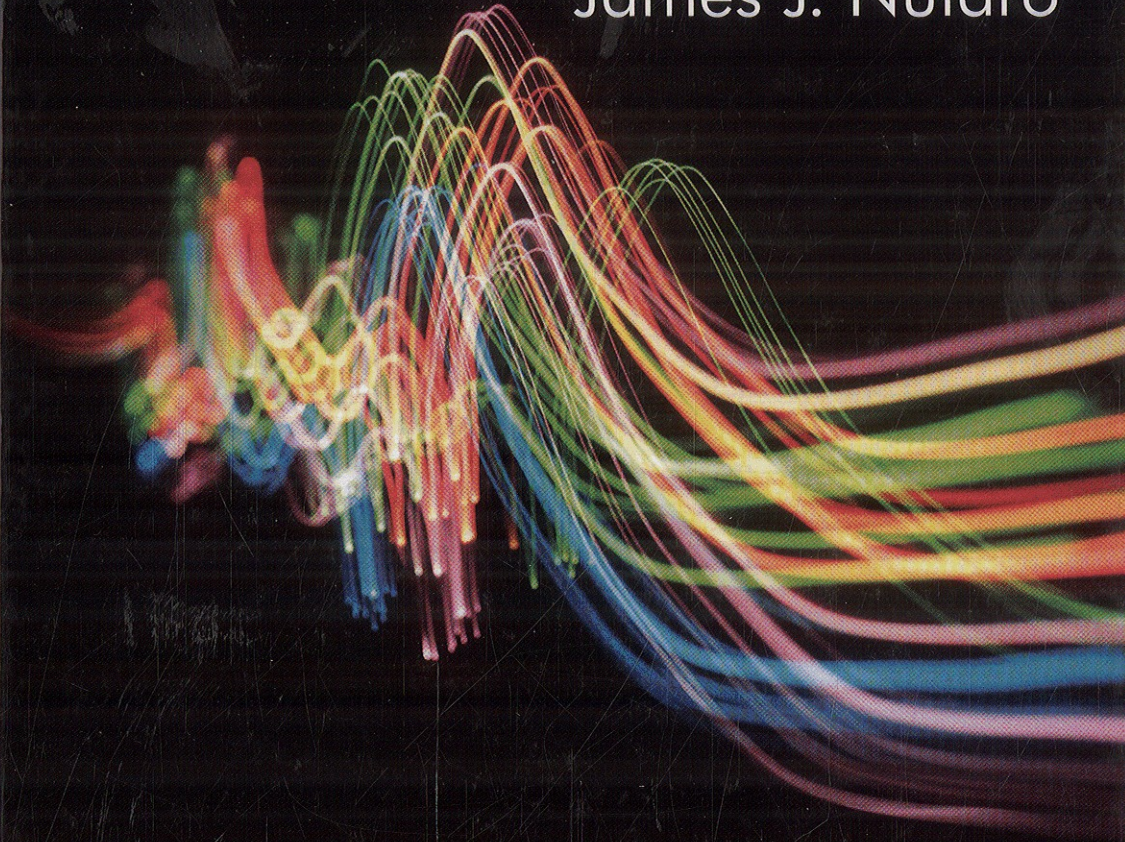


James J. Nutaro

An abstract graphic consisting of numerous thin, curved lines in various colors (red, orange, yellow, green, blue, purple) that appear to be flowing or vibrating, creating a sense of motion and energy. The lines are set against a solid black background.

Building Software *for* Simulation

THEORY AND ALGORITHMS,
WITH APPLICATIONS IN C++

 WILEY

CONTENTS

PREFACE

ix

1 INTRODUCTION

1

- 1.1 Elements of a Software Architecture / 2
- 1.2 Systems Concepts as an Architectural Foundation / 4
- 1.3 Summary / 5
- 1.4 Organization of the Book / 6

2 FIRST EXAMPLE: SIMULATING A ROBOTIC TANK

- 2.1 Functional Modeling / 8
- 2.2 A Robotic Tank / 9
 - 2.2.1 Equations of Motion / 11
 - 2.2.2 Motors, Gearbox, and Tracks / 13
 - 2.2.3 Complete Model of the Tank's Continuous Dynamics / 17
 - 2.2.4 The Computer / 18
 - 2.2.5 Complete Model of the Tank / 22
- 2.3 Design of the Tank Simulator / 23
- 2.4 Experiments / 25
- 2.5 Summary / 30

3 DISCRETE-TIME SYSTEMS

32

- 3.1 Atomic Models / 33

3.1.1	Trajectories /	33
3.1.2	The State Transition and Output Function /	35
3.1.3	Two Examples of Atomic, Discrete-Time Models /	39
3.1.4	Systems with Bags for Input and Output /	42
3.1.5	A Simulator for Atomic Models /	42
3.2	Network Models /	53
3.2.1	The Parts of a Network Model /	54
3.2.2	The Resultant of a Network Model /	55
3.2.3	An Example of a Network Model and Its Resultant /	56
3.2.4	Simulating the Resultant /	61
3.3	A Simulator for Discrete-Time Systems /	77
3.4	Mealy/Moore-Type Systems /	89
3.5	Cellular Automata /	91
3.6	Summary /	98

4 DISCRETE-EVENT SYSTEMS

100

4.1	Atomic Models /	101
4.1.1	Time and Trajectories /	101
4.1.2	The State Transition Function /	103
4.1.3	The Output Function /	105
4.1.4	Legitimate Systems /	106
4.1.5	An Example of an Atomic Model /	107
4.1.6	The Interrupt Handler in the Robotic Tank /	110
4.1.7	Systems with Bags for Input and Output /	114
4.1.8	A Simulator for Atomic Models /	114
4.1.9	Simulating the Interrupt Handler /	118
4.2	Network Models /	125
4.2.1	The Parts of a Network Model /	125
4.2.2	The Resultant of a Network Model /	126
4.2.3	An Example of a Network Model and Its Resultant /	128
4.2.4	Simulating the Resultant /	132
4.3	A Simulator for Discrete-Event Systems /	143
4.3.1	The Event Schedule /	144
4.3.2	The Bag /	153
4.3.3	The Simulation Engine /	157
4.4	The Computer in the Tank /	170
4.5	Cellular Automata Revisited /	176
4.6	Summary /	180

5 HYBRID SYSTEMS**182**

- 5.1 An Elementary Hybrid System / 185
- 5.2 Networks of Continuous Systems / 186
- 5.3 Hybrid Models as Discrete-Event Systems / 187
- 5.4 Numerical Simulation of Hybrid Systems / 190
- 5.5 A Simulator for Hybrid Systems / 198
- 5.6 Interactive Simulation of the Robotic Tank / 211
 - 5.6.1 Correcting the Dynamics of a Turn / 211
 - 5.6.2 A Simplified Model of the Motor / 213
 - 5.6.3 Updating the Display / 218
 - 5.6.4 Implementing the Tank Physics / 219
- 5.7 Approximating Continuous Interaction Between Hybrid Models / 225
- 5.8 A Final Comment on Cellular Automata / 229
 - 5.8.1 Differential Automata with Constant Derivatives / 229
 - 5.8.2 Modeling Asynchronous Cellular Automata with Differential Automata / 230
 - 5.8.3 A Homomorphism from Differential Automata to Asynchronous Cellular Automata / 232
- 5.9 Summary / 236

6 APPLICATIONS**237**

- 6.1 Control Through a Packet-Switched Network / 237
 - 6.1.1 Model of the Pendulum and Its PID Controller / 238
 - 6.1.2 Integration with an Ethernet Simulator / 244
 - 6.1.3 Experiments / 249
- 6.2 Frequency Regulation in an Electrical Power System / 255
 - 6.2.1 Generation / 257
 - 6.2.2 Transmission Network and Electrical Loads / 259
 - 6.2.3 Frequency Monitoring and Load Actuation / 260
 - 6.2.4 Software Implementation / 261
 - 6.2.5 Experiments / 262
- 6.3 Summary / 269

7 THE FUTURE**271**

- 7.1 Simulation Programming Languages / 271
- 7.2 Parallel Computing and Discrete-Event Simulation / 273
- 7.3 The Many Forms of Discrete Systems and Their Simulators / 276
- 7.4 Other Facets of Modeling and Simulation / 277

APPENDIX A	DESIGN AND TEST OF SIMULATIONS	279
A.1	Decomposing a Model / 280	
A.1.1	Bottom-Up Testing / 280	
A.1.2	Invariants and Assertions / 281	
A.2	Input and Output Objects / 281	
A.2.1	Simple Structures / 282	
A.2.2	Unions / 282	
A.2.3	Pointers and Hierarchies of Events / 284	
A.2.4	Mixing Strategies with Model Wrappers / 286	
A.3	Reducing Execution Time / 291	
APPENDIX B	PARALLEL DISCRETE EVENT SIMULATION	296
B.1	A Conservative Algorithm / 298	
B.1.1	Lookahead / 300	
B.1.2	The Algorithm / 303	
B.2	Implementing the Algorithm with OpenMP / 304	
B.2.1	Pragmas, Volatiles, and Locks / 304	
B.2.2	Overview of the Simulator / 308	
B.2.3	The <i>LogicalProcess</i> / 309	
B.2.4	The <i>MessageQ</i> / 318	
B.2.5	The <i>ParSimulator</i> / 321	
B.3	Demonstration of Gustafson's and Amdahl's Laws / 325	
APPENDIX C	MATHEMATICAL TOPICS	331
C.1	System Homomorphisms / 331	
C.2	Sinusoidal State-Steady Analysis / 333	
REFERENCES		335
INDEX		345