaining $C^* = 30\text{--}40\%$. The incident occurred 500 m downstream of the ramp, being less distant than the previous Cases. Table 8.13 summarises the results from the simulation. The table shows that the benefits from *ALINEA\Q* and *FLC* were less profound than the previous Cases: *ALINEA\Q* gained a *TTT* saving of 6.49%, an increase in the *MS* of 5.6%, a reduction in the *MD* of 6.72%, and a reduction in the *max Q_exp* of 13.97%. The improvements in *TTT*, *MS*, *MD*, and *max_q_exp* under *FLC* were 11.34%, 11.88%, 13.13%, and 19.69% respectively, that are remarkably higher than *ALINEA\Q*. Nevertheless, *ALINEA\Q* and *FLC* incurred 22.27% and 10.19% more of *TWT* than *No control*, respectively. In particular, the two control algorithms yielded 1.25% and 0.81% of the total mileage *TTD* less than *No control*, although there was no ramp queue at the end of the simulation time (see Figure 8.26). This is probably due to the fact that when the ramp queue reaches the ramp's physical storage capacity, the vehicles arrive at the ramp will not proceed to join the queue, but divert to the parallel street, described in Appendix D, Page D-11.

Table 8.13: MOEs for high demand - Case 4

MOE	Unit	No control	ALINEA\Q		FLC	
		value	value	% change	value	% change
TTT	veh.h	57.41	53.68	-6.49	50.89	-11.34
TWT	veh.h	23.05	28.18	22.27	25.40	10.19
TTS	veh.h	80.45	81.86	1.75	76.29	-5.18
TTD	veh.km	2509.66	2478.26	-1.25	2489.29	-0.81
MS	km/h	43.72	46.16	5.60	48.91	11.88
MD	veh/km	30.72	28.65	-6.72	26.68	-13.13
max Q_exp	veh	127.19	109.42	-13.97	102.14	-19.69
max Q_ramp	veh	60.00	60.00	0.00	60.00	0.00

Figure 8.26 demonstrates the profiles of simulation results for a number of *MOEs* under three control alternatives: *No control*, *ALINEA\Q*, and *FLC*. The simulation

horizon can be classified into three periods: pre-incident (from the 1st to the 1,800th second, incident (from the 1,801st to the 3,600th second), and post-incident (from the 3,601st to the 5,400th second). It could be seen from the Figure that the rates of increase in the *TTDs* were lower during the incident due to the decrease in the throughput (reduced capacity) and speed. In contrast, the rates of increase in the *TTTs* were higher during the incident period, primarily due to the significant increase in travel time of individual vehicles.

Figure 8.26: Profiles of simulation results - High demand, Case 4

The profiles of the mainline queue can be classified into several segments, the rates of which depend on the imbalance between demand and capacity, and the rate of the ramp traffic that was metered into the expressway (see the 4th panel from top):

Under *No control*, the ramp rate was equal to the actual ramp demand of $400 \pm 10\%$ (veh/h) in the pre-incident period, then reduced to zero when the ramp was totally closed as the mainline queue reached the ramp at the 2,200th second; under *ALINEA\Q*, the ramp rate successively decreased from the actual ramp rate as the mainline occupancy increased, then totally closed as mainline queue reached the ramp at the 2,350th second. On the other hand, *FLC* acted more actively by gradually decrease the ramp rate in response to the development of the mainline queue. The ramp closures of three control methods remained until the mainline queue was shortened downstream of the ramp (at the 3,850th second under *No control*, and the 3,800th second under *ALINEA\Q* and *FLC*). From these time points, the ramp was metered according to the control algorithms as explained earlier in Section 8.4.2.

The ramp queue formed when the actual ramp rate was lower than the ramp demand, at the 2,200th, 2,070th, and 2,250th second under *No control*, *ALINEA\Q*, and *FLC* respectively. The rate of change of the ramp queue under *No control* was constant in each sub-segment due to the sudden changes of the actual ramp rates given the (fairly) constant ramp arrivals, while that under *ALINEA\Q* and *FLC* were not constant since the actual ramp rates changed gradually. The maximum ramp queue sustained until the time the actual ramp rates were greater than the ramp demand (at the 3,850th second under *No control*, and the 3,800th second under *ALINEA\Q* and *FLC*). The ramp queues dissipated at the 4,150th, 4,300th, and 4,400th second under *No control*, *ALINEA\Q*, and *FLC* respectively.

8.4.4 Uncertainty analysis

Section 8.4.3 presents the results from the simulation experiment in different traffic demand (*low, medium, high*) and incident scenarios (*capacity reduction, incident location*). Nevertheless the scenarios were associated with fixed hypothetical network geometries (the network length of 1.5 km (upstream section = 1.0 km, downstream section = 0.5 km) and the ramp storage capacity of 60 vehicles), and with the simulation time of 90 minutes. These network and simulation parameters have significant implications on the model performances, and it is uncertain that the

comparative performances of the control methods are still valid if the input parameters change.

To investigate the impacts of variations in these parameters on comparative performances of the control methods and to increase confidence in the model performances in an uncertain environment, a uncertainty analysis is performed. Since mathematical equations that describe the relationships between the model inputs and outputs are not available, running of the models many times is the only way to obtain outputs from the respective inputs. Since these input parameters are non-correlated, the uncertainty analysis is performed for each parameter individually such that in each run one parameter is changed while the other parameters remain constant.

As described in Section 8.4.1(v), there are many quantities (MOEs) that can be used simultaneously to measure the model performances. The use of all MOEs in this analysis would be very confusing, thus it would be hardly to obtain a clear understanding of how the model outputs vary with changes in its inputs. The mean traffic speed could be the best MOE in this uncertainty analysis. The followings are primary reasons:

- The mean speed directly describes the relationship between the total vehicle mileage (TTD) and the total vehicle travel time (TTT), being two other primary MOEs.
- The mean speed is a key parameter that reflects the operational condition on the mainline. Apart from TTD and TTT, it has direct links with the other operational quantities such as the mean density (MD) and the average travel time.

Given these, the mean speed on expressway (more precisely: the relative change in the mean speed) is proposed as the single candidate of MOEs to be considered in this analysis. Since it is easier to detect the comparative performances of various control methods using the relative performances, the relative change in the mean speed (Δ_{MS}) of the control methods over *No control* is used and is calculated as:

$$\Delta_{MS}^i = 100 \times \frac{MS_i - MS_{No}}{MS_{No}} (\%) \quad (8.35)$$

where i denotes either *ALINEA\Q* or *FLC* method, MS_i denotes the mean speed under the control method i , and MS_{No} denotes the mean speed under *No control*. If Δ_{MS} is positive, the change indicates an improvement in the mean speed, and vice versa.

The uncertainty analysis is performed for the scenario "*high mainline and ramp demands*", where the mainline demand changes in the range 1,000-1,100 veh/h, the ramp demand fluctuates in the range $400 \pm 10\%$ veh/h, and the capacity remaining $C^* = 40\text{-}50\%$. The active mode of ramp closure is applied for *ALINEA\Q* and *FLC* control methods (the ramp is assumed to be closed as the mainline queue reaches 50% of the upstream-incident section), while in *No control* the passive mode of ramp closure is applied.

(i) Network Length

The simulated mainline consists of the upstream and downstream sections of the incident. Since the impacts of the incident can mostly be observed upstream, this analysis investigates how the Δ_{MS} change with a change in the length of the upstream section. Four scenarios are extended to that in Section 8.4.3 with the length of the upstream section increased from 1.0 km to 1.5, 2.0, 2.5, and 3.0 km, respectively. The length of the downstream section in the four scenarios remains 0.5 km.

Figure 8.27 plots the percentage changes of the mean speed (Δ_{MS}) in *ALINEA\Q* and *FLC* in comparison to *No control*, in different Network Length scenarios. The figure indicates that Δ_{MS} in both *ALINEA\Q* and *FLC* are highly sensitive to the length of the upstream section, and the superiorities of the control methods over *No control* deteriorate as the network length increases: For a relatively short simulated network (the length of the upstream section = 1.0-1.5 km), a small change in the

network length may lead to a large change in Δ_{MS} , but for a relatively long simulated network (the length of the upstream section = 2.5-3.0 km) the change in Δ_{MS} against a change in the network length is smaller. The Δ_{MS} of *ALINEA\Q* and *FLC* in the 3-km upstream section scenario are only 7.42% and 12.45% compared to 24.90% and 27.47% respectively in the 1-km upstream section scenario.

Figure 8.27: Δ_{MS} versus the length of upstream section

A possible reason under this phenomenon could be due to the fact that for a given traffic demand and incident parameters, when the upstream section is shorter, the traffic condition is more critical. The *ALINEA\Q* and *FLC* algorithms are designed to alleviate mainline congestion by regulating the ramp flow, therefore the effectiveness of control in a more critical mainline condition is higher. By contrast, when the network length is large the traffic condition is less severe, and the effectiveness of the control is lower.

(ii) Ramp storage capacity

Another important network configuration is the storage capacity of the ramp. To investigate the impacts of the physical ramp constraints on the model performances, the ramp storage capacity of 60 vehicles in Section 8.4.3 is changed to 20, 40, 80 and 100 vehicles, respectively. Although the ramp storage capacity of 100 vehicles is very rare, the scenario is extended so as to explore how the Δ_{MS} varies over a possible physical coverage of a ramp. Figure 8.28 depicts the change in the Δ_{MS} as the ramp storage capacity changes in the range from 20 to 100 vehicles.

Figure 8.28: Δ_{MS} versus the ramp storage capacity

Figure 8.28 shows that in both control methods the Δ_{MS} varies slightly in the range 23-28%, and the values of Δ_{MS} increase as the ramp storage capacity increases. A possible reason could be due to the fact that when the ramp storage capacity increases, the ramp can accommodate more vehicles, hence fewer vehicles have to divert from the ramp. Consequently, given a long ramp and regardless of the control

method, more vehicles can be metered into the mainline. However, while in *No control* the passive mode of ramp closure is applied, in *ALINEA\Q* and *FLC* the active mode of ramp closure is employed that activates the ramp closure as soon as the mainline queue reaches a critical level (such as 50% of the upstream section). Given this, in control algorithms under critical mainline conditions fewer ramp vehicles are metered and the mainline condition can be maintained better than in *No control*, for the same ramp storage capacity scenario. Therefore, the Δ_{MS} of the control algorithms increase as the ramp storage capacity increases.

(iii) Simulation time

Simulation time, or more precisely, the temporal structure of the simulation, could be another important parameter that has significant impacts on the model outputs. The previous simulation (Section 8.4.3) investigated the model performances for the simulation time of 30 minutes for each of the pre-incident, incident, and post-incident periods (named hereafter as scenario "30-30-30"). To explore how the improvement in the mean speed changes with simulation time, in this analysis the simulation time is extended to two scenarios 30-60-30 and 30-60-60 minutes, respectively. The evaluation times for the scenarios 30-30-30, 30-60-30, and 30-60-60 are 75/90, 105/120, and 135/150 minutes respectively (excluding 15-minute warm-up period).

Figure 8.29 illustrates the Δ_{MS} for the three scenarios. The figure indicates that in both control methods, the benefits in the mean speed are highest in the simulation 30-60-30 (31.93% and 37.47% for *ALINEA\Q* and *FLC* respectively), followed by the simulation 30-60-60. The ratios of the incident and non-incident period in the evaluation period are 30/45 (0.67), 60/45 (1.33), and 60/75 (0.80), respectively. This indicates that when the ratio of the incident and non-incident period is higher, the improvement in the mean speed of the control algorithms over *No control* increases. This coincides with the findings in (i) and (ii) that the effectiveness of control is higher in more critical mainline conditions.

Figure 8.29: Δ_{MS} versus the simulation period

8.5 EXTENSION: PROPOSED KB-DSS ARCHITECTURE

8.5.1 Discussion

Although the development of the User Interface is not in the scope of this research, this section discusses issues involving functional integration between components of the KB-DSS for corridor traffic control. Since most of the traffic monitoring and control routines for an expressway network are implemented at the Traffic Management Centre (TMC), the KB-DSS architecture should be designed to assist TMC personnel in corridor control. In previous chapters, however, the presented MS-FLC is designed to react with local impacts due to an incident. Although the local control accounts for a majority of applications, the corridor control requires coordination of controllers along expressway corridors.

Figure 8.30: Corridor MS-FLC architecture

Figure 8.30 illustrates the proposed architecture of a corridor control following the fuzzy logic approach. An expressway corridor is divided into N sections, each of which has a local FLC to manage traffic at the corresponding section. The local FLCs have their own rule bases, while the CW-FLC, known as the coordinator, has specific rules to govern the coordination between FLCs.

In normal conditions the FLCs work independently of the CW-FLC. Under special conditions, the FLC immediate downstream and the FLCs upstream of the incident form a *control group*, which is taken over by the CW-FLC with coordinated rules activated. The immediate downstream FLC is used to encourage the utilisation of available capacity downstream of the incident, while the immediate upstream FLC is the prime controller that has the most direct and primary impacts to the control scheme. In principle, each local FLC issues control actions based on the measurements in its upstream and downstream segments, but adjusted to control directives from the CW-FLC, which passes corridor information on individual segments along the corridor, and receives the local measurements as well as control actions from individual FLCs. The rule base of the CW-FLC involves mechanisms that make coherent integration between rules in the local FLCs, such as rules to increase or reduce flows in several ramps subjected to the status of the other ramps in the control group so that corridor objectives are obtained. In case conflicts or incoherence between FLCs appear, the CW-FLC coordinator dissolves the problems by setting priorities over local FLCs.

8.5.2 Components of a KB-DSS for traffic control

Like any conventional DSS, the KB-DSS for traffic control includes three major components: the data base, the model base and the user interface (Figure 8.31).

Figure 8.31: Proposed architecture of the KB-DSS for traffic control

(i) The data base

Unlike the databases presented in Chapter 4 in which the data collection requires network transmission with a client-server architecture, the databases in this KB-DSS may be located at the TMC's workstation. The database component consists of HDB and RTDB databases with functionalities described in Chapter 4. In particular, HDB is unique in the sense that it is a global storage of facts from that the knowledge in the form of articulated expertise is acquired. The HDB also provides resources for SVM training and operator's references in case real-time data is missing or contaminated. The RTDB, on the other hand, provides raw data as direct

inputs for the MS-FLC in the evaluation of traffic conditions and for SVM in traffic forecasting. As presented in Chapter 4, the RTDB is encoded so that only relevant data within the rolling horizon is maintained to reduce the cost of data transaction and manipulation. Since the traffic and incident data are characterized by homogeneous attributes, and the relational model provides great advantages of database administration, the HDB and RTDB contain a number of relational tables. The database management system (DBMS) administers the databases by techniques such as SQL that create, modify, retrieve, and maintain data. The SQL query also allows mapping between tables as required.

Depending on the size of the network, the RTDB can be either centralized or localized. The centralized RTDB is a globally shared database, where data is collected and stored at the TMC's workstation that administers and makes data accessible to all FLCs in the network. The centralized model has an advantage that each FLC has access to a complete set of data, thus a global outlook of traffic conditions prevailing in the network can be obtained. However, administering data in this manner requires extensive data manipulation and may incur delays by having concurrent access from all FLCs. In the localized-database administration, the network-wide data collected from detectors are partitioned into localized databases, each of which covers a single expressway corridor and the associated urban sub-network. Each FLC control group is dedicated with an exclusive access to the relevant localized database. This option allows a large reduction of data processing having a control group of FLCs processes data locally. Furthermore, a localized-database preserves control over its portion of data set, thus the local autonomy is enhanced. Since the traffic data in expressways are relatively independent from each other, and the local or coordinated control are primary control styles, for a large network the localized-database administration is recommended.

(ii) The model base

The model base component consists of the fuzzy rule base, the SVM traffic prediction package, and the TSC.

a) The fuzzy rule base is the repository of rules used in the MS-FLC model. Part of the rules are extracted from the HDB through the rule extraction process, as described in Chapter 4, while a substantial number of rules can be established using engineering knowledge and commonsense reasoning. Owing to the fact that the MS-FLC reasoning is a data-driven process, most of the rules adopt forward-chaining sequence that matches the antecedents of rules against available facts to evaluate traffic state or control action. The fuzzy rule base is constructed off-line and embedded into the MS-FLC in on-line implementations.

For the KB-DSS, rules should be designed to support all decision-making phases. Rules at strategic level such as determination of control strategy (Figure 7.5) are used to support decisions made by traffic operators. Strategic rules are not inclusively encoded in the rule base, but are presented to the operator by means of interactive dialogs in the GUI in advance of the control implementation. By contrast, control rules are at an operational level, encoded in the FLC, and then embedded in the TSC. The TSC and SVM execution starts as soon as the control strategy is decided and associated controlled ramps are selected by traffic operators.

b) The SVM module predicts short-term traffic variables required by the FLC. The SVM is linked with HDB and RTDB so that data can be continually retrieved for its operation. For SVM training, the empirical NN method (Section 6.2.7) can be used to locate the most similar traffic patterns from the HDB within the desirable rolling horizon (Ω). By using the NN method, the SVM operation is accelerated so that stringent synchronisation requirements of the complex system can be met. Namely, the result from the prediction should be available before the timeline required by the MS-FLC in each interval.

SVM mode of operation

In a continually changing traffic environment, the traffic monitoring and control system must respond quickly and reliably to changes in incoming data. Since there are still unknown factors in traffic patterns and variation in travel behaviour, the predicted travel demands usually incur errors. The rolling horizon approach

proposed by Peeta and Mahmassani (1995) provides potential to reduce the uncertainty of traffic prediction routines.

In the rolling horizon approach (Figure 8.32) the time window (T) is divided into a number of stages, each consists of short intervals. In each stage, data in the rolling horizon Ω (Section 6.2.2) are available. The prediction interval Δ may be as short as a few minutes to as long as 60 minutes, however, in dynamic changing incident conditions, the long intervals are usually susceptible to large errors, thus intervals of less than 15 minutes are recommended, as is drawn from Sections 6.2 and 6.3.

Figure 8.32: The rolling horizon approach in SVM prediction

Using the short-term predicted traffic data, the MS-FLC estimates the ramp flow that can be accommodated in the next interval, and then the time horizon is forwarded by the rolling step (δ) to the next stage. The rolling step should be selected in correspondence with the granularity of the collected data. In this new

stage the SVM uses the updated actual data collected by traffic surveillance systems. The cycle is repeated until the incident traffic congestion is dissolved.

In the control implementation, the ramp flow for the first stage may be calculated with the real-time data since the situation is acute. For subsequent stages, the TSC uses the predicted traffic data for proactive control. If θ indicates the prediction time required by SVM, the input data must be available θ seconds prior to the beginning of every stage to provide output for the TSC control. For this reason, the applicability and the effectiveness of the control system are highly dependent on the granularity and the quality of the data collection system.

c) The TSC supports the operator's decision making at the operational level. It is mapped with the RTDB and SVM model for the real-time and predicted data as inputs. Given the selected control strategy, the TSC calculates the control settings and supplies the results to the operators. The output from the TSC (ramp flow rate) is used to calculate the signal settings at the ramp entrance. The TSC and SVM operations are both monitored by the operator through the GUI interface.

(iii) The user interface

The user interface component provides the traffic operator with a communication means to integrate with the other components of the KB-DSS. To improve communication and presentation, an integrated Graphical User Interface (GUI) is needed. With the GUI interface, the two-way communication between the operator and the computer can be greatly enhanced. GUI offers various types of dialog, including questions and answers, menus, commands, input-output streams, and natural language interface, etc. Among these, menu is the most popular method of dialogs, where a series of choices is presented for the user's selection. Importantly, GUI facilitates graphical displays that make the man-machine interaction more user-friendly and efficient than traditional textual interfaces.

A standard GUI for traffic control should provide adequate functionalities for efficient communication. The key elements of the GUI should involve:

- A platform on which icons are placed
- Windows that house applications and documents
- Folders that store other folders and external documents
- Menus that allow the operator to make choices and provide feedbacks.

Figure 8.33: A template of GUI for traffic control

Figure 8.33 illustrates a GUI template for traffic control, developed in Visual Basic 6.0. The GUI contains a main control desktop with buttons for loading forms with specific functions, including databases, maps, camera locations, web services, fuzzy rule base, simulation, and external files from Window applications.

8.6 SUMMARY AND COMMENTS

This chapter presented the development and validation of the TSC model, and evaluation of the FLC embedded in the TSC framework. The conceptual model of TSC (Figure 8.1) consists of the CFM and the TC, and has been converted into the implementation model with great details. The implementation model follows the

model logic and has been verified by comparing the computer representation with the conceptual model to ensure that a right model has been built.

The CFM simulates the car-following behaviour and delivers the aggregated traffic parameters to the TC for traffic control. As presented in Section 8.1, the CFM does not capture lane-changing behaviour, having considered that the mainline traffic has little opportunity for lane changing under congested conditions, and traffic from the ramp adapts to the gaps in the mainline. Without capturing the lane-changing behaviour, the parameters of the CFM were calibrated through an iterative process of comparing the model with data from the HDB.

Having been calibrated, the model is validated by fitting to observed data in another segment of PIE expressway. The result shows that aggregated speed and volume discrepancies fall in small ranges. Ideally, the validation process should be conducted for a variety of network geometries. Nevertheless, limited by time constraints, the validated segment and traffic conditions have been chosen carefully for adequate representation so that the model can be used for experimental purposes. The model validation should not be considered to be an isolated but integral part of the model development process, and subsequent steps of validation should be carried out in later stages for actual applications.

The validated TSC model has been used for evaluating the FLC controller in comparison with the *No-control* scenario and *ALINEA* ramp control algorithm, conducted for a variety of traffic conditions and incident scenarios. For evaluation of performance, the most important MOEs such as travel time, travel distance, speed, density, and queue length, have been considered. The results from the experiment shows that under low and medium traffic, the control intervention provided by *ALINEA* and FLC algorithms bring no significant benefit, and even imposed a ramp queue in the case of *ALINEA*. By contrast, under high traffic demands, the control models (*ALINEA/Q* and FLC) showed substantial benefits. Particularly, under high traffic demand and severe capacity reduction, the FLC brings higher travel time savings as well as improvements of traffic conditions on both the mainline and ramp. Not only do the FLC outperform *ALINEA/Q* in managing ramp traffic, it also outperforms *ALINEA/Q* in managing the mainline

flow under critical incident congestion. It should be noted that this comparison is based on a simplified network (single lane equivalent expressway, one-lane ramp) even though the ramp traffic has been reduced accordingly. In the reality the mainline usually has 3-4 lanes, and the ramp may have 2 lanes, the benefits (savings in travel times, distances, etc) should be adjusted accordingly.

Since the simulation Scenarios in Section 8.4.3 were conducted for fixed network geometries and simulation time, the model evaluation was extended to a uncertainty analysis in Section 8.4.4 to investigate the impacts of variations in these parameters on the model outputs. The key findings from the uncertainty analysis are summarised in Chapter 9.

CHAPTER 9 CONCLUSIONS AND RECOMMENDATIONS

9.1 SUMMARY AND CONCLUSIONS

The development and validation of a fuzzy KBS for incident management on expressways following a fuzzy logic approach have been presented in the previous chapters. The essence of the fuzzy KBS is the Multi Stage-Fuzzy Logic Controller (MS-FLC), which is decomposed into three stages in deriving control actions. The fuzzy KBS targets assisting traffic operators in decision making regarding non-recurring congestion management in a systematic and structured manner.

The methodological overview of the fuzzy KBS for incident management was presented in Chapter 3. The decision-making process for traffic control during incidents on expressways considers three major tasks: (i) the evaluation of incident traffic conditions, (ii) the prediction of congestion tendency during the incident, and (iii) the recommendation of local control strategies and control actions to alleviate the congestion. Following this logic, the multi-stage composite structure of the FLC was proposed. The MS-FLC consists of three stages corresponding to the three aforementioned tasks where rules are executed sequentially from one stage to another.

The preliminary step in the development of the fuzzy KBS, the data collection and analysis, was presented in Chapter 4. The HDB and RTDB for traffic and incident data with the focus on the study area on PIE were established. The chapter addressed important issues in designing membership functions including the determination of the universes of discourse, fuzzy partitioning, and fuzzy rule generation. Various methods of fuzzy partition using engineering knowledge, grid clustering, and fuzzy C-Means clustering were explored. The concept of fuzzy rule generation that develops FIS systems using the framework of adaptive neural networks was investigated.

The first stage of the MS-FLC - the evaluation of prevailing traffic conditions - presented in Chapter 5, is characterised by a thorough assessment of traffic conditions of real-time data, attributed to congestion level, congestion mobility, and congestion status.

The second stage - presented in Chapter 6, involves the prediction of short-term traffic and incident conditions. In the face of random fluctuation during incidents, this stage confronts a challenging issue in traffic forecasting. For this reason, a considerable effort has been made to explore SVM, an advanced technique in pattern recognition. Short-term predictions of traffic volume and travel time have been investigated and the temporal behaviour of prediction and the effect of rolling horizon have been explored. In particular, an empirical NN method that accelerates SVM training but maintains the prediction accuracy has been proposed. The predicted traffic volume is used in the MS-FLC model for prediction of congestion trend, while the forecasted travel time is essential for ATIS systems in recommending travel routes. Since incident traffic conditions are less predictable and the accuracy of traffic forecast may decline, the results from automated forecast were adjusted by introducing a risk factor to cater for unknown and unexpected impacts from traffic environment.

The third stage of the MS-FLC, presented in Chapter 7, describes the decision-making logic for local control actions. The stage receives inputs from the previous stages and other relevant information to recommend solutions. The decision-making logic is a general-to-specific process and is represented by three blocks: (i) intervention level, (ii) control strategy, and (iii) control action. The first two blocks set rules at strategic level, presented to the operator for his decision, while the last block translates the selected control strategy into control implementation at the operational level.

Chapter 8 presented the development and validation of the TSC model, and the evaluation of the MS-FLC. The TSC consists of two main components: the CFM and the TC. Theoretical concepts behind the car-following behaviour have been utilised to develop the CFM. In calibrating the CFM, through fitting the model with observed data on a number of segments of PIE, the most influencing parameters

have been identified and adjusted. The model validation shows that the calibrated CFM has errors within small ranges. Finally, in the simulation experiment, the MS-FLC performances were compared with ALINEA, the most efficient local ramp control algorithm. Important evaluation criteria include travel time, waiting time on-ramp, total travel distance, mean speed, mean density, and queue length. The experiment evaluated the control algorithms under various traffic demand levels and incident scenarios. A uncertainty analysis is performed to investigate the impacts of variations in the input parameters on the model outputs, and to increase confidence in the model performances in an uncertain environment.

The advantages of the MS-FLC

A review of the MS-FLC identifies the following advantages:

- The MS-FLC provides a systematic procedure in deriving control actions. Through the systematic assessment of prevailing traffic conditions in advance of control actions, the MS-FLC ensures that salient-influencing factors can be considered for proper control actions. With respect to computing efficiency, the partition of the control process into multi stages significantly reduces the number of rules and complexity of the model. Therefore, although divided into three stages, the computing speed is nearly as fast as that of ALINEA.
- The rule base in the MS-FLC governs the control settings with complete rule sets that correspond to complete courses of control action. This allows active response and smooth transition from one action to another.
- The rule base can be modified and upgraded at any time in the MS-FLC design phase. Rules in the rule base can be tuned using expert knowledge, traffic engineering knowledge and judgments of the control operators who confront daily to actual control operations.
- In traffic control for incident management, many types of data and information need to be gathered and analysed. The MS-FLC dissolves this

complicated problem by data-handling capability with simplified linguistic values instead of numerical values.

- Flexibility of the performance: the comparison with ALINEA algorithm identifies the following advantages of the MS-FLC control:
 - Unlike ALINEA whose control algorithm does not consider incident situation, MS-FLC specifically designed for incident management. Issues such as capacity reduction and queue management are addressed. However, MS-FLC can also be applied for recurring congestion management since the problem-solving strategy for the two types of congestion are similar in that both of them strike for the demand-capacity balance by proper handling of traffic demand.
 - Unlike ALINEA that targets a single control objective (optimal mainline occupancy), MS-FLC gives priority to the mainline and at the same time prevents excessive queue on the ramp.
 - Unlike ALINEA where the control algorithm does not consider flow upstream of the ramp, MS-FLC closely examines the relationship between traffic demand and road capacity using the V/C^* ratio, given which the ramp traffic is regulated accordingly.
 - Unlike ALINEA whose control algorithm is traffic-responsive that relies merely on current measurement, MS-FLC employs both reactive and proactive types of control. The former associates with critical conditions where urgent actions need to be carried out, while the latter governs the whole control process, especially during non-heavy congestion. With traffic condition anticipation, the MS-FLC control attempts to provide active response to mitigate the risk of overreaction.
- Data imprecision: A paramount advantage of the MS-FLC is its capability in capturing uncertainty and imprecision in data. The whole MS-FLC design process reflects an attempt to handle the problem of uncertainty, where data

is described by fuzzy values instead of purely numerical inputs. The parallel rule evaluation process increases the robustness of the model against data imprecision. The MS-FLC pre-processes data before calculating metering rate rather than calculating directly from raw data to prevent accumulation of errors. Therefore, as compared to analytical control methods, the fuzzy controller is less susceptible to bad data: in the presence of data inaccuracy, the solutions of the analytical approaches may become invalid, while in the MS-FLC many fuzzy sets are activated simultaneously to produce more reliable results.

Research findings

The following section briefly summarises important findings from the research.

In traffic flow and travel time forecasting:

The following findings are obtained from the traffic flow (Section 6.2.7) and travel time forecasting (Section 6.3.7). These findings, however, are stated for the described prediction model parameters, data sets and study segments on PIE. They have not been verified for other data sets and locations:

- (i) In normal traffic condition the SVM predictor in general outperforms the other predictors (see Figure 6.8, Figure 6.20, Figure 6.23, and Table 6.3).
- (ii) Under recurring congestion, SVM traffic volume prediction outperforms the baseline predictors (see Figure 6.10, Figure 6.11). However, under non-recurring congestion, SVM travel time prediction deteriorates severely if the data aggregation interval is large (see Figure 6.25).
- (iii) Through simulation, it is found that in non-recurring congestion, with higher data resolution and if similar patterns are available in the training set, the prediction accuracy of SVM can significantly be improved (see Figure 6.26). However this finding is stated with reservation since in actual situations there are difficulties in finding similar incident traffic patterns, as is presented in Section 6.5

- (iv) The higher data resolution (shorter interval), the better the prediction system reflects the actual traffic conditions and incident dynamics in the TSC model. Nevertheless, higher data resolutions impose hard technological challenges to data collection and transmission systems. Reasonably short data granularity is one that compromises these.
- (v) In SVM traffic prediction, the training size is not critical to the prediction accuracy, providing that the learning set has enough support vectors to construct hyperplanes. By contrast, the similarity of patterns between training and testing sets is of particular importance.
- (vi) The Nearest Neighbour could be a good method for SVM training. The method allows significant reduction of the size of data to shorten the training time while to maintain the prediction quality. However, the method requires additional pre-processing time to search for similar patterns, thus for online applications computational procedure should be established so as to incorporate this step into the overall prediction process in an efficient manner.
- (vii) For SVM travel time forecasting, the use of the NN value instead of the historical mean allows significant improvement in prediction accuracy (see Table 6.5).
- (viii) The rolling horizon approach, to some extent, has positive effects on SVM's prediction accuracy (see Figure 6.12). For short-term forecast, the longer the rolling horizon, the higher accuracy the prediction may achieve, due to the capability of SVM in solving complex classification problems in high dimensions.

As presented in Section 3.4.1, although the SVM technique seems less in line with the family of the fuzzy logic approach that is the skeleton of the KBS presented in this thesis, it exhibits an attempt to investigate a new and advanced technique in pattern recognition to improve the state-of-the-art traffic forecasting practices for incident applications.

In evaluation of MS-FLC:

The study of results from simulation scenarios (Section 8.4.3) provides the following significant findings:

- (i) The benefits of control intervention (*ALINEA* and *FLC*) depend on the magnitude of traffic demand and incident situation. Broadly speaking, more significant benefits can be achieved under high traffic demands and critical incident conditions than under favourable conditions, specifically:
 - Under "low-medium" traffic demand on the mainline and ramp, the control intervention does not show control effects. This implies that under "low-medium" traffic condition, ramp control intervention might not be necessary.
 - Under "medium-high" traffic demand the MS-FLC control contributes a significant enhancement in TTT saving, speed and density. However, the control methods have the same total vehicle mileage since the traffic states were similar across all control scenarios at the start and end of the evaluation period.
 - Under "high" traffic demand and severe capacity reduction, the control intervention allows more significant improvements of mainline travel conditions to be obtained. MS-FLC outperforms *ALINEA* with respect to global objectives. While the *ALINEA* algorithm gives control preferences to the mainline where TTT saving and mainline improvements are obtained at the expenses of the ramp traffic, the MS-FLC algorithm gains a better balance between the two. Again, it should be noted that this is stated with reservation since the experiment is based on a simplified network (single lane equivalent expressway, one-lane ramp).

The study of results from the uncertainty analysis (Section 8.4.4) provides further understanding on how the control performances change with the change in the input parameters. In the following findings, the "benefit" of the control intervention is

characterised by the relative improvement in the mainline speed, for the reasons presented in Section 8.4.4:

- (ii) The benefits of control intervention are highly sensitive to the length of the network, in particular to the length of the upstream section. In general, and the superiorities of the control methods over *No control* deteriorate as the network length increases.
- (iii) The superiorities of the control methods are less sensitive to the ramp storage capacity, in comparison to the network length. In general, the benefits of the control methods increase as the ramp storage capacity increases.
- (iv) The level of out-performance of the control algorithms is subjected to the temporal structure of the simulation: when the ratio of the incident and non-incident period is higher, the benefits of the control algorithms over “No control” increase.

It should be noted that the aforementioned findings are obtained from the model evaluation (Section 8.4) that was performed on a simplified network with an onramp, upstream and downstream incident segments, and a segment upstream of the ramp, given the local control stated in the research scope. Although the model properties were further explored through a uncertainty analysis with variations in the network length and simulation parameters, they are not verified for a more complicated network such as a corridor-wide control.

It should also be noted that there are no clear cuts between the terms *low*, *medium*, and *high* demands. They are loosely defined based on traffic demand in association with the reduced capacity. The question "to what range each of the demand categories covers" has not been verified numerically. More precise definitions can be derived from matching the flows with predefined percentile values of AADT. The inspection of daily traffic volume profiles reveals that *low-medium* demand level is usually associated with nighttime, while *medium-high* and *high* demands can mostly be observed in the daytime. Therefore, the MS-FLC has opportunities

for practical applications in most of the time domain (daytime) when the control intervention should be in operation.

9.2 LIMITATIONS

Despite important operational advantages discussed above, the fuzzy KBS has a number of limitations:

- The essence of the fuzzy KBS is the fuzzy rule base that formulates rules following fuzzy logic concept. In fuzzy logic, the input parameters are represented by fuzzy terms that are normally ill defined. In some cases, the partition of fuzzy sets must rely purely on personal judgements or commonsense reasoning without having reference data to justify these based on solid technical grounds. Therefore, while fuzzy terms are more generalized and easy to be perceived by the control operators, the partition of fuzzy sets is not unique and hence arguable.
- The fuzzy KBS only enhances its performance if the rule base is well formulated with appropriate membership function design and input-output mapping. Otherwise, the system performance deteriorates seriously. Therefore, it is important that effort be devoted from the very initial design step to establish an appropriate rule base.
- In calibrating parameters of membership functions of the fuzzy rule base, certain level of knowledge and expertise is required. The process of learning fuzzy rules requires a long time and the derivation of the membership functions could be tedious. However, to a great extent, the influence of the parameters is of local scale that facilitates manual tuning.
- In general, in the design of control system the stability analysis is one of the fundamental concerns. As a FLC, the MS-FLC is a highly non-linear system with complex stability behaviour. However, there exists no systematic methodology with respect to the stability analysis of the MS-FLC, to the best of the author's knowledge.

- The MS-FLC is complex and operationally expensive. It employs a considerable number of input parameters, thus extensive observations and measurements from the network are required.
- In the MS-FLC developed in this study, most of the rules in the third stage belong to state-action rules, where the control actions are issued based on the observation of the system state, and then the state is re-evaluated in the next control step and the cycle repeated. This system is known as *state evaluation fuzzy control*. In predictive fuzzy control (called Object Evaluation Fuzzy Control in Lee, 1990) the system predicts the future state given a hypothetical action, then control actions are issued so that the desired system state is obtained. A predictive control rule is described as:

$$\text{if } (y \text{ is } Y_i \rightarrow (x_1 \text{ is } X_1 \dots \text{and } x_j \text{ is } X_j)) \text{ then } y \text{ is } Y_i$$

The rule can be interpreted as: “if the (state) variables x_1 is $X_1 \dots$ and x_j is X_j when the (control) variable y is Y_i , then the control action Y_i should be selected”.

This type of rules is significant in that the control action is implemented conditionally to achieve a desired state. Nevertheless, the implementation of this type of rules is extremely complicated, and has not been feasible in the Fuzzy Inference System in MATLAB that is used in this research.

Limitation of the CFM

In this research, the CFM is developed following the modelling concepts from Gazis-Herman-Rothery (GHR) family of models (Gazis et al., 1963), referred to as general CFMs. In the models, the relationship between the leader and the following vehicles is a stimulus-response type of function (Equation 8.5), where the acceleration/deceleration of the following vehicle is proportional to its speed, the relative speed, and spacing between the leader and the following vehicles. Although following behaviour of individual cars is simulated, the parameters of interest from the model are the macroscopic outputs such as average flow, speed, and density.

The development of the CFM following the GHR concept has specific shortcomings. Specifically, the GHR models assume that the following vehicle reacts to small changes in the relative speed and to actions of its leader even though the distance may be very large. The response disappears only when the relative speed is zero (see the formulae of the response e_I in Figure 8.4). This may not be properly reflected in actual driver response. Psycho-physical models (in Brackstone and McDonald, 1998) address the issue by using thresholds: drivers react to changes in relative speed or spacing only when these thresholds are reached (Leutzbach, 1988).

Another shortcoming of the CFM is that, like most of car-following models, the CFM in this research uses a constant value of reaction time for all vehicles in each specific traffic condition such that drivers are typically more alert in congested traffic conditions and thereby have a shorter reaction time than in non-congested situations. This may be applicable to the models that are used to estimate macroscopic parameters (such as in this research), but is not very realistic from a micro perspective since reaction time varies from driver to driver. A possible way to improve this could be to assign reaction times to individual drivers according to a distribution of reaction times obtained from detailed field studies.

9.3 RECOMMENDATIONS FOR FUTURE RESEARCH

Despite effort that has been made in designing a high performance fuzzy KBS, this research has identified potential avenues that can be extended from this research, as summarised in the followings:

(i) More robust techniques for designing membership functions

Even though several advanced learning techniques have been investigated, including statistical fuzzy partitioning, fuzzy-C-Mean, and ANFIS, potential techniques in learning membership functions parameters such as Genetic Algorithm should be explored. The algorithm is an evolutionary approach in natural selection and may provide a promising direction for fine-tuning of membership functions as well as fuzzy adaptive systems, which are the bottleneck in fuzzy logic theory.

(ii) Research on adaptive MS-FLC

The rule base of the current MS-FLC is static in nature, where rules are designed off-line. In circumstances such as changes in traffic environment or the shifting from daytime to nighttime, etc., some rules should be adaptable. Furthermore, the existence of a number of site-specific parameters in the MS-FLC would inevitably reduce the transferability of the model. Therefore, a MS-FLC with an adaptation component where parameters can be calibrated and rules can be modified on-line is worth exploring and should be the subject of future research.

(iii) Development of the GUI interface

The current MS-FLC is designed as an automated system. In actual control operation, a seamless and iterative man-machine two-way communication is required. For the system to undertake full requirements of DSS's functions, further study may involve the design of a GUI platform that offers dialogs and graphical displays to assist the operators in acquiring information, selecting and issuing control directives.

(iv) Research on coordinated and integrated systems

As specified previously, severe incident problems may require area-wide control, which is coordinated or integrated system. The architecture for functional integration between the KS-DSS components for corridor traffic control has been proposed in Section 8.5.1. Further study may develop a rule base of the CW-FLC coordinator that monitors the operation of the whole FLC control group. In addition, research on ATMS-ATIS system integration following fuzzy logic approach would be significant. The essence of this is the integration between ramp control and route control. To this end, knowledge on fuzzy route choice behaviour and on bi-level integration problems is required.

REFERENCES

- Aarts, L., and Schagen, I. V. (2006). "Driving speed and the risk of road crashes: A review". Accident Analysis & Prevention, Vol 38 (2).
- Ahmed, K. I. (1999). "Modelling drivers' acceleration and lane changing behaviour". Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA.
- Al-Kaisy, A., Jung, Y., and Rakha, H. (2005). "Developing Passenger Car Equivalency Factors for Heavy Vehicles during Congestion". Journal of Transportation Engineering – ASCE, 131(7), pp. 514-523.
- Al-Deek, H., D'Angelo, M. and Wang, M. (1998). "Travel time prediction with non-linear time series". Proceeding of the ASCE 5th International Conference on Applications of Advanced Technologies in Transportation, Newport Beach, Calif., pp. 317-324.
- Arslan, T. C. and Khisty, C. J. (2005). "A rational reasoning method from fuzzy perceptions in route choice". Fuzzy Sets and Systems 150, pp. 419-435.
- Brackstone, M. and McDonald, M. (1998). "Car-following: a historical review." Transportation Research F 2, pp. 181–196.
- Ben-Akiva, M., Cuneo, D., Hasan, M., Jha, M. and Yang, Q. (2003). "Evaluation of freeway control using a microscopic simulation laboratory". Transportation Research Part C 11, pp. 29-50.
- Berkan, R. C. and Trubatch, S. L. (1997). Fuzzy System design Principles. Building Fuzzy IF-THEN rule bases. IEEE Press, New York.
- Bidgoli, H. (1989). Decision Support Systems – Principles and practice. West Publishing Company, United States of America.
- Blanzieri, E., and Bryl, A. (2007). "Highest probability SVM nearest neighbour classifier for spam filtering". Technical report #DIT-07-007. University of Trento, Department of Information and Communication Technology.
- Bogenberger, K. and Keller, H. (2001). "An Evolutionary Fuzzy System for Coordinated and Traffic Responsive Ramp Metering". Proceedings of the 34th Hawaii International Conference on System Sciences.
- Chan, Y., Regan, E. J. and Pan, W. M. (1986). "Inferring an origin-destination matrix directly from network flow sampling". Transportation Planning and Technology 11, pp. 27-46.
- Chang, C. C., and Lin, C. J. (2001). "LIBSVM: a library for support vector machines". Available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. Access date: Feb. 15th, 2005.
- Chang, J. L., and Miaou, S. P. (1999). "Real-time prediction of traffic flows using dynamic generalized linear models". Transportation Research Record 1678, pp. 168–178.
- Chen, H., and Grant-Muller, S. (2001). "Use of sequential learning for short-term traffic flow forecasting". Transportation Research Part C 9, pp. 319-336.

- Chen, H. P. and Parng, T. M. (1996). "A new approach of multi-stage fuzzy logic inference". Fuzzy Sets and Systems 78, pp. 51-72.
- Chen, J. E. and Kevin K. N. (1995). "Constructing membership functions using interpolation and measurement theory". Fuzzy Sets and Systems 73, pp. 313-327.
- Chen, L. C., May A. D. and Auslander, D. M. (1990). "Freeway Ramp Control using Fuzzy Set Theory for Inexact Reasoning". Transportation Research part A 24, No. 1, pp. 15-25.
- Chen, M. and Chien, S. (2001). "Dynamic Freeway Travel Time Prediction Using Probe Vehicle Data: Link-based vs. Path-based". TRB, 80th Annual Meeting, Washington, D.C.
- Chen, M. S. and Wang, S. W. (1999). "Fuzzy clustering analysis for optimizing fuzzy membership functions". Fuzzy Sets and Systems 103, pp. 239-254
- Chien, S. I. J. and Kuchipudi, C. M. (2003). "Dynamic travel time prediction with real-time and historic data". Journal of Transportation Engineering – ASCE, 129(6), pp. 608-616.
- Chu, L., Oh, J. S. and Recker, W. (2005). "Adaptive Kalman Filter Based Freeway Travel time Estimation". 84th TRB Annual Meeting, Washington, D.C.
- Chun-Hsin, W., Chia-Chen, W., Da-Chun, S., Ming-Hua, C. and Jan-Ming, H. (2003). "Travel time prediction with support vector regression". Proceedings of the 2003 IEEE Conference on Intelligent Transportation Systems, Shanghai, China.
- Chung, F. L. and Duan, J. C. (2000). "On multistage fuzzy neural network modelling". IEEE Trans. Fuzzy Systems 8 (2), pp. 125-142.
- Cragg, C. A. and Demetsky, M. J. (1995). "Simulation analysis of route diversion strategies for freeway incident management". Final Report, Virginia Transportation Research Council, Virginia.
- Cristianini, N. (2000). An Introduction to Support Vector Machines. Cambridge University Press.
- Cuena, J., Hernandez, J. and Molina, M. (1995). "Real-time traffic management through knowledge-based models: the TRYS approach". Available at: www.erudit.de/erudit/events/tc-c/05_Trys.pdf, Access date: October 15th, 2002.
- Davis, G. A. and Nihan, N. L. (1991). "Nonparametric regression and short-term freeway traffic forecasting". ASCE Journal of Transportation Engineering 117 (2), pp. 178-188.
- Dia, H. (1996). "Artificial neural network for automated freeway incident detection". PhD Thesis, Monash University.
- Dia, H. (2001). "An object-oriented neural network approach to short-term traffic forecasting". European Journal of Operational Research 131, pp. 253-261.
- Diakaki, C., Papageoriou, M. and McLean, T. (1997). "Simulation studies of integrated corridors control in Glasgow". Transportation Research C 5, pp. 211-224.
- Dubois, D. and Prade, H. (2000). Fundamental of fuzzy sets. Kluwer Academic Publishers, USA.

- Fan, H. S. L. (1990). "Passenger car equivalents for vehicles on Singapore expressways". Transportation Research A, Vol. 24, No. 3, pp. 391-396.
- Fan, H. S. L., Olszewski, P., Lam, S. H., Lum, K. M. and Tan, Y. W. (1995). "Speed-flow relationships for expressways and arterial roads". Research report NTU/CTS/95-1, Centre for Transportation Studies, Nanyang Technological University, Singapore.
- Fu, L. and Rilett, L. R. (2000). "Estimation of Time-Dependent, Stochastic Route Travel Times using Artificial Neural Networks". Transportation Planning and Technology, Vol. 24, No. 1, pp. 25-36.
- Furey, T. S., Cristianini, N. D., Bednarski, D. W., Schummer, M. and Haussler, D. (2000). Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data. Bioinformatics, 16(10), pp. 906-914, Oxford University Press.
- Gazis, D. C., R. Herman, and R. W. Rothery (1963). "Analytical Methods in Transportation: Mathematical Car-Following Theory of Traffic Flow". Journal of the Engineering Mechanics Division, ASCE. Paper 372, pp. 29-46.
- Gunn, S. R. (1998). "Support vector machine for classification and regression". Technical Report, University of Southampton, Southampton, U.K.
- Gupta, A., Maslanka, V. J. and Spring, G. S. (1992). "Development of prototype knowledge-based expert system for managing congestion on Massachusetts Turnpike". TRB 71st Annual Meeting, Transportation Research Record 1358, pp. 60-66.
- Henn, V. (2000). "Fuzzy route choice model for traffic assignment". Fuzzy Sets and systems 116, pp. 77-101.
- Hong, T. P. and Chen, J. B. (1999). "Finding relevant attributes and membership functions". Fuzzy Sets and Systems 103, pp. 389-404
- Hong, T. P. and Lee, C. Y. (1996). "Induction of fuzzy rules and membership functions from training examples". Fuzzy Sets and Systems 84, pp. 33-47
- Hoogendoorn S. P., Coppinga, G. and Kaymak, U. (1998). "Perspectives of Fuzzy Logic in Traffic Engineering". Research Report, TRAIL Research School, Delft, report on behalf of Dutch Ministry of Transport.
- Ishak, S. and Al-Deek, H. (2002). "Performance Evaluation of Short-Term Time-Series Traffic Prediction Model". Journal of Transportation Engineering, Vol. 128, No. 6.
- Jian, W. (2001). "Development of fuzzy logic controller design techniques". Master thesis. School of EEE. The Nanyang Technological University, Singapore.
- Jiangtao, R., Xiaoling, O., Yi, Z., Jingyan, S. and Dongcheng, H. (2002). "SVM based detection of traffic incident". The 9th World Congress on Intelligent Transport Systems, Chicago, USA.
- Jou, R. C., Lam, S. H., Liu, Y. H. and Chen, K. H. (2005). "Route switching behavior on freeways with the provision of different types of real-time traffic information". Transportation Research Part A 39 pp. 445-461.

- Kachroo, P. and Özbay, K. (1999). Feedback Control Theory for Dynamic Traffic Assignment. Advances in Industrial Control, Springer.
- Kachroo, P., Özbay, K., Kang, S. and Burns, J. A. (1998). "System Dynamics and Feedback Control Problem Formulations for Real Time Dynamic Traffic Routing". Mathl. Comput. Modelling, Vol. 27, No. 9-11, pp. 27-49.
- Kecman, V. (2001). Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models. The MIT press, Cambridge, Massachusetts, London, England.
- Kim, K. J. (2003). "Financial time series forecasting using support vector machines". Neurocomputing 55, pp. 307-319.
- King P. J. and Mamdani, E. H. (1975). "The application of fuzzy control systems to industrial processes". IFAC World Congress, MIT, Boston.
- Kirby, H. R., Watson, S. M. and Dougherty, M. S. (1997). "Should we use neural networks or statistical models for short-term motorway traffic forecasting?". International Journal of Forecasting 13, pp. 43-50.
- Klir, G. J. and Folger T. A. (1988). Fuzzy sets, uncertainty, and information. Englewood Cliffs, NJ, Prentice Hall.
- Kosko, B. (1992). Neural Networks and Fuzzy Systems: A Dynamical System Approach to Machine Intelligence. Prentice-Hall, Englewood Cliffs, N. J.
- Kwon, J., Coifman, B. and Bickel, P. (2000). "Day-to-Day Travel Time Trends and Travel Time Prediction from Loop Detector Data". TRB 80th Annual Meeting, Washington D.C.
- Larkin, L. (1985). A fuzzy logic controller for aircraft fight control. In: Sugeno, M. (Ed.). Industrial applications of fuzzy control. Elsevier Science (North-Holland), New York, pp. 87-103.
- Larsen, P. M. (1980). "Industrial applications of fuzzy logic control" International Journal of Man-Machine Studies, Vol. 12, No. 1, pp. 3-10.
- Lee, C. C. (1990). "Fuzzy logic in control systems: Fuzzy logic controller - Part I". IEEE Transactions on Systems, Man, and Cybernetics. Vol. 20, No. 2.
- Leutzbach, W. (1988). Introduction to the theory of traffic flow. Berlin, Springer Verlag.
- Li, R., Wang, H. N., He, H., Cui, Y. M., and Du, Z. L. (2007). "Support Vector Machine combined with K-Nearest Neighbours for Solar Flare Forecasting". Chinese Journal of Astronomy and Astrophysics, Vol. 7(3), pp. 441-447.
- Lin, C. K. and Chang, G. L. (1998). "Development of a Fuzzy-Expert System for Incident Detection and Classification". Mathl. Comput. Modelling Vol. 27, No. 9-11, pp. 9-25.
- Lindley, J. A. (1987). "Urban freeway congestion: Quantification of the problem and effectiveness of potential solutions". ITE Journal, pp. 27-32.

- Lint, V., Hoogendoorn, S. P. and Zuylen, H. J. (2002). "Freeway Travel Time Prediction with State-Space Neural Networks". TRB 81st Annual Meeting, Washington D.C.
- Lint, V., Hoogendoorn, S. P. and Zuylen, H. J. (2005). "Monitoring and predicting freeway travel time reliability. Using width and skew of the day-to-day travel time distribution". TRB Annual Meeting, Washington D.C.
- Logi, F. and Ritchie, S. G. (2001). "Development and evaluation of a knowledge based systems for traffic congestion management and control". Transportation Research part C 9, pp. 433-459.
- Mak, C. L. (2002). "New dual-variable algorithms for detecting lane-blocking incidents on expressways". PhD Thesis, Nanyang Technological University, Singapore.
- Makridakis, S., Wheelwright, S. C. and McGee V. E. (1983). Forecasting: Methods and Applications. Wiley, New York.
- Mamdani, E. (1974). "Application of fuzzy algorithms for control of simple dynamic plant". Proceedings IEE 121, pp. 1585-1588.
- MATLAB (2002). User Manual. Version 6.5, Release 13.
- May, A. D. (1990). Traffic flow fundamentals. University of California, Berkeley. Prentice hall, New Jersey, USA.
- Mendel, J. (1995). "Fuzzy logic systems for engineering: a tutorial". Proceedings of the IEEE 83, pp. 345-377.
- Minh, N. N. (2004). "Two-stage support vector machines for protein structure and solvent prediction". PhD. Thesis, School of Computer Engineering, Nanyang Technological University, Singapore.
- Montgomery, D. C. and Runger, G. C. (2002). Applied Statistics and Probability for Engineers. Third Edition. John Wiley & Sons, Inc., USA.
- Moreno-Banos, J. C., Papageoriou, M. and Schaffner, C. (1993). "Integrated optimal flow control in traffic networks". European Journal of Operations Research 71, pp. 317-323.
- MOT, 2002. "Facts and Statistics". <http://www.mot.gov.sg>. Accessed 14 Sep 2002.
- Nakatsuyama, M., Nagahashi, N. and Nishizuka, N. (1983). "Fuzzy logic phase controller for traffic functions in the one way arterial road". Proceedings IFAC 9th Triennial World Congress. Pergamon Press, Oxford, pp. 2865-2870.
- Nam, D. H. and Drew, D. R. (1996). "Traffic dynamics: Method for estimating freeway travel times in real time from flow measurements". Journal of Transportation Engineering, 122(3), pp. 186-191.
- Nayaka, P. C, Sudheer, K. P., Rangan, D. M. and Ramasastrid, K. S. (2004). "A neuro-fuzzy computing technique for modelling hydrological time series". Journal of Hydrology 291, pp. 52-66.
- Ngo, C. Y. and Victor O. K. L. (1994). "Freeway traffic control using fuzzy logic controllers". Information sciences 1, pp. 59-76.

- Okutani, I. and Stephanedes, Y. J. (1984). "Dynamic prediction of traffic volume through Kalman filtering theory". Transportation Research part B 18 (1), pp. 1-11.
- Olaru, D. and Smith, B. (2005). "Modelling behavioural rules for daily activity scheduling using fuzzy logic". Transportation 3, pp. 423-441.
- Ostergard, J. J. (1976). "Fuzzy logic control of a heat exchanger process". Internal Report 7601, Electric Power Engineering Department, Technical University of Denmark, Lyngby.
- Oswald, R. K., Scherer, W. T. and Smith, B. L. (2001). "Traffic Flow Forecasting Using Approximate Nearest Neighbour Nonparametric Regression". Research Project Report. Report No. UVACTS-15-13-7, University of Virginia.
- Ozbay, K. and Kachroo, P. (1999). Incident Management in Intelligent Transportation Systems. Artech House Boston, London.
- Papageoriou, M. (1994). "An integrated control approach for traffic corridors". Transportation Research C 3, pp. 19-30.
- Papageorgiou, M. (1999). Handbook of Transportation Science. Chapter 8: Traffic Control, pp. 233-267. Kluwer Academic Publishers, Boston.
- Papageorgiou, M., Hadj Salem, H. and Blosseville, J. M. (1991). "ALINEA: A Local Feedback Control Law for On-Ramp Metering". Transportation Research Record 1320, pp. 58-64.
- Papageorgiou, M., Hadj-Salem, H. and Middelham, F. (1997). "ALINEA Local Ramp Metering. Summary of Field Results". TRB 76th Annual Meeting, Washington, D.C.
- Pappis C. D. and Mamdani E. H. (1977). "A fuzzy logic controller for a traffic junction". IEEE Transactions on Systems, Man and Cybernetics SMC-7, pp. 707-717.
- Peeta, S. and Mahmassani, H. S. (1995). "Multiple user classes real-time traffic assignment for online operations: A rolling horizon solution framework". Transportation Research part C 3(2), pp. 83-98.
- Quadstone (2004). Paramics V5.1: Modeller Reference Manual. Quadstone Limited, Version No. 1.0. Quadstone Limited, Scotland.
- Raju, G. V. S. and Zhou, J. (1993). "Adaptive hierarchical fuzzy controller". IEEE Trans. Systems Man Cybernet. 23, pp. 973-980.
- Ran, B. and Boyce, D. (1996). Modelling dynamic transportation networks. Heidelberg, Springer.
- Rice, J. and Zwet, E. V. (2004). "A simple and effective method for predicting travel times on freeways". IEEE Intelligent Transportation Systems conf. Proc., pp. 227-232.
- Rilett, L. and Park, D. (1999). "Direct forecasting of freeway corridor travel times using spectral basis neural networks". TRB 78th Annual Meeting (CD-ROM), Transportation Research Record 1617, Washington, D.C., pp. 163-170.

- Ritchie, S. (1990). "A knowledge-based decision support architecture for advanced traffic management". Transportation Research Part A, 24(1), pp. 27-37.
- Roden, D. (1996). "Forecasting Travel Time". Transportation Research Record 1518, National Research Council, Washington D.C., pp.7-12.
- Ross, T. J. (1995). Fuzzy Logic with Engineering Applications. McGraw-Hill, New York.
- Sadek, A. V. (1998). "Case-Based Reasoning for Real-Time Traffic Flow Management". Ph.D. Thesis, University of Virginia.
- Sakai, Y. (1985). "A fuzzy controller in turning process automation". Industrial Applications of Fuzzy Control. M. Sugeno, Ed. Amsterdam: North-Holland, pp. 139-152.
- Sasaki, T. and Akiyama, T. (1987). "Fuzzy on-ramp control model on urban expressway and its extension". Transportation and Traffic Theory. Elsevier Science, New York, pp. 377-395.
- Sasaki, T. and Akiyama, T. (1988). "Traffic control process of expressway by fuzzy logic". Fuzzy Sets and Systems 26, pp. 165-178.
- Scholkopf, A. and Smola, A. J. (2002). Learning with Kernels: Support vector machines, Regularization, Optimization, and beyond. The MIT Press, Cambridge, Massachusetts, London, England.
- Schrader, C. C., Kornhauser, A. L. and Friese, L. M. (2004). "Using historical information in forecasting travel times". TRB 83rd Annual Meeting, CD-ROM, National Academies Press, Washington D.C.
- Sheu, J. B. (2002). "A fuzzy clustering-based approach to automatic freeway incident detection and characterization". Fuzzy Sets and Systems 128, pp. 377-388.
- Sintas, A. F, Cadenas, J. M. and Fernando Martin, F. (1999). "Membership functions in the fuzzy C-Means algorithm". Fuzzy Sets and Systems 101, pp. 49-58.
- Smaragdis, E. and Papageorgiou, M. (2003). "A Series of New Local Ramp Metering Strategies". TRB 82nd Annual Meeting, Washington, D.C.
- Smith, B. L., Demetsky, M. J. (1994). "Short-term traffic flow prediction: Neural Network approach". Transportation Research Record 1453, pp. 98-104.
- Smith, B. L. and Demetsky, M. J. (1997). "Traffic flow forecasting: comparison of modelling approaches". Journal of Transportation Engineering. 123(4), pp. 261-266.
- Smith, B. L., Williams, B. M. and Oswald, R. K. (2000). "Parametric and nonparametric traffic volume forecasting". TRB 79th Annual Meeting, Washington, D.C.
- Smola, A. J. and Scholkopf, B. (1998). "A tutorial on Support vector Regression". NeuroCOLT2 Technical Report Series, NC2-TR-1998-030.
- Stephanedes, Y. and Chang, K. K. (1993). "Optimal control of freeway corridors". Journal of Transportation Engineering 119, ASCE, pp. 504-514.

- Stern, E. (1999). "Reactions to congestion under time pressure". Transportation Research 7, pp. 75-90.
- Sugeno, M. and Nishida, M. (1985). "Fuzzy control of model car". Fuzzy Sets and Systems 16, pp. 103-113.
- Sun, J., and Cho, S. (2006). "Pattern Selection for Support Vector Regression based on Sparseness and Variability", International Joint Conference on Neural Networks, Vancouver, BC, Canada.
- Takagi, T. and Sugeno, M. (1983). "Derivation of fuzzy control rules from human operator's control actions". Proceeding of the IFAC Symposium, in Fuzzy Information, Knowledge Representation and Decision Analysis, pp. 55-60.
- Taylor, C. and Meldrum, D. (2000). "Evaluation of a fuzzy logic ramp metering algorithm: a comparative study between three ramp metering algorithms used in the Greater Seattle area". Washington State Department of Transportation, USA. Available:
http://www.wsdot.wa.gov/regions/northwest/traffic/tsmc/RampMeters/eval_report.pdf, Access date: Apr. 24th, 2005.
- Taylor, M. A. P, Bonsall, P. W. and Young, W. (2000). Understanding Traffic Systems: Data, Analysis and Presentation. Ashgate Publishing Ltd, UK.
- Teodorović, D. (1999). "Fuzzy logic systems for transportation engineering: the state of the art". Transportation Research part A, 33, pp. 337-364.
- Teodorović, D. and Kalić, M. (1996). "Solving the modal split problem by fuzzy rules generated by learning from examples". Proceedings of Information Technologies, Zabljak, Yugoslavia, pp. 48-54.
- Teodorović, D., Kalić, M. and Pavković, G. (1994). "The potential for using fuzzy set theory in airline network design". Transportation Research part B 28, pp. 103-121.
- Teodorović, D. and Vukadinović, K. (1998). Traffic control and transport planning: A Fuzzy Sets and Neural Networks Approach. Kluwer Academic Publishers. Boston/Dordrecht/London.
- Timm, H., Borgelt, C., Döring, C. and Kruse., R. (2004). "An extension to possibilistic fuzzy cluster analysis". Fuzzy Sets and Systems 147, pp. 3-16.
- Toan, T. D. and Lam, S. H. (2005). "Development of a Rule-Based System for Congestion Management". TRB 84th Annual Meeting, Washington, D.C.
- Trabia, M. B., Kaseko, M. S. and Ande, M. (1999). "A two-stage fuzzy logic controller for traffic signals". Transportation Research part C 7 (6), pp. 353-367.
- TRB (1997). Highway Capacity Manual. Transportation Research Board, National Research Council. Special Report 209.
- Turban, E. (1992). Expert system and applied Artificial Intelligence. Marmillan Publishing Company, New York.
- Van Arem. B., Kirby, H. R., Van Der Vlist, M. J. M. and Whittaker, J. C. (1997). "Recent advances in the field of short-term traffic forecasting". International Journal of Forecasting 13, pp. 1-12.

- Vanajakshi, L. (2004). "Travel Time Prediction Using Support Vector Machine Technique". Institute of Transportation Engineers 2004 Daniel B. Fambro Student Paper Award.
- Vanajakshi, L. and Rilett, L. R. (2004). "A comparison of the performance of artificial neural network and support vector machines for the prediction of traffic speed". IEEE Intelligent Vehicles Symposium, University of Parma, Italy.
- Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer Verlag, New York.
- Vert, J. P. (2001). "Introduction to support vector machines (SVM) and application in bioinformatics". Seminar, Bioinformatics Centre, Kyoto University.
- Wang, C. F. (1972). "On a ramp-flow assignment problem". Transportation Science 6, pp. 114-130.
- Wang, J. J. and May, A. D. (1973). "Computer model for optimal freeway on-ramp control". Highway Research Record 469, pp. 16-25.
- Wang, W., and Xu Z. (2004). "A heuristic training for support vector regression". Neurocomputing 61, pp. 259 – 275.
- Wattleworth, J. A. (1967). "Peak Period Analysis and Control of a Freeway System". Highway Research Record 157, pp. 1-21.
- Whittaker, J., Garside, S. and Lindveld, K. (1997). "Tracking and predicting a network traffic process". International Journal of Forecasting 13, pp. 51-61.
- Williams, B. M. (2001). "Multivariate vehicular traffic flow prediction: an evaluation of ARIMAX modelling". TRB 80th Annual Meeting, Washington, D.C.
- Williams, B. M., Durvasula, P. K. and Brown, D. E. (1997). "Urban freeway travel prediction: application of seasonal ARIMA and Exponential Smoothing Models". TRB 77th Annual Meeting, D.C.
- Yuan, F. and Cheu, R. L. (2003). "Incident detection using support vector machines". Transportation Research Part C 11, pp. 309-328.
- Xu, H., Kwan, C. M., Haynes, L. and Pryor, J. D. (1998). "Real-time adaptive on-line traffic incident detection". Fuzzy Sets and Systems 93, pp. 173-183.
- Xu, W. and Chan, Y. (1993). "Estimating an origin-destination matrix with fuzzy weights. Part 1: Methodology". Transportation Planning and Technology 17, pp. 127-144.
- Yager, R. R and Filev, D. P. (1994). "Essentials of Fuzzy Modelling and Control". A Wiley-Interscience Publication, New York.
- Yasunobu, S., Miyamoto, S. and Ihara, H. (1983). "Fuzzy control for automatic train operation system". Proceeding of 4th IFAC/IFIP/IFORS Int. Congress on Control in Transportation Systems, Baden-Baden.
- Yasunobu, S. and Miyamoto, S. (1985). "Automatic train operation by predictive fuzzy control," in Industrial Application of Fuzzy Control, M. Sugeno, Ed. Amsterdam: North-Holland, pp. 1-18.

Yeh, Z. M. and Li, K. H. (2004). "A systematic approach for designing multistage fuzzy control systems". Fuzzy Sets and Systems 143, pp. 251-273.

Zadeh, L., A. (1994). "Fuzzy logic, neural networks, and soft computing". Journal of Communication of the ACM. Vol. 37, No. 3.

Zhang, H., Berg, A. C., Maire, M., and Malik, J. (2006). "SVM-kNN: Discriminative Nearest Neighbour Classification for Visual Category Recognition".

Source: http://www.cs.berkeley.edu/~aberg/papers/zbmm_cvpr06_knnsvm.pdf

Accessed date: Nov 22nd, 2007.

Zhang, H., Ritchie, S. and Recker, W. (1996). "Some general results on the optimal ramp metering control problem". Transportation Research C 4, pp. 51-69.

Zhang, X. Y. and Rice, J. A. (2003). "Short-term travel time prediction". Transportation Research Part C, 11(3-4), pp. 187-210.

APPENDIX A: HDB DATABASE

Overview

The HDB stores traffic data, incident information and network attributes. The HDB has data collected since October 2002 and hosts millions of records. Given the huge amount of data with the substantial proportion of redundant data, the HDB has been experienced a process known as normalization to reduce the repetition and redundancies of data in a database. In this part, the normalization and cleaning of data is presented in a practical way other than the three-normal forms in a standard normalization process, considering the structure and the attributes of data provided in the LTA's database, and the need of the end user.

In normalisation and cleaning of the data in the HDB, specific tasks have been carried out, including the removal of redundant data and the transformation of incident data to a manageable format.

1. Removing redundant data

HDB: Traffic

Traffic data are retrieved from the LTA database using My SQL and are stored in relational tables in MS Access format, where rows uniquely indicate data records and columns indicate traffic attributes. The attributes of traffic data include *RoadID*, being the location of the traffic detector, *Speed*, *Volume* (which has been actually converted to hourly flow, i.e. flow rate), *Date*, and *Time*:

$$Traffic_Attributes = \{RoadID, Speed, Volume, Date, Time\}$$

For traffic management purpose, expressways are divided into segments of lengths ranging from 0.9-1.7 km, each covered by several video-based stations, uniquely identified by its *RoadID*. Since the data are collected for both expressways and urban road network, while only data on expressways and ramps are of interest in this study. Therefore, the data are filtered from the retrieved data using Structural Query Language (SQL) statements:

Select {set of selected attributes} from (databases' tables) where RoadID = {set of selected road IDs};

Through the filtering step, a substantial portion of unrelated data was excluded from the retrieved data. In addition, data are sometimes contaminated due to the system's errors. In MS Access form, rows (records) with abnormal data can be easily sorted out from the retrieved data. Those data include:

- Rows with negative traffic volume or speed;
- Rows with extraordinary high speed and volume;
- Rows of identical records.

Having been cleaned, the HDB data are stored in the MS Access database (global database), and in separate files in Excel format where each file stores data in one day, and grouped into folders for subsequent data manipulations and analyses.

HDB: Incident

The connection and processing of incident data involve two steps:

- *Connection and archiving*
- *Transformation of incident data format*

A program is coded to connect to the LTA database using a client-server architecture, and retrieve data into the Centre for Transportation Studies (CTS) workstation in folders. The folders (named by timestamps at which data is collected) are obtained at 5-minute interval; therefore, data in one day includes 288 folders (Figure A1).

Figure A1: Archiving of HDB incident database

Each folder in Figure A1 consists of 10 files (Figure A2).

Figure A2: Content of one folder

A file typically consists of several incidents in the expressways and urban arterials. The incident data is continually updated in subsequent files, and is described in text format (Figure A3). The information on each incident consists of:

- *IncidentID* that help to identify each incident, classified into incident on expressway network detected by the EMAS system (labelled as EM, followed by the incident ID), and incidents detected by the Green Link system (labelled as GL, followed by the incident ID);
- *LinkID*, being the location where the incident takes place. Unlike the traffic data, the linked is a sub-segment that covers a portion of RoadID (of prevailing length of 100-150m, while in particular there are sub-segments of 50-60m long).
- *Start Date* and *Start time*, being the date and the time the incident happen;
- *Incident type* reflects the cause of incidents (accident, vehicle breakdown, obstacles, roadwork, heavy traffic). The incident types reported in the LTA incident database include planned incident (roadwork) and congestion information (labelled as *Heavy traffic*) without stating whether the congestion is recurring or non-recurring. To investigate if “Heavy traffic” is due to incident congestion or not, the traffic data is linked with incident data by mapping two database Tables (Figure 4.7). If the congestion is recurring, the reported “incident” will be removed from the incident database since it is not genuine incident.
- *End time*, being the reported end time of the incident.

Figure A3: Content of an incident file

2. Transformation of incident data

Since the incident data is reported in the text format, it does not permit direct manipulation and linking with traffic data, and does not facilitate data analysis. In addition, the LTA's reported incident information continually update real-time incident information on the network in subsequent intervals without the *End time* until when the incident is over, while for offline data analysis the HDB incident database only need to store records with full information of incident attributes. Therefore, to facilitate data manipulation and to substantially cut down redundant data, a program (shown later in Appendix A) is coded to transform the data in text format into tabular format that can be converted easily into Excel or MS Access, and at the same time combine the contents from hundreds of thousands of files and folders (in the previous figures) into a single file (Figure A4). The length of the aggregated file depends on the user. In Figure A4, each incident is represented in one row, and each row consists of columns associating with incident attributes, such as incident ID, link ID, date, incident start and end time, incident type, etc.

Incident_ID	Link_ID	Start_Date	Start_Time	Incident_Type	End_Date	End_Time
EM146003	10106243	28/8	22:28	V	28/8	22:44
EM145999	10097698	28/8	21:36	H	28/8	22:49
EM146001	10095583	28/8	21:45	V	28/8	21:54
EM146000	10095643	28/8	21:48	H	28/8	22:40
EM146003	10106243	28/8	22:28	V	28/8	22:44
EM145999	10097698	28/8	21:36	H	28/8	22:49
EM146001	10095583	28/8	21:45	V	28/8	21:54
EM146000	10095643	28/8	21:48	H	28/8	22:40
EM146003	10106243	28/8	22:28	V	28/8	22:44
EM145999	10097698	28/8	21:36	H	28/8	22:49
EM146001	10095583	28/8	21:45	V	28/8	21:54
EM146000	10095643	28/8	21:48	H	28/8	22:40
EM146003	10106243	28/8	22:28	V	28/8	22:44
EM145999	10097698	28/8	21:36	H	28/8	22:49
EM146001	10095583	28/8	21:45	V	28/8	21:54
EM146000	10095643	28/8	21:48	H	28/8	22:40
EM146003	10106243	28/8	22:28	V	28/8	22:44
EM145999	10097698	28/8	21:36	H	28/8	22:49
EM146001	10095583	28/8	21:45	V	28/8	21:54
EM146000	10095643	28/8	21:48	H	28/8	22:40
EM146003	10106243	28/8	22:28	V	28/8	22:44
EM145999	10097698	28/8	21:36	H	28/8	22:49
EM146001	10095583	28/8	21:45	V	28/8	21:54
EM146000	10095643	28/8	21:48	H	28/8	22:40
EM146003	10106243	28/8	22:28	V	28/8	22:44
EM145999	10097698	28/8	21:36	H	28/8	22:49
EM146000	10095643	28/8	21:48	H	28/8	22:40

Figure A4: Transformation and aggregation of incident data

(Columns from left to right: Incident_ID, Link_ID, Start_Date, Start_Time, Incident_Type, End_Date, End_Time)

```

import java.io.*;

class AppendData {
    // data of an incident
    private String incidentID;
    private String linkID;
    private String incidentStartDate;
    private String incidentStartTime;
    private String incidentEndDate;
    private String incidentEndTime;
    private String typeOfIncident;

    // data of a previous incident
    private String pIncidentID;
    private String pLinkID;
    private String pIncidentStartDate;
    private String pIncidentStartTime;
    private String pIncidentEndDate;
    private String pIncidentEndTime;
    private String pTypeOfIncident;

    // output to result file
    private FileOutputStream writeStream = null;

    AppendData (){
        pIncidentID = "";
        pLinkID = "";
        pIncidentStartDate = "";
        pIncidentStartTime = "";
        pIncidentEndDate = "";
        pIncidentEndTime = "";
        pTypeOfIncident = "";
    }

    // private variables of incident
    /*
    * This function read data from config.txt to verify that
    * which folder will be read.
    * Output: The folder's string
    */
    private String readConfigInfo(){
        String inputFileName = "config.txt";
        String result = null;
        try {
            File inputFile = new File(inputFileName);
            FileInputStream in = new FileInputStream(inputFile);
            byte bt[] = new byte[(int)inputFile.length()];
            in.read(bt);
            result = new String(bt);
            in.close();
        } catch (java.io.IOException e) {
            System.out.println("Cannot read from config.txt");
        }
        return result.trim();
    }
}

/*

```

```

* This method combines contents of all files under a folder
*/
public void combine()
{
    // get path name in config.txt
    String pathName = readConfigInfo();
    try {
        // get the folder which is referred in config.txt
        File inputFolder = new File(pathName);
        String newFileName = pathName + File.separatorChar + inputFolder.getName() +
".txt";
        // create new file in this folder
        //File newFile = File.createTempFile(inputFolder.getName(), ".txt", inputFolder);
        File newFile = new File(newFileName);
        writeStream = new FileOutputStream(newFile);

        if(inputFolder.isDirectory()){
            combineFolder(inputFolder, inputFolder.getName());
        }
        else {
            combineFile(inputFolder, inputFolder.getName());
        }

        writeStream.close();

    } catch(java.io.IOException e) {
        System.out.println("Cannot combine, please check folder in config file");
    }
}

/*
* This method opens a file content in fileReading and appends it to
* writeStream
* Input :
* fileReading file to read
* Exception :
* IOException throw IOException when it can't create FileInputStream
*/

private void combineFile(File fileReading, String folderName) throws IOException
{
    int i = 0;
    int j = 0;
    int k = 0;
    int fileLength = 0;
    int stringLength = 0;

    String temp;
    String result;
    FileInputStream in = new FileInputStream(fileReading);
    byte bt[] = new byte[(int)fileReading.length()];

    in.read(bt);
    fileLength = bt.length;

    // convert data in a file to string temp
    temp = new String(bt);
    temp.trim();

```

```

// parsing all characters in this file. If a character is a tab key
// check if asserting folder name to result file or not.
stringLength = temp.length();

while (temp.indexOf("IncidentID:") > -1)
{
    // get index of '.'
    i = temp.indexOf('.');

    // just consider the line that contains
    // a digit before '.', (cleared at )
    if ( ((i-3) >= 0) &&
        Character.isDigit(temp.charAt(i-1)) &&
        Character.isDigit(temp.charAt(i-2)) &&
        (temp.charAt(i-3) == '.'))
    {
        // Find IncidentID
        j = temp.indexOf("IncidentID:");
        j = j + 12;
        k = j;
        while (temp.charAt(k) != '\t') k++;
        incidentID = temp.substring(j, k);
        //System.out.println(incidentID);

        // Find LinkID
        j = temp.indexOf("LinkID:");
        j = j + 8;
        k = j;
        while (temp.charAt(k) != '\r') k++;
        linkID = temp.substring(j, k);
        //System.out.println(linkID);

        // Find incidentStartDate
        j = temp.indexOf(' ');
        k = j;
        while (temp.charAt(k) != ')') k++;
        incidentStartDate = temp.substring(j+1, k);
        //System.out.println(incidentStartDate);

        // Find incidentStartTime
        j = k;
        k = j + 1;
        while (temp.charAt(k) != ')') k++;
        incidentStartTime = temp.substring(j+1, k);
        //System.out.println(incidentStartTime);

        // Find typeOfIncident
        typeOfIncident = temp.substring(k+1, k+2);
        //System.out.println(typeOfIncident);

        // Find incidentEndTime
        j = temp.indexOf('.');
        k = j;
        while (temp.charAt(k) != ')') k--;
        incidentEndTime = temp.substring(k+1, j);
        //System.out.println(incidentEndTime);
    }
}

```

```

// Find incidentEndDate
j = k;
k = j-1;
while (temp.charAt(k) != '(') k--;
incidentEndDate = temp.substring(k+1, j);
//System.out.println(incidentEndDate);

// just consider new incident
if ( ! ( incidentID.equals(pIncidentID) &&
        linkID.equals(pLinkID) &&
        incidentStartDate.equals(pIncidentStartDate) &&
        incidentStartTime.equals(pIncidentStartTime) &&
        typeOfIncident.equals(pTypeOfIncident) &&
        incidentEndTime.equals(pIncidentEndTime) &&
        incidentEndDate.equals(pIncidentEndDate)
      ) )
{

    //System.out.println(pIncidentID + "\t" + pLinkID);

    result = "";
    result = result.concat(incidentID).concat("\t");
    result = result.concat(linkID).concat("\t");
    result = result.concat(incidentStartDate).concat("\t");
    result = result.concat(incidentStartTime).concat("\t");
    result = result.concat(typeOfIncident).concat("\t");
    result = result.concat(incidentEndDate).concat("\t");
    result = result.concat(incidentEndTime).concat("\r\n");

    writeStream.write(result.getBytes());

    //System.out.print(result);

    pIncidentID = incidentID;
    pLinkID = linkID;
    pIncidentStartDate = incidentStartDate;
    pIncidentStartTime = incidentStartTime;
    pTypeOfIncident = typeOfIncident;
    pIncidentEndTime = incidentEndTime;
    pIncidentEndDate = incidentEndDate;

}

}

temp = temp.substring(i+1);

}

in.close();
}

/*
 * This method opens a folder content in folderReading and parses this folder.
 *
 * Input :
 * folderReading folder to read

```

```
* Exception :
* IOException throw IOException when it receives from combineFile
*/

private void combineFolder(File folderReading, String folderName) throws IOException
{
    File[] fileList = folderReading.listFiles();

    for (int i = 0; i < fileList.length ; i++){
        if (fileList[i].isDirectory()){
            combineFolder(fileList[i], fileList[i].getName());
        }
        else{
            combineFile(fileList[i], folderName);
        }
    }
}

public static void main(String[] args){
    System.out.println(" Starting ");
    System.out.println(" Running, please wait..... ");
    AppendData appendData = new AppendData();
    appendData.combine();
    System.out.println(" Finishing, please review the result. ");
}
}
```

APPENDIX B1: RTDB – FTP

(Data connection and transfer using: File Transfer Protocol (FTP)

Data type: Incident data; Duration: Sep. 2002 to Jun. 2004)

```
import com.enterprisedt.net.ftp.FTPException;
import com.enterprisedt.net.ftp.FTPClient;
import com.enterprisedt.net.ftp.FTPTransferType;
import com.enterprisedt.net.ftp.FTPConnectMode;
import java.util.*;
import java.io.*;
import java.sql.*;
import java.text.*;

public class FTPClientLoop{
    //Database variables
    static Connection con = null;
    static String str_query = " ";
    static Statement stmt = null;
    static ResultSet rs = null;

    public static void main(String args[]) {
        // getting remote host, user name and password
        if (args.length < 4) {
            System.out.println(args.length);
            usage();
            System.exit(1);
        }
        try {
            // assign args to make it clear
            String host = args[0];
            String user = args[1];
            String password = args[2];
            String directory = args[3];
            String local_path = args[4];
            int dataLength = Integer.parseInt(args[5]); //maximum length of database
            int retainDataLength = Integer.parseInt(args[6]); //data length to be retained

            //downloaded file parameters
            String new_dir = " ";
            java.util.Date this_time, sDate, eDate, mDate;
            DataOutputStream dos;
            String file_path;
            int count = 0;
            File[] fileList;

            // ODBC data source
            String dsn = "jdbc:odbc:Incident";
            String db_user = " ";
            String db_pass = " ";

            //open database connection
            Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
            con = DriverManager.getConnection(dsn, db_user, db_pass);
```



```

while(true){
    //get new data from FTP server
    // connect
    FTPClient ftp = new FTPClient(host, 21);
    ftp.login(user, password);

    // PASV
    ftp.setConnectMode(FTPConnectMode.PASV);
    ftp.setConnectMode(FTPConnectMode.ACTIVE);

    // change dir
    ftp.chdir(directory);

    // ASCII transfer
    ftp.setType(FTPTransferType.ASCII);

    //create a new directory to store new files
    this_time = new java.util.Date();
    new_dir = local_path + this_time.toString().replace(':', '-');
    File output_dir = new File(new_dir);
    output_dir.mkdir();

    //get new files
    String[] listings = ftp.dir();
    for (int i = 0; i < listings.length; i++){
        file_path = new_dir + "\\" + listings[i];
        dos = new DataOutputStream(new FileOutputStream(file_path));
        ftp.get(dos, listings[i]);
    }

    //File output_dir = new File("E:\\Programming\\TempDB\\data");
    fileList = output_dir.listFiles();
    for (int i = 0; i < fileList.length ; i++){
        //parse files to get fields and put into database
        parseFile(fileList[i]);
    }

    //number of times data is recorded
    count = count + 1;

    //check to delete old data
    stmt = con.createStatement
    (ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_UPDATABLE);
    boolean hasResults = stmt.execute("Select * from tb_TempIncident");

    if(hasResults){
        rs = stmt.getResultSet();
        rs.first();

        //get timestamp of the first and last row
        sDate = new java.util.Date(rs.getString("TStamp"));
        //System.out.println(sDate.toString());
        rs.last();
        eDate = new java.util.Date(rs.getString("TStamp"));
        //System.out.println(eDate.toString());

        if(dateDiff(sDate,eDate) >= dataLength){
            //if dataLength is reached, proceed to delete data

```

```

        //get the lowest row number to be retained
        rs.first();
        mDate = new java.util.Date(rs.getString("TStamp"));
        while(dateDiff(sDate,mDate) < retainDataLength){
            rs.next();
            mDate = new java.util.Date(rs.getString("TStamp"));
            //System.out.println(mDate.toString());
        }
        //delete all rows before the retained row number
        boolean del = stmt.execute("Delete * From tb_TempIncident
        where ID < " + rs.getInt("ID"));
    }
}

System.out.println("Wrote new files " + count + " times. Wait for another 5
mins...");

    //quit ftp
    ftp.quit();
    Thread.currentThread().sleep(300000);
}
//end while, close database connection
}

catch(Exception ex){
    System.out.println("Caught exception: " + ex.getMessage());
}
}

/* Basic usage statement
public static void usage() {
    System.out.println("Usage: ");
    System.out.println("FTPClientLoop " + "remotehost user password directory
    local_directory");
}

/* This method opens a file content in fileReading and set values for data fields
    * Input :
    * fileReading file to read
    * Exception:
    * IOException throws IOException when it can't create FileInputStream

public static void parseFile(File fileReading) throws IOException
{
    // data of an incident
    String incidentID = " ";
    String linkID = " ";
    String incidentStartDate = " ";
    String incidentStartTime = " ";
    String incidentEndDate = " ";
    String incidentEndTime = " ";
    String typeOfIncident = " ";
    String tStamp = " ";

    // data of a previous incident
    String pIncidentID = " ";
    String pLinkID = " ";
    String pIncidentStartDate = " ";

```

```

String pIncidentStartTime = " ";
String pIncidentEndDate = " ";
String pIncidentEndTime = " ";
String pTypeOfIncident = " ";
int i = 0;
int j = 0;
int k = 0;
int fileLength = 0;
int stringLength = 0;

String temp;
Locale currentLocale = new Locale("en", "US");
DateFormat formatter = DateFormat.getDateInstance(
    (DateFormat.MEDIUM, DateFormat.MEDIUM, currentLocale));

FileInputStream in = new FileInputStream(fileReading);
byte bt[] = new byte[(int)fileReading.length()];

in.read(bt);
fileLength = bt.length;

// convert data in a file to string temp
temp = new String(bt);
temp.trim();

// parsing all characters in this file. If a character is a tab key
// check if asserting folder name to result file or not.
stringLength = temp.length();

while (temp.indexOf("IncidentID:") > -1)
{
    // Find IncidentID
    j = temp.indexOf("IncidentID:");
    j = j + 12;
    k = j;
    while (temp.charAt(k) != '\t') k++;
    incidentID = temp.substring(j, k);
    //System.out.println(incidentID);

    // Find LinkID
    j = temp.indexOf("LinkID:");
    j = j + 8;
    k = j;
    while (temp.charAt(k) != '\r') k++;
    linkID = temp.substring(j, k);
    //System.out.println(linkID);

    // Find incidentStartDate
    j = temp.indexOf(" ");
    k = j;
    while (temp.charAt(k) != ')') k++;
    incidentStartDate = temp.substring(j+1, k);
    //System.out.println(incidentStartDate);

    // Find incidentStartTime
    j = k;
    k = j + 1;
    while (temp.charAt(k) != ')') k++;

```

```

incidentStartTime = temp.substring(j+1, k);
//System.out.println(incidentStartTime);

// Find typeOfIncident
typeOfIncident = temp.substring(k+1, k+2);
//System.out.println(typeOfIncident);

//check if this is a cleared incident
// get index of '.'
i = temp.indexOf('.');

if ( ((i-3) >= 0) &&
    Character.isDigit(temp.charAt(i-1)) &&
    Character.isDigit(temp.charAt(i-2)) &&
    (temp.charAt(i-3) == ':'))
{
    // Find incidentEndTime
    j = temp.indexOf('.');
    k = j;
    while (temp.charAt(k) != ')') k--;
    incidentEndTime = temp.substring(k+1, j);
    //System.out.println(incidentEndTime);

    // Find incidentEndDate
    j = k;
    k = j-1;
    while (temp.charAt(k) != '(') k--;
    incidentEndDate = temp.substring(k+1, j);
    //System.out.println(incidentEndDate);
}
else {
    //reset incident end date, time
    incidentEndTime = " ";
    incidentEndDate = " ";
}

//just consider new incidents
if ( ! ( incidentID.equals(pIncidentID) &&
    linkID.equals(pLinkID) &&
    incidentStartDate.equals(pIncidentStartDate) &&
    incidentStartTime.equals(pIncidentStartTime) &&
    typeOfIncident.equals(pTypeOfIncident) &&
    incidentEndTime.equals(pIncidentEndTime) &&
    incidentEndDate.equals(pIncidentEndDate)
    ))
{
    try{
        //check if linkID is in database
        str_query = "Select LinkID from tb_LinkID";
        stmt = con.createStatement();
        if(stmt.execute(str_query)){
            //put fields into TempIncident
            tStamp = formatter.format(new java.util.Date()).toString();

            str_query = "INSERT INTO
            tb_TempIncident(incidentid,linkid,startdate,starttime,incidenttyp
            e,enddate,endtime,tstamp) VALUES( '"+incidentID+"','"+

```

```

linkID + ";" + incidentStartDate + ";" + incidentStartTime +
";" + typeOfIncident + ";" + incidentEndDate + ";" +
incidentEndTime + ";" + tStamp + ")";
//System.out.println(str_query);
stmt.execute(str_query);
}
}catch(Exception ex){
System.out.println(ex.toString());
}
pIncidentID = incidentID;
pLinkID = linkID;
pIncidentStartDate = incidentStartDate;
pIncidentStartTime = incidentStartTime;
pTypeOfIncident = typeOfIncident;
pIncidentEndTime = incidentEndTime;
pIncidentEndDate = incidentEndDate;

}

//go to next incident
temp = temp.substring(i+1);
}
in.close();
}

*/ This method calculates eDate - sDate and return the difference in hours
*/ Input : Two date objects. Exception: None
*/
public static int dateDiff(java.util.Date sDate, java.util.Date eDate){
    long l1 = sDate.getTime();
    long l2 = eDate.getTime();
    long difference = l2 - l1;
    return Math.round(difference/(1000*60*60));
}
}

```

APPENDIX B2: RTDB –XML

(Data connection and transfer using: Extensible Markup Language XML)

Data type: Incident and traffic data. Duration: Since Jul., 2004)

```
// xmltransDlg.cpp : implementation file
#include "stdafx.h"
#include "xmltrans.h"
#include "xmltransDlg.h"
#include ".\xmltransdlg.h"
#include "CXMLDOMNode.h"
#include <string.h>
#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// CxmltransDlg dialog

#define WM_APP_ADD_DB      WM_APP+1

CxmltransDlg::CxmltransDlg(CWnd* pParent /*=NULL*/)
: CDialog(CxmltransDlg::IDD, pParent)
, m_intInterval(1)
, m_strDataFile(_T(""))
, m_strControlFile(_T(""))
, m_strLog(_T(""))
, m_strMDBFile(_T(""))
, m_pXMLDOMData(NULL)
, m_pXMLDOMCtrl(NULL)
, m_bThreadTerminated(FALSE)
, m_bThreadRunning(false)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CxmltransDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    DDX_Text(pDX, IDC_EDIT_INTERVAL, m_intInterval);
    DDV_MinMaxInt(pDX, m_intInterval, 1, 999999);
    DDX_Text(pDX, IDC_EDIT_DATA, m_strDataFile);
    DDV_MaxChars(pDX, m_strDataFile, 1024);
    DDX_Text(pDX, IDC_EDIT_CONTROL, m_strControlFile);
    DDV_MaxChars(pDX, m_strControlFile, 1024);
    DDX_Text(pDX, IDC_EDIT_LOG, m_strLog);
    DDX_Text(pDX, IDC_EDIT_MDBFILE, m_strMDBFile);
    DDX_Control(pDX, IDC_EDIT_LOG, m_ctrlLog);
    DDX_Control(pDX, IDC_BUTTON_STOP, m_ctrlStop);
    DDX_Control(pDX, IDC_BUTTON_START, m_ctrlStart);
}

BEGIN_MESSAGE_MAP(CxmltransDlg, CDialog)
    ON_WM_PAINT()
```

```

ON_WM_QUERYDRAGICON()
//}}AFX_MSG_MAP
ON_BN_CLICKED(IDOK, OnBnClickedOk)
ON_BN_CLICKED(IDC_BUTTON_DATA BROWSER, OnBnClickedButtonDatabrowser)
ON_BN_CLICKED(IDC_BUTTON_CONTROL BROWSE, OnBnClickedButtonControlbrowse)
// ON_BN_CLICKED(IDC_BUTTON1, OnBnClickedButton1)
ON_BN_CLICKED(IDC_BUTTON_MDB BROWSER, OnBnClickedButtonMdbbrowser)
// ON_BN_CLICKED(IDC_BUTTON3, OnBnClickedButton3)
ON_BN_CLICKED(IDC_BUTTON_STOP, OnBnClickedButtonStop)
ON_BN_CLICKED(IDC_BUTTON_START, OnBnClickedButtonStart)
ON_BN_CLICKED(IDC_BUTTON1, OnBnClickedButton1)
END_MESSAGE_MAP()

```

// CxmltransDlg message handlers

```

BOOL CxmltransDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Set the icon for this dialog. The framework does this automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE); // Set big icon
    SetIcon(m_hIcon, FALSE); // Set small icon

    // TODO: Add extra initialization here
    m_pXMLDOMData = new CXMLDOMDocument();
    m_pXMLDOMCtrl = new CXMLDOMDocument();
    // new will throw exception automatically
    HRESULT hr = NULL;
    COleException ex;
    hr = m_pXMLDOMData->CreateDispatch("Microsoft.XMLDOM", &ex);
    if (!SUCCEEDED(hr)) {
        delete m_pXMLDOMData;
        m_pXMLDOMData = NULL;
        delete m_pXMLDOMCtrl;
        m_pXMLDOMCtrl = NULL;
        AfxThrowOleException(ex.m_sc);
    }
    m_pXMLDOMData->put_async(FALSE);

    hr = m_pXMLDOMCtrl->CreateDispatch("Microsoft.XMLDOM", &ex);
    if (!SUCCEEDED(hr)) {
        delete m_pXMLDOMData;
        m_pXMLDOMData = NULL;
        delete m_pXMLDOMCtrl;
        m_pXMLDOMCtrl = NULL;
        AfxThrowOleException(ex.m_sc);
    }
    m_pXMLDOMCtrl->put_async(FALSE);
    m_bThreadRunning = false;
    return TRUE; // return TRUE unless you set the focus to a control
}

```

*// If you add a minimize button to your dialog, you will need the code below
 // to draw the icon. For MFC applications using the document/view model,
 // this is automatically done for you by the framework.*

```

void CxmltransDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }
}

// The system calls this function to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR CxmltransDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

void CxmltransDlg::OnBnClickedOk()
{
    // TODO: Add your control notification handler code here
    if (m_bThreadRunning) {
        AfxMessageBox(_T("Please wait for working thread to finish."));
    }
    else {
        releaseXMLObj();
        OnOK();
    }
}

void CxmltransDlg::OnBnClickedButtonDatabrowser()
{
    // TODO: Add your control notification handler code here
    // szFilters is a text string that includes two file name filters:
    // "*.my" for "MyType Files" and "*.xml" for "All Files."
    const char *szFilters="XML Files (*.xml)|*.xml|All Files (*.*)|*.*|";

    // Create an Open dialog; the default file name extension is ".my".
    CFileDialog fileDlg (TRUE, "xml", "*.xml",
        OFN_FILEMUSTEXIST| OFN_HIDEREADONLY, szFilters, this);

    // Display the file dialog. When user clicks OK, fileDlg.DoModal()
    // returns IDOK.
    if( fileDlg.DoModal ()==IDOK )

```



```

{
    m_strDataFile = fileDlg.GetPathName();
    logString(CString("Choose data file"));
    UpdateData(FALSE);
}
}

void CxmltransDlg::OnBnClickedButtonControlbrowse()
{
    // TODO: Add your control notification handler code here
    // szFilters is a text string that includes two file name filters:
    // "*.my" for "MyType Files" and "*.*" for "All Files."
    const char *szFilters="XML Files (*.xml)|*.xml|All Files (*.*)|*.*|";

    // Create an Open dialog; the default file name extension is ".my".
    CFileDialog fileDlg (TRUE, "xml", "*.xml",
        OFN_FILEMUSTEXIST| OFN_HIDEREADONLY, szFilters, this);

    // Display the file dialog. When user clicks OK, fileDlg.DoModal()
    // returns IDOK.
    if( fileDlg.DoModal ()==IDOK )
    {
        m_strControlFile = fileDlg.GetPathName();
        logString(CString("Choose control file"));
        UpdateData(FALSE);
    }
}

void CxmltransDlg::OnBnClickedButtonMdbbrowser()
{
    // TODO: Add your control notification handler code here
    // szFilters is a text string that includes two file name filters:
    // "*.my" for "MyType Files" and "*.*" for "All Files."
    const char *szFilters="Access database (*.mdb)|*.xml|All Files (*.*)|*.*|";

    // Create an Open dialog; the default file name extension is ".my".
    CFileDialog fileDlg (TRUE, "mdb", "*.mdb",
        OFN_FILEMUSTEXIST| OFN_HIDEREADONLY, szFilters, this);

    // Display the file dialog. When user clicks OK, fileDlg.DoModal()
    // returns IDOK.
    if( fileDlg.DoModal ()==IDOK )
    {
        m_strMDBFile = fileDlg.GetPathName();
        UpdateData(FALSE);
    }
}

int CxmltransDlg::logString(CString& logStr)
{
    m_strLog += logStr;
    m_strLog += "\r\n";
    //TRACE("FIXME: should scroll log window to the last line");
    //fix me: scroll to last line here
    //UpdateData(FALSE);
    m_ctrlLog.SetWindowText(m_strLog);
    return 0;
}

```

```

#ifdef 0
void CxmltransDlg::OnBnClickedButton1()
{
    // TODO: Add your control notification handler code here
    // Test button
        workingThread(this);
        return ;
}

void CxmltransDlg::OnBnClickedButton3()
{
    // TODO: Add your control notification handler code here
    COleException ex;
    CXMLDOMDocument* pDom = new CXMLDOMDocument();
    LPDISPATCH lpDis = NULL;
    BOOL result=FALSE;
    VARIANT src;
    CString dataTag=_T("TData");
    CXMLDOMNodeList* pNodeList = NULL;
    HRESULT hr=0;

    hr = pDom->CreateDispatch("Microsoft.XMLDOM", &ex);
    if (!SUCCEEDED(hr)) {
        AfxThrowOleException(ex.m_sc);
    }
    pDom->put_async(FALSE);
    VariantInit(&src);
    src.vt = VT_BSTR;
    src.bstrVal = m_strDataFile.AllocSysString();

    result = pDom->load(src);

    if (result) {
        long nodeNo = 0;
        logString(CString("Load XML successful"));
        if (pDom->hasChildNodes()) {
            logString(CString("XML has child nodes"));
        }
        else {
            logString(CString("XML has no child nodes"));
        }
        lpDis = pDom->getElementsByTagName((LPCTSTR)dataTag);
        pNodeList=new CXMLDOMNodeList(lpDis);
        pNodeList->reset();
        nodeNo = pNodeList->get_length();
        if (nodeNo == 0) {
            logString(CString("No data tag found"));
        }
        else {
            CString msg;
            msg.Format(_T("Found %ld data tag"),nodeNo);
            logString(msg);
        }
        pNodeList->ReleaseDispatch();
        delete pNodeList;
        pNodeList = NULL;
    }
}

```

```

    }
    else {
        logString(CString("Load XML successful"));
    }
    pDom->ReleaseDispatch();
    delete pDom;
    SysFreeString(src.bstrVal);
    src.bstrVal = NULL;
    UpdateData(FALSE);
}

#endif

UINT CxmltransDlg::workingThread(LPVOID pSelfObj)
{
    VARIANT src;
    int r = 0;
    HRESULT hr = NULL;
    BOOL result = FALSE;
    CString msg;
    CxmltransDlg *pSelf = (CxmltransDlg*) pSelfObj;
    CXMLDOMDocument *pXMLCtrl = NULL;
    CXMLDOMDocument *pXMLData = NULL;
    CString xmlData;
    // should give proper init time here
    COleDateTime lastCheckedTime( 1970, 1, 1, 0, 0, 0 );
    COleDateTimeSpan timeElapsed;
    double secondElapsed = 0.0;

    //UpdateData();

    TRACE("\nFIXME: should have proper init time in lastCheckedTime variable");

    if (pSelf == NULL) {
        AfxMessageBox(_T("Critical error, can not proceed"));
        pSelf->m_bThreadRunning = false;
        return 0;
    }
    pXMLCtrl = pSelf->m_pXMLDOMCtrl;
    pXMLData = pSelf->m_pXMLDOMData;

    pSelf-
>logString(CString(_T("=====")));
    pSelf->logString(CString(_T("Thread started")));
    // first, load the control xml file
    msg.Format(_T("Control XML file is: %s"),(LPCTSTR) pSelf->m_strControlFile);
    pSelf->logString(msg);
    msg.Format(_T("Data XML file is: %s"),(LPCTSTR) pSelf->m_strDataFile);
    pSelf->logString(msg);
    msg.Format(_T("MDB file is: %s"),(LPCTSTR) pSelf->m_strMDBFile);
    pSelf->logString(msg);

    pSelf->logString(CString(_T("Start polling")));
    TRACE("\nFIXME: use custom file loading to avoid null chars at the end of file. Should use
method load() when xml control file is compliant to standard.");
    //VariantInit(&src);
    //src.vt = VT_BSTR;

```

```

//src.bstrVal = pSelf->m_strControlFile.AllocSysString();
do {
    //result = pXMLCtrl->load(src);
    //because control file tends to end with NULL chars -> need to load to a string to
    //avoid that first
    //better algorithm should be used here
    result = pSelf->loadXMLCtrlFile(pSelf->m_strControlFile, xmlData);
    result = pXMLCtrl->loadXML((LPCTSTR)xmlData);
    if (result) {
        // read timestamp
        LPDISPATCH lpDis = NULL;
        CXMLDOMNodeList *pNodeList = NULL;
        long nodeNo = 0;
        CString dataTag = _T("timestamp");
        lpDis = pXMLCtrl->getElementsByTagName((LPCTSTR)dataTag);
        pNodeList=new CXMLDOMNodeList(lpDis);
        pNodeList->reset();
        nodeNo = pNodeList->get_length();
        if (nodeNo == 0) {
            pSelf->logString(CString("No timestamp tag found"));
        }
        else {
            CString msg;
            msg.Format(_T("Found %ld timestamp tag"),nodeNo);
            pSelf->logString(msg);
        }

        LPDISPATCH lpnode=pNodeList->get_item(0);
        CXMLDOMNode aNode(lpnode);

        CString anItem;
        anItem=aNode.get_text();
        TRACE("\nFIXME: should have better date/time parsing method here");
        // remove .0 (fraction of second) at the end
        // because date/time only parse up to second. Putting fraction of second
        // would result an error.
        anItem.TrimRight(_T(".0"));
        msg.Format(_T("Check time: %s"), anItem);
        pSelf->logString(anItem);
        // now, convert date str to time
        COleDateTime parsedDate;
        bool parseOK=false;
        parseOK = parsedDate.ParseDateTime((LPCTSTR) anItem);
        if (parseOK) {
            pSelf->logString(CString(_T("Parse date/time ok")));
            // calculated elapsed
            timeElapsed = parsedDate - lastCheckedTime;
            secondElapsed = timeElapsed.GetTotalSeconds();
            msg.Format(_T("Time span in second: %lf"), secondElapsed);
            pSelf->logString(msg);
            //pSelf->UpdateData();
            msg.Format(_T("Check interval is: %d"),pSelf->m_intInterval);
            pSelf->logString(msg);
            if (secondElapsed > (double)pSelf->m_intInterval){
                // perform update db action now
                pSelf->logString(CString(_T("Start update db cycle")));
                r = 0;
            }
        }
    }
}

```

```

        r = pSelf->updateDatabase(pSelf, pSelf-
        >m_strDataFile, pSelf->m_strMDBFile);
        if (r) {
            pSelf->logString(CString(_T("db update
            error")));
        }
        else {
            pSelf->logString(CString(_T("db update
            successful")));
        }
        lastCheckedTime = parsedDate;
    }
}
else {
    pSelf->logString(CString(_T("Parse date/time fail")));
}

aNode.ReleaseDispatch();
pNodeList->ReleaseDispatch();
delete pNodeList;
pNodeList = NULL;
}
else {
    pSelf->logString(CString(_T("Load XML control file failed, can not
    proceed")));
}
// sleep for 1s
//pSelf->logString(CString(_T("Waiting for next cycle...")));
Sleep(1000);
}
while (!pSelf->m_bThreadTerminated);
//::SysFreeString(src.bstrVal);
pSelf->logString(CString(_T("Thread stopped")));
pSelf->m_ctrlStart.EnableWindow(TRUE);
pSelf->m_ctrlStop.EnableWindow(FALSE);
pSelf->m_bThreadRunning = false;
return 0;
}

// Release XML DOM objs
int CxmltransDlg::releaseXMLObj(void)
{
    if (m_pXMLDOMCtrl!=NULL) {
        m_pXMLDOMCtrl->ReleaseDispatch();
        delete m_pXMLDOMCtrl;
        m_pXMLDOMCtrl = NULL;
    }
    if (m_pXMLDOMData!=NULL) {
        m_pXMLDOMData->ReleaseDispatch();
        delete m_pXMLDOMData;
        m_pXMLDOMData = NULL;
    }
    return 0;
}

void CxmltransDlg::OnCancel(void)
{
    if (m_bThreadRunning) {

```

```

        AfxMessageBox(_T("Please wait for working thread to finish."));
    }
    else {
        releaseXMLObj();
        CDialog::OnCancel();
    }
}

void CxmltransDlg::OnBnClickedButtonStop()
{
    // TODO: Add your control notification handler code here
    m_bThreadTerminated = TRUE;
    UpdateData(FALSE);
}

// load and pre-validate data in control file
// FIXME: load 4096 byte in control file only
int CxmltransDlg::loadXMLCtrlFile(const CString& filename, CString& data)
{
    CFile sourceFile;
    unsigned char buffer[4096];
    DWORD dwRead;
    // we'll use a CFileException object to get error information

    CFileException ex;
    // open the source file for reading

    if (!sourceFile.Open(filename, CFile::modeRead | CFile::shareDenyWrite, &ex))
    {
        // complain if an error happened
        // no need to delete the ex object
        TCHAR szError[1024];
        ex.GetErrorMessage(szError, 1024);
        CString msg;
        msg.Format(_T("Couldn't open file: %s"), szError);
        AfxMessageBox(msg);
        return 1;
    }
    else
    {
        // Read in 4096-byte blocks, hope it'll be enough
        // fix me: better reading here
        TRACE("\nFIXME: should have better file handling here");
        dwRead = sourceFile.Read(buffer, 4095);
        if (dwRead < 0) dwRead = 0;
        buffer[dwRead] = '\0';
        sourceFile.Close();
        data = buffer;
    }
    return 0;
}

int CxmltransDlg::updateDatabase(CxmltransDlg *pSelf, const CString& xmlDataFile, const
CString& dbFile)
{
    //CDAODatabase db;
    // load xml file first
    VARIANT src;

```

```

HRESULT hr = NULL;
BOOL result = FALSE;
CString msg;
int accumulateErrors = 0;

CXMLDOMDocument *pXMLData = NULL;

if (pSelf == NULL) {
    AfxMessageBox(_T("Critical error, can not proceed"));
    return 0;
}

pXMLData = pSelf->m_pXMLDOMData;

pSelf->logString(CString(_T("Start loading XML data file")));
VariantInit(&src);
src.vt = VT_BSTR;
src.bstrVal = pSelf->m_strDataFile.AllocSysString();
result = pXMLData->load(src);
if (result) {
    pSelf->logString(CString(_T("Load XML data file successful")));

    // read timestamp
    LPDISPATCH lpDis = NULL;
    CXMLDOMNodeList *pNodeList = NULL;
    long nodeNo = 0;
    CString dataTag = _T("TData");
    lpDis = pXMLData->getElementsByTagName((LPCTSTR)dataTag);
    pNodeList = new CXMLDOMNodeList(lpDis);
    pNodeList->reset();
    nodeNo = pNodeList->get_length();
    if (nodeNo == 0) {
        pSelf->logString(CString("No data tag found"));
    }
    else {
        CString msg;
        msg.Format(_T("Found %ld data tag"), nodeNo);
        pSelf->logString(msg);
    }
}
/*
// open database
try
{
    if (pSelf->m_objDatabase.IsOpen())
        pSelf->m_objDatabase.Close();
    pSelf->m_objDatabase.Open(dbFile);
}
catch( CException* e )
{
    //AfxMessageBox(
    //e->m_pErrorInfo->m_strDescription,
    //MB_ICONEXCLAMATION );
    e->ReportError();
    // Delete the incomplete recordset object
    e->Delete();
    pNodeList->ReleaseDispatch();
    delete pNodeList;
    return -1;
}

```

```

}
// Successful open the db
*/
// iterate each node
for (int i=0;(i<nodeNo) && (accumulateErrors <3) && (!pSelf-
>m_bThreadTerminated); i++) {
LPDISPATCH lpnode=pNodeList->get_item(i);
CXMLDOMNode aNode(lpnode);
if (aNode.hasChildNodes()){
    // read data and add record to db
    CTime currentTime=CTime::GetCurrentTime();
    CString entryDate="";
    CString entryID="";
    CString entrySpeed="";
    CString entryVolume="";
    CString entryUpdate =currentTime.Format(_T ("%d/%m/%Y
%H:%M:%S"));

    LPDISPATCH lpChildList=aNode.get_childNodes();
    CXMLDOMNodeList childList(lpChildList);
    for (int j=0;j<childList.get_length() && (accumulateErrors <3);j++) {
        LPDISPATCH lpChild=childList.get_item(j);
        CXMLDOMNode aChild(lpChild);
        if (aChild.get_nodeName() == _T("Date")) {
            entryDate = aChild.get_text();
        }
        else if (aChild.get_nodeName() == _T("LinkID")) {
            entryID = aChild.get_text();
        }
        else if (aChild.get_nodeName() == _T("Speed")) {
            entrySpeed = aChild.get_text();
        }
        else if (aChild.get_nodeName() == _T("Volume")) {
            entryVolume = aChild.get_text();
        }
        aChild.ReleaseDispatch();
    } // end for child
    childList.ReleaseDispatch();
    if (entryDate!="") {
        unsigned char *sqlString = NULL;
        // only keep info for non null date
        // fix this for proper validation
        msg.Format(_T("Try add to db with:
(%s,%s,%s,%s)",entryDate, entryID, entrySpeed, entryVolume);
        pSelf->logString(msg);

        CString updateSQL;
        updateSQL.Format(_T("INSERT INTO DataTable VALUES
(%s,%s,%s,%s,%s)",entryDate, entryID, entrySpeed,
entryVolume, entryUpdate);
        TRACE(updateSQL);
        sqlString = new unsigned char[updateSQL.GetLength()+1];
        //memset(sqlString, 0, updateSQL.GetLength()+1);
        strcpy((char *)sqlString,updateSQL);
        // send back sql cmd to GUI thread so that
        CDaoDatabase.Execute() is executed in that context
        result = pSelf->SendMessage(WM_APP_ADD_DB, 0,
(LPARAM)sqlString);
    }
}

```



```

        if (result) {
            accumulateErrors++;
            pSelf->logString(CString(_T("DAO exception happens")));
        }
        /*
        try {
            // Execute SQL statement
            CString updateSQL;
            updateSQL.Format(_T("INSERT INTO DataTable
VALUES ( '%s',%s,%s,%s', '%s');"), entryDate, entryID,
entrySpeed, entryVolume, entryUpdate);
            TRACE(updateSQL);
            TRACE("\nFIXME: should use transaction instead of
update sql for every record");
            //db.Execute(updateSQL);
            //pSelf->m_objDatabase.Execute(updateSQL);
        }
        catch (CDaoException* e) {
            AfxMessageBox(
e->m_pErrorInfo->m_strDescription,
MB_ICONEXCLAMATION );
            // Delete the incomplete recordset object
            e->Delete();
            accumulateErrors++;
            pSelf->logString(CString(_T("DAO exception happens")));
        } // end catch
        */
    } // end if entryDate != ""
} // end if has child
else {
    pSelf->logString(CString(_T("Empty data tag found")));
}
aNode.ReleaseDispatch();
}

pSelf->logString(CString(_T("Close database")));
//db.Close();
//pSelf->m_objDatabase.Close();
pNodeList->ReleaseDispatch();
delete pNodeList;
}
else {
    pSelf->logString(CString(_T("Load XML data file failed")));
    return -1;
}
return accumulateErrors;
}

void CxmItransDlg::OnBnClickedButtonStart()
{
    // TODO: Add your control notification handler code here
    UpdateData();
    if ((m_strMDBFile != "") && (m_strControlFile != "") && (m_strDataFile != "")) {
        m_ctrlStop.EnableWindow(TRUE);
        m_bThreadTerminated = FALSE;
        m_bThreadRunning = true;
        // prepare database connection (?)
    }
}

```

```

        try
        {
            if(m_objDatabase.IsOpen())
                m_objDatabase.Close();
            m_objDatabase.Open(m_strMDBFile);

            AfxBeginThread(CxmltransDlg::workingThread,(LPVOID) this);
            m_ctrlStart.EnableWindow(FALSE);
        }
        catch( CException* e )
        {
            //AfxMessageBox(
            //e->m_pErrorInfo->m_strDescription,
            //MB_ICONEXCLAMATION );
            e->ReportError();
            // Delete the incomplete recordset object
            e->Delete();
        }
    }
    else {
        AfxMessageBox(_T("Please choose Access database file, XML data file, and XML
control file"));
    }
}

void CxmltransDlg::OnBnClickedButton1()
{
    // TODO: Add your control notification handler code here
    //CDaoDatabase db;
    UpdateData();
    // open database
    try
    {
        if(m_objDatabase.IsOpen())
            m_objDatabase.Close();
        m_objDatabase.Open(m_strMDBFile);
    }
    catch( CException* e )
    {
        //AfxMessageBox(
        //e->m_pErrorInfo->m_strDescription,
        //MB_ICONEXCLAMATION );
        e->ReportError();
        // Delete the incomplete recordset object
        e->Delete();
    }
    // Successful open the db
    //db.Close();
    m_objDatabase.Close();
}

LRESULT CxmltransDlg::WindowProc(UINT message, WPARAM wParam, LPARAM lParam)
{
    if(message == WM_APP_ADD_DB){
        CString sqlCommand;
        int retcode = 0;
        sqlCommand = (unsigned char *) lParam;
        TRACE("\nExec sql cmd %s", sqlCommand);
    }
}

```

```
//return 1;
try {
    // Execute SQL statement
    m_objDatabase.Execute(sqlCmd);
}
catch (CDAOException* e) {
    AfxMessageBox(
        e->m_pErrorInfo->m_strDescription,
        MB_ICONEXCLAMATION );
    // Delete the incomplete recordset object
    e->Delete();
    retcode = 1;
} // end catch

delete[] (unsigned char *)lParam;
return retcode;
}
else
    return CWnd::WindowProc(message, wParam, lParam);
}
```

APPENDIX C: CAR-FOLLOWING MODEL

THE TSC ARCHITECTURE

APPENDIX D: TRAFFIC CONTROLLER

THE TSC ARCHITECTURE

2.4 RAMP DIVERSION

Note: Ramp "arrival" means the traffic that actually enters the ramp.

Ramp "departure" (see Block 2: TRAFFIC CONTROLLER) means traffic that actually be metered into the mainline.

*The criteria for passing Input 1 is Input 2 ≥ 60 ; otherwise the data will be passed through Input 3.
 (if the ramp queue reaches 60 vehicles, there will be no more traffic entering the ramp, ramp "arrival" equals zero. if the ramp queue is less than 60, ramp "arrival" equals the ramp demand).*