
CONTENTS

Gas Turbine Abbreviations	xxi
1 Preliminaries	1
1.0 Introduction	1
1.1 Outline of Chapters	2
1.2 Contributions to Current Research	5
2 Controller and Engine Simulation Implementation	6
2.0 Introduction	6
2.1 Engines – An Introduction	6
2.1.1 Compressor	7
2.1.2 Combustion Chamber	7
2.1.3 Turbine	8
2.2 Gas Turbine Types	8
2.3 Afterburning or Reheat	9
2.4 Requirements and Current Trends in Engine and Controller Design	9
2.5 Compressor Surge	10
2.6 Engine Modelling	11
2.6.1 Fuel System	12
2.6.2 Gas Generator Dynamics	13
2.6.3 Turbine Blade Temperature (TBT)	14
2.6.4 Engine Thrust	14
2.7 The Controller	14
2.8 Engine Starting	15
2.8.1 Temperature Limiting During Starting	15
2.9 Normal Operation	16
2.9.1 Governor Control	17
2.9.2 Acceleration Control Loop	17
2.9.3 Deceleration Control Loop	17
2.9.4 Overfuelling and Underfuelling Limiter Tasks	18
2.9.5 TBT Limiter	18
2.9.6 Low Pressure Turbine Control	19
2.10 Reheat and Nozzle Control	19
2.10.1 Nozzle Control	20
2.10.2 Fuel Flow Demand Control	20
2.10.3 Reheat Sequencing Control	21

2.11	Baseline Simulation	21
2.11.1	Controller Simulation	21
2.12	TSIM Simulation Implementation	22
2.13	Pole-Zero Mapping	25
2.14	Matlab Simulation Implementation	25
2.15	Concluding Remarks	28
2.15.1	Engine Modelling	28
2.15.2	Baseline Simulation	28
2.15.3	Digital Simulation	28
3	Controller and Engine Simulation on the Inmos Transputer	29
3.0	Introduction	29
3.1	Occam and the Transputer	29
3.1.1	Von Neumann Computers	30
3.1.2	Inmos Transputer	31
3.1.2.1	Inmos Serial Links	32
3.1.2.2	Memory	32
3.2	Occam	32
3.2.1	The Occam Constructs	33
3.2.2	Further Aspects of Occam	35
3.3	Occam Simulation of Engine Model	35
3.3.1	Operating Modes of Simulation	35
3.3.2	Simulation Initialisation	36
3.3.3	Program Options	37
3.3.4	Results from Simulation	37
3.4	Multi-Processor Implementation	40
3.4.1	Engine Simulation Load Balancing	41
3.4.2	Controller Simulation	43
3.4.3	Modified Controller Simulation	44
3.5	Load Balancing	46
3.5.1	Mapping Considerations	47
3.5.2	Controller Mapping onto 2 Processors	49
3.5.3	Controller Mapping onto 3 and 4 Processors	51
3.6	Model Enhancements	53
3.6.1	Addition of Reheat	53
3.6.2	Reheat Mapping onto 2 Processors	55
3.6.3	Reheat Mapping onto 3 and 4 Processors	56
3.6.4	Start-up Control and In-flight Relighting	58
3.7	Concluding Remarks	59
3.7.1	The Transputer and Occam	59
3.7.2	Mapping Issues	59
3.7.3	Performance Estimation and Improvements	59
3.7.4	Architectural Issues	60
3.7.5	Achievements	60

4	Alternative Approaches to Parallel Processing – The Butterfly Plus and Helios Systems	62
	Introduction	62
4.1	The Helios Operating System	63
4.1.1	Helios Shell Interface	63
4.1.2	I/O Server	63
4.1.3	Helios C Compiler	63
4.1.4	Static and Dynamic Load Balancing	64
4.2	Interprocess Communication	64
4.2.1	Communication	64
4.2.2	Communication Failure	66
4.2.3	Synchronisation	66
4.2.4	Implementation of Communication	66
4.3	Programming in Helios	66
4.3.1	Modifications Needed to C Code	66
4.3.2	Debugging	66
4.3.3	Assignment of Parallel Tasks	67
4.3.4	Task Force Manager (TFM)	68
4.4	Analysis of System Performance	68
4.4.1	Timing Using the Real-Time Clock	68
4.4.2	Execution Times	69
4.4.3	Task Force Manager Overheads	70
4.4.4	Consistency of Timings	71
4.5	General Comments on the Helios System	71
4.6	Conclusions	71
4.6.1	Suitability for Real-Time Control	71
4.6.2	Fault Tolerance	71
4.6.3	General Comments	72
4.6.4	Reliability	72
4.6.5	Farm Construct	72
4.7	The Butterfly Plus Computer	73
4.8	Overview of Hardware	74
4.8.1	Processor Node Card	74
4.8.2	Switch Interface and Deadlock Handling	75
4.8.2.1	Receiver	76
4.8.2.2	Transmitter	76
4.8.3	Switch Card	77
4.8.3.1	Example of Butterfly Switch Efficiency	78
4.8.3.2	Switch Protocol	78
4.8.3.3	Switch Protocol Example	79
4.8.3.4	Switch Contention Handling	79
4.9	Programming the Butterfly Plus	81
4.9.1	Introduction	81
4.9.2	Utilities Available	81
4.9.3	Communication Synchronisation	81

4.9.4	Programming Alternatives	81
4.9.5	Uniform System Approach	82
4.10	Summary of Software Design	83
4.11	Comparison of Sun 3 and Butterfly Plus	83
4.12	GIST Analysis	84
4.12.1	Introduction	84
4.12.2	Typical GIST Trace	84
4.12.3	Parallel Mapping	84
4.13	GIST Analysis of Program	84
4.13.1	System Performance: 1–5 Processors	86
4.13.2	Detailed Analysis of System Performance on 4 and 5 Processors	88
4.13.2.1	Four Processors – Engine Simulation	88
4.13.2.2	Four Processors – Controller Simulation	88
4.13.2.3	Five Processors – Engine Simulation	91
4.13.2.4	Five Processors – Controller Simulation	91
4.13.3	Detailed Analysis of System Performance on 4 and 5 Processors – Concluding Remarks	91
4.14	Task Length Modification	92
4.14.1	Modified Simulation Results	92
4.15	Summary of Butterfly Plus Results	94
4.15.1	Program Organisation	94
4.15.2	Task Generators	94
4.15.3	Real-Time Scheduler	95
4.16	Comparison of Helios and Butterfly Plus	96
4.17	Concluding Remarks	96
4.17.1	Summary	96
4.17.2	Helios Operating System	97
4.17.3	Butterfly Plus and Uniform System	97
4.17.4	Automatic Load Balancing	98
5	Formal Methods and System Specifications	99
5.0	Introduction	99
5.1	System Requirements	100
5.2	The Need for Formal Methods	102
5.2.1	The Advantages of Formal Methods	102
5.2.2	The Need for a Mathematical Specification	103
5.2.3	The Structure of Formal Methods	104
5.2.3.1	The Vienna Development Method (VDM)	105
5.2.3.2	The Z Specification Language	105
5.3	Summary of Disadvantages of Formal Methods	106
5.4	System Validation and Verification	107
5.4.1	System Validation	107
5.4.2	System Verification	108

5.5	Conventional Testing	108
5.5.1	Static Code Analysis	109
5.5.2	Automated Tools for Analysis	109
5.6	Formal Methods for Hardware Specification	110
5.6.1	ELLA	110
5.6.2	VIPER	110
5.6.3	Formal Methods Applied to the T800 Transputer Floating Point Unit	111
5.7	Application of Formal Methods to Transputers	112
5.7.1	Problems of Asynchronous Processors	112
5.7.2	The Trace Model of CSP	114
5.8	Use of Formal Methods in Fault Tolerant Systems	115
5.8.1	Fail-Safe or Shut-Down Systems	115
5.8.2	Control and Protection Systems	115
5.8.3	Formal Specification of N-Lane Replication ($N > 1$) ..	115
5.8.4	Reconfigurable Systems	116
5.9	Specifications for Gas Turbine Systems	116
5.9.1	Top-Level Specification of Gas Turbine Controller Software	116
5.9.2	Review of CSAN 1454 Draft Guidelines Document with Regard to Parallel Processing and in Particular to the use of Transputer Arrays	117
5.9.3	Intercommunication	117
5.9.4	Input-Output Conditioning	118
5.9.5	Cycle Segregation	119
5.9.6	Interrupts	120
5.9.7	Fault Detection and System Monitoring Principles ...	120
5.9.8	Precision of Monitoring	121
5.9.9	Independence of Comparison	121
5.9.10	Scope of Self-Check Program	122
5.9.11	Protection Against Dormant Faults	122
5.9.12	Computer Autonomy	122
5.9.13	Fault Modes and Effects	123
	5.9.13.1 Fault Categories	123
	5.9.13.2 Spurious Faults	124
5.9.14	Final Failure Case	124
5.10	Concluding Remarks	125
5.10.1	Summary	125
5.10.2	Perceived Benefits of Formal Methods	125
5.10.3	Design of Gas Turbine Controller Software	125
6	Failure Management and its Application in Gas Turbine Engine Control	126
6.0	Summary	126
6.1	Introduction	126

6.2	The History of Fault Tolerance in Computers	127
6.3	Characterisation of Faults	127
6.3.1	Design Faults	128
6.3.2	Operational Faults	128
6.3.3	Byzantine Generals Disagreement	129
6.3.4	Power Supply Failure	130
6.3.5	Fault Categories Addressed	130
6.4	Desired Response from a Fault Tolerant System	130
6.5	Fault Tolerant Techniques	131
6.5.1	Fault Tolerant Techniques	131
6.6	Hardware Redundancy	131
6.6.1	Active Replication	131
6.6.2	Passive Replication	132
6.7	Software Redundancy	134
6.8	Designing Fault Tolerance into the Process	137
6.9	Evaluation Measures	138
6.10	Application of Fault Tolerance to Systems	138
6.10.1	Long-Life Applications	138
6.10.2	Critical Computations	138
6.10.3	Maintenance Avoidance	138
6.10.4	Availability	139
6.11	Fault Tolerant Processor Topologies	139
6.11.1	Loop Topology	139
6.11.2	Tree Architectures	140
6.11.3	Array Processors	140
6.12	Review of Fault Tolerant Designs Already Implemented	141
6.13	Design and Implementation of Fault Tolerant Gas Turbine Engine Controllers	141
6.13.1	Aim of the Investigation	141
6.14	Design Constraints of Aero-Engines	142
6.14.1	Design Philosophy	142
6.15	Backward Error Recovery/TMR Scheme (BER/TMR)	143
6.15.1	Normal Mode	146
6.15.2	Voter/Identifier Mode	146
6.15.3	Conclusions From BER/TMR Method	147
6.15.4	Summary of BER/TMR System	148
6.16	Method of Overlapping Triads (DTMR System)	149
6.16.1	Evolution of the Processor Topology	149
6.16.2	Reliability Analysis of the Configuration	151
6.16.3	Input Data Validation	152
6.16.4	Controller Task Calculation	154
6.16.5	Data Output Validation	159
6.16.6	Relay Selection of Output DACs	159
6.16.7	Link Adaptor Shutdown Logic	159
6.16.8	System Testing	160

6.17	Communication Channel Failure	162
6.17.1	System Synchronisation	162
6.17.2	Communication Failure Detection and Accommodation	164
6.17.3	Cascade Approach	165
6.17.4	"Lemming" Approach	165
6.17.5	Self-Adjusting Time Frame Technique	165
6.17.6	Link Failure Handling Software	167
6.17.7	Limitations Imposed by Link Failure Handling	167
6.17.8	Modified Input Data Validation	170
6.17.9	Conclusions of "Overlapping Triads" Method	170
6.18	"Hot Sparring" Technique	173
6.18.1	Data Input	174
6.18.2	Controller Task Implementation	174
6.18.3	Data Output	176
6.18.4	Byzantine Disagreement Handling	176
6.18.5	Problems Encountered with the Inmos C004	176
6.18.6	Voted Deselect	177
6.18.7	Link Relays	177
6.18.8	Conclusions of "Hot Sparring" Implementation	179
6.19	Concluding Remarks	181
6.19.1	Summary	181
6.19.2	BER/TMR Backward Error Recovery System	181
6.19.3	Overlapping Triad System	181
6.19.4	Hot Sparring System	181
6.19.5	Transputer Architectural Restrictions	182
7	Concluding Remarks	183
7.0	Project Motivation	183
7.1	Gas Turbine Engine Modelling	183
7.2	Transputer Implementation	183
7.3	Comparison Using Diverse Architectures	185
7.4	System Requirements and Formal Methods	186
7.5	Fault Tolerant Systems	187
7.6	Areas for Further Research	188
7.6.1	T9000 Series of Transputers	189
7.6.2	Future Software Developments for the T9000	190
	Appendices	191
A	Hardware Development	191
A.1	Introduction	191
A.2	Single Eurocard Development Rack	192
A.3	Inmos Strategy	192

A.4	Overview of Transputer Products	193
A.4.1	T414 Transputer	193
A.4.2	T800 Transputer	194
A.4.3	T212 Transputer	194
A.4.4	M212 Transputer	194
A.4.5	C004 Programmable Link Switch	195
A.4.6	The Inmos Link Adaptor	196
A.5	The Transputer Architecture	197
A.5.1	Clock	197
A.5.2	Analyse and Error	197
A.5.3	Event Request and Event Acknowledge	197
A.5.4	External Memory Interface	197
A.5.5	External Memory Interface Program	199
A.5.6	Reset	200
A.5.7	Booting	200
A.6	Development Transputer Card	202
A.6.1	Static RAM Board	202
A.6.2	Dynamic RAM Board	203
A.6.3	Software Testing	204
A.7	The Single Eurocard Transputer Card	204
A.8	Standalone Network Loader Card	204
A.8.1	Design Overview	207
A.8.2	EPROM Programming	208
A.9	Reset/Analyse Debugging	208
A.9.1	Debugging Facilities	208
A.9.2	Debugger Operation	209
A.10	Programmable Link Switch Card	209
A.10.1	Performance of Link Switch	210
A.11	12-Bit Transputer Multi-Channel ADC Card	211
A.12	12-Bit Transputer Multi-Channel DAC Card	211
A.13	VME Bus Interface Card	212
A.13.1	VME Bus Transputer Interface (VMETI)	213
A.14	Summary	215
B	Helios Default Maps	216
C	Text of Section 21 of DS 00-55	217
D	Evaluation Measures of Fault Tolerant Techniques	218
D.1	Terminology	218
D.1.1	Coverage	218
D.1.2	Reliability	218
D.1.3	Mission Time	219
D.1.4	Availability	219
D.1.5	Testability and Verifiability	219
D.1.6	Other Important Factors	219

D.2	Quantitative Measures	220
D.2.1	Combinatorial Modelling	220
D.2.2	Markov Modelling	220
D.2.3	Reliability Analysis of Passive Replication	221
D.2.4	Series Reliability	222
D.2.5	Parallel Reliability	222
D.2.6	Simplex System	222
D.2.7	Duplex System	223
D.2.8	M of N Systems	223
D.2.9	Reliability Comparison	224
D.2.10	MTTF Comparison	224
D.2.11	Mission Time Comparison	225
E.	Overview of Fault Tolerant Designs Already Realised	227
E.1	Tandem 16 Nonstop	227
E.2	Augusta A129	228
E.3	The General Purpose Digital Controller (GPDC)	228
E.4	SIFT	229
E.5	FTMP	229
E.6	MAFT	229
E.7	FASP	230
E.8	Fault Tolerance in VLSI (The Intel iAPX 432)	230
F.	The Fault Integrator	232
G.	Formal Expression of Overlapping Triads Technique	236
G.1	The "Overlapping Triads" Technique	236
G.1.1	Voting	237
G.1.2	Voting Action	237
G.2	Monitor Action	238
G.2.1	Fault Categories	238
G.3	Example	239
G.3.1	Reconfiguration	239
G.3.2	Failure of Monitor Transputers	239
H.	Self-Test Procedure	241
I.	Formal Expression of Extension of Overlapping Triads to Hot Sparing System	246
I.1	The "Hot Sparing" Technique	246
I.1.1	Voting	247
I.1.2	Voting Action	247
I.2	Monitoring Action	248
I.2.1	Fault Categories	248

I.3	Example	250
I.3.1	Reconfiguration	252
I.3.2	Failure in Monitoring of Transputers	252
References		254
Subject Index		261