

Contents

1	The Paradigm of Self-optimization	1
1.1	From Mechatronics to Intelligent Technical Systems	2
1.2	Introduction to Self-optimization	5
1.3	Architecture of Self-optimizing Systems	8
1.3.1	Structure of Self-optimizing Mechatronic Systems	8
1.3.2	Operator-Controller-Module	10
1.4	Self-optimization in Intelligent Technical Systems	12
1.4.1	Model-Based Self-optimization	12
1.4.2	Behavior-Oriented Self-optimization	16
1.4.3	Self-optimization by Reconfiguration	19
1.5	Structure of This Book	22
	References	23
2	Examples of Self-optimizing Systems	27
2.1	Rail Technology – RailCab	27
2.1.1	Self-optimizing Operating Point Control	29
2.1.2	intelligent Drive Module (iDM)	33
2.1.3	Active Guidance Module	36
2.1.4	Active Suspension Module	38
2.1.5	Hybrid Energy Storage System (HES)	42
2.1.6	Crosslinked Test Benches	46
2.1.7	Convoy Mode	49
2.2	Miniature Robot BeBot	50
2.2.1	Basic Vehicle	50
2.2.2	Extension Modules	51
2.2.3	Operating System	53
2.2.4	Implementing Self-X Properties	54
2.3	X-by-Wire Test Vehicle	56
2.3.1	Vehicle Dynamics	57
2.3.2	Self-optimizing Integrated Vehicle-Dynamics Control	58
	References	61

3	Development of Self-optimizing Systems	65
3.1	Design Methodology for Self-optimizing Systems	66
3.2	Domain-Spanning Conceptual Design	69
3.3	Domain-Specific Design and Development	74
3.3.1	Mechanical Engineering	76
3.3.2	Control Engineering	78
3.3.3	Software Development	82
3.3.4	Electrical and Electronic Engineering	88
3.3.5	(Sub)system Integration	91
3.3.6	Interaction of the Domains in the Design and Development	93
3.4	Self-optimizing Development Process	96
3.4.1	Framework of a Self-optimizing Development Process	96
3.4.2	Systematic Planning of Synchronizations	105
	References	113
4	Methods for the Domain-Spanning Conceptual Design	117
4.1	Specification Technique CONSENS for the Description of Self-optimizing Systems	119
4.2	Software Support for the Specification of the Principle Solution	127
4.3	Consistency Analysis of Application Scenarios	129
4.4	Design of the System of Objectives	133
4.5	Design Framework for the Integration of Cognitive Functions Based on Solution Patterns	139
4.5.1	Systems Analysis	139
4.5.2	Functional Description	141
4.5.3	Solution Selection	143
4.5.4	Systems Specification	147
4.6	Product Structuring for Self-optimizing Systems	154
4.7	Early Probabilistic Reliability Analysis Based on the Principle Solution	162
4.8	Evaluation of the Economic Efficiency	171
	References	180
5	Methods for the Design and Development	183
5.1	Automatic Model Transformation and Synchronization	186
5.1.1	Example Scenario	186
5.1.2	Deriving Initial Domain-Specific Models from the System Model	188
5.1.3	Synchronizing Models during the Domain-Specific Refinement Phase	193
5.2	Software Design	197
5.2.1	Component Model	198

5.2.2	Decompose Communication Requirements	199
5.2.3	Real-Time Coordination Patterns	200
5.2.4	Discrete Behavior	210
5.2.5	Simulation of Hybrid Behavior	220
5.2.6	Specification of Deployment	221
5.2.7	Integration of Self-healing Behavior	221
5.2.8	Code Generation	222
5.3	System Optimization	222
5.3.1	Set-Oriented Multiobjective Optimization	224
5.3.2	Hierarchical Multiobjective Optimization	229
5.3.3	Hierarchical Modeling of Mechatronic Systems	232
5.3.4	Parametric Multiobjective Optimization	236
5.3.5	Computation of Robust Pareto Points	237
5.3.6	Optimal Control of Mechanical and Mechatronic Systems	239
5.3.7	Motion Planning with Motion Primitives	244
5.3.8	Hierarchical Hybrid Planning	246
5.3.9	Statistical Planning	250
5.3.10	Behavior Planning in Nondeterministic Environment ...	253
5.3.11	FIPA Conform Cross-Domain Communication	257
5.3.12	Preparing Solution Pattern "Hybrid Planning"	261
5.4	Dynamic Reconfiguration	265
5.4.1	Fine-Grained Reconfigurable Architectures	265
5.4.2	Coarse-Grained Reconfigurable Architectures	267
5.4.3	Modelling	267
5.4.4	Design Methods for Dynamic Reconfigurable Systems	271
5.4.5	Platforms and Applications	279
5.5	System Software	291
5.5.1	Architecture for Self-optimizing Operating Systems ...	293
5.5.2	Self-optimized Flexible Resource Management	297
5.5.3	Self-optimization in the Operating System	299
5.5.4	Hierarchical Flexible Resource Manager	305
5.6	Virtual Prototyping	310
5.6.1	Virtual Prototypes and Virtual Environments	311
5.6.2	Automatic Model Linking	313
5.6.3	Visualization Agents	320
5.6.4	Virtual Test Bench	330
	References	337
6	Summary and Outlook	351
Index		353