

Embedded Robotics

Thomas Bräunl

EMBEDDED ROBOTICS

Mobile Robot Design
and Applications
with Embedded Systems

Third Edition

With 305 Figures and 32 Tables

 Springer

Thomas Bräunl
School of Electrical, Electronic and Computer Engineering
The University of Western Australia
35 Stirling Highway, M018
Crawley, Perth, WA 6009
Australia

ACM Computing Classification (1998): I.2.9, C.3

ISBN 978-3-540-70533-8 e-ISBN 978-3-540-70534-5

Library of Congress Control Number: 2008931405

© 2008, 2006, 2003 Springer-Verlag Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permissions for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover design: KünkelLopka, Heidelberg

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

PREFACE

.....

The EyeBot controller and mobile robots have evolved over more than a decade. This book gives an in-depth introduction to embedded systems and autonomous mobile robots, using the EyeBot controller (EyeCon) and the EyeBot mobile robot family as application examples.

This book combines teaching and research material and can be used for courses in Embedded Systems as well as in Robotics and Automation. We see labs as an essential teaching and learning method in this area and encourage everybody to reprogram and rediscover the algorithms and systems presented in this book.

Although we like simulations for many applications and treat them in quite some depth in several places in this book, we do believe that students should also be exposed to real hardware in both areas, embedded systems and robotics. This will deepen the understanding of the subject area and of course create a lot more fun, especially when experimenting with small mobile robots.

The original goal for the EyeBot project has been to interface an embedded system to a digital camera sensor (EyeCam), process its images locally in real-time for robot navigation, and display results on a graphics LCD. All of this started at a time before digital cameras came to the market – in fact the EyeBot controller was one of the first “embedded vision systems”.

As image processing is always hungry for processing power, this project requires somewhat more than a simple 8-bit microprocessor. Our original hardware design used a 32-bit controller, which was required for keeping up with the data delivered by the image sensor and for performing some moderate image processing on board. Our current design uses a fast state-of-the-art embedded controller in combination with an FPGA as hardware accelerator for low-level image processing operations. On the software application level (application program interface), however, we try to stay compatible with the original system as much as possible.

The EyeBot family includes several driving robots with differential steering, tracked vehicles, omnidirectional vehicles, balancing robots, six-legged walkers, biped android walkers, and autonomous flying and underwater robots. It also comprises simulation systems for driving robots (EyeSim) and underwater

Preface

robots (SubSim). EyeBot controllers are used in several other projects, with and without mobile robots. We use stand-alone EyeBot controllers for lab experiments in a course in Embedded Systems as part of the Electrical Engineering, Computer Engineering, and Mechatronics curriculum, while we and numerous other universities use EyeBot controllers together with the associated simulation systems to drive our mobile robot creations.

Acknowledgments

While the controller hardware and robot mechanics were developed commercially, several universities and numerous students contributed to the EyeBot software collection. The universities involved in the EyeBot project are as follows:

- Technical University München (TUM), Germany
- University of Stuttgart, Germany
- University of Kaiserslautern, Germany
- Rochester Institute of Technology, USA
- The University of Auckland, New Zealand
- The University of Manitoba, Winnipeg, Canada
- The University of Western Australia (UWA), Perth, Australia

The author thanks the following students, technicians, and colleagues: Gerrit Heitsch, Thomas Lampart, Jörg Henne, Frank Sautter, Elliot Nicholls, Joon Ng, Jesse Pepper, Richard Meager, Gordon Menck, Andrew McCandless, Nathan Scott, Ivan Neubronner, Waldemar Spädt, Petter Reinholdtsen, Birgit Graf, Michael Kasper, Jacky Baltes, Peter Lawrence, Nan Schaller, Walter Banks, Barb Linn, Jason Foo, Alistair Sutherland, Joshua Petitt, Axel Waggershauser, Alexandra Unkelbach, Martin Wicke, Tee Yee Ng, Tong An, Adrian Boeing, Courtney Smith, Nicholas Stamatiou, Jonathan Purdie, Jippy Jungpakdee, Daniel Venkitachalam, Tommy Cristobal, Sean Ong, and Klaus Schmitt.

Thanks to the following members for proofreading the manuscript and giving numerous suggestions: Marion Baer, Linda Barbour, Adrian Boeing, Michael Kasper, Joshua Petitt, Klaus Schmitt, Sandra Snook, Anthony Zaknich, and everyone at Springer.

Contributions

A number of colleagues and former students contributed to this book. The author thanks everyone for their effort in putting the material together.

Preface

JACKY BALTES	The University of Manitoba, Winnipeg, contributed to the section on PID control
ADRIAN BOEING	UWA, coauthored the chapters on the evolution of walking gaits and genetic algorithms, and contributed to the section on SubSim and car detection
MOHAMED BOURGOU	TU München, contributed the section on car detection and tracking
CHRISTOPH BRAUNSCHÄDEL	FH Koblenz, contributed data plots to the sections on PID control and on/off control
MICHAEL DRTL	FH Koblenz, contributed to the chapter on AUVs
LOUIS GONZALEZ	UWA, contributed to the chapter on AUVs
BIRGIT GRAF	Fraunhofer IPA, Stuttgart, coauthored the chapter on robot soccer
HIROYUKI HARADA	Hokkaido University, Sapporo, contributed the visualization diagrams to the section on biped robot design
SIMON HAWE	TU München, reimplemented the ImprovCV framework
YVES HWANG	UWA, contributed to the chapter on genetic programming
PHILIPPE LECLERCQ	UWA, contributed to the section on color segmentation
JAMES NG	UWA, coauthored the sections on probabilistic localization, Bug algorithms, and Brushfire algorithm
JOSHUA PETITT	UWA, contributed to the section on DC motors
KLAUS SCHMITT	Univ. Kaiserslautern, coauthored the section on the Ro-BIOS operating system
TORSTEN SOMMER	TU München, contributed the graphics part of the neural network demonstration program
ALISTAIR SUTHERLAND	UWA, coauthored the chapter on balancing robots
NICHOLAS TAY	DSTO, Canberra, coauthored the chapter on map generation
DANIEL VENKITACHALAM	UWA, coauthored the chapters on genetic algorithms and behavior-based systems and contributed to the chapter on neural networks
BERNHARD ZEISL	TU München, coauthored the section on lane detection
EYESIM	Implemented by Axel Waggershauser (V5) and Andreas Koestler (V6), UWA, Univ. Kaiserslautern, and FH Giessen
SUBSIM	Implemented by Adrian Boeing, Andreas Koestler, and Joshua Petitt (V1), and Thorsten Rühl and Tobias Bielohlawek (V2), UWA, FH Giessen, and Univ. Kaiserslautern

Additional Material

Hardware and mechanics of the “EyeCon” controller and various robots of the EyeBot family are available from INROSOFT and various distributors:

<http://inrosoft.com>

All system software discussed in this book, the RoBIOS operating system, C/C++ compilers for Linux and Windows/Vista, system tools, image processing tools, simulation system, and a large collection of example programs are available free from the following website:

<http://robotics.ee.uwa.edu.au/eyebot/>

Third Edition

Almost five years after publishing the original version, we have now completed the third edition of this book. This edition has been significantly extended with new chapters on CPUs, robot manipulators and automotive systems, as well as additional material in the chapters on navigation/localization, neural networks, and genetic algorithms. This not only resulted in an increased page count, but more importantly in a much more complete treatment of the subject area and an even more well-rounded publication that contains up-to-date research results.

This book presents a combination of teaching material and research contents on embedded systems and mobile robots. This allows a fast entry into the subject matter with an in-depth follow-up of current research themes.

As always, I would like to thank all students and visitors who conducted research and development work in my lab and contributed to this book in one form or another.

All software presented in this book, especially the RoBIOS operating system and the EyeSim and SubSim simulation systems can be freely downloaded from the following website:

<http://robotics.ee.uwa.edu.au>

Lecturers who adopt this book for a course can receive a full set of the author’s course notes (PowerPoint slides), tutorials, and labs from this Web site. And finally, if you have developed some robot application programs you would like to share, please feel free to submit them to our Web site.

Perth, Australia, August 2008

Thomas Bräunl

CONTENTS

PART I: EMBEDDED SYSTEMS

1	Robots and Controllers	3
1.1	Mobile Robots	4
1.2	Embedded Controllers	7
1.3	Interfaces	10
1.4	Operating System	13
1.5	References	15
2	Central Processing Unit	17
2.1	Logic Gates	18
2.2	Function Units	23
2.3	Registers and Memory	28
2.4	Retro	30
2.5	Arithmetic Logic Unit	32
2.6	Control Unit	34
2.7	Central Processing Unit	35
2.8	References	47
3	Sensors	49
3.1	Sensor Categories	50
3.2	Binary Sensor	51
3.3	Analog versus Digital Sensors	51
3.4	Shaft Encoder	52
3.5	A/D Converter	54
3.6	Position Sensitive Device	55
3.7	Compass	57
3.8	Gyroscope, Accelerometer, Inclinometer	59
3.9	Digital Camera	62
3.10	References	70
4	Actuators	73
4.1	DC Motors	73
4.2	H-Bridge	76
4.3	Pulse Width Modulation	78
4.4	Stepper Motors	80
4.5	Servos	81
4.6	References	82

Contents

5	Control	83
5.1	On-Off Control	83
5.2	PID Control	89
5.3	Velocity Control and Position Control	94
5.4	Multiple Motors – Driving Straight	96
5.5	V-Omega Interface	98
5.6	References	101
6	Multitasking	103
6.1	Cooperative Multitasking	103
6.2	Preemptive Multitasking	105
6.3	Synchronization	107
6.4	Scheduling	111
6.5	Interrupts and Timer-Activated Tasks	114
6.6	References	116
7	Wireless Communication	117
7.1	Communication Model	118
7.2	Messages	120
7.3	Fault-Tolerant Self-Configuration	121
7.4	User Interface and Remote Control	123
7.5	Sample Application Program	126
7.6	References	127

PART II: MOBILE ROBOT DESIGN

8	Driving Robots	131
8.1	Single Wheel Drive	131
8.2	Differential Drive	132
8.3	Tracked Robots	136
8.4	Synchro-Drive	137
8.5	Ackermann Steering	139
8.6	Drive Kinematics	141
8.7	References	145
9	Omni-Directional Robots	147
9.1	Mecanum Wheels	147
9.2	Omni-Directional Drive	149
9.3	Kinematics	151
9.4	Omni-Directional Robot Design	152
9.5	Driving Program	154
9.6	References	155
10	Balancing Robots	157
10.1	Simulation	157
10.2	Inverted Pendulum Robot	158
10.3	Double Inverted Pendulum	162
10.4	References	163

Contents

11	Walking Robots	165
11.1	Six-Legged Robot Design	165
11.2	Biped Robot Design	168
11.3	Sensors for Walking Robots	172
11.4	Static Balance	174
11.5	Dynamic Balance	175
11.6	References	182
12	Autonomous Planes	185
12.1	Application	185
12.2	Control System and Sensors	188
12.3	Flight Program	189
12.4	References	192
13	Autonomous Vessels and Underwater Vehicles	195
13.1	Application	195
13.2	Dynamic Model	197
13.3	AUV Design Mako	197
13.4	AUV Design USAL	201
13.5	References	204
14	Robot Manipulators	205
14.1	Homogeneous Coordinates	206
14.2	Kinematics	207
14.3	Simulation and Programming	212
14.4	References	213
15	Simulation Systems	215
15.1	Mobile Robot Simulation	215
15.2	EyeSim Simulation System	216
15.3	Multiple Robot Simulation	221
15.4	EyeSim Application	222
15.5	EyeSim Environment and Parameter Files	223
15.6	SubSim Simulation System	228
15.7	Actuator and Sensor Models	230
15.8	SubSim Application	232
15.9	SubSim Environment and Parameter Files	234
15.10	References	237

PART III: MOBILE ROBOT APPLICATIONS

16	Localization and Navigation	241
16.1	Localization	241
16.2	Probabilistic Localization	245
16.3	Coordinate Systems	249
16.4	Environment Representation	251
16.5	Visibility Graph	253
16.6	Voronoi Diagram	255
16.7	Potential Field Method	258

Contents

16.8	Wandering Standpoint Algorithm	259
16.9	Bug Algorithm Family	260
16.10	Dijkstra's Algorithm	263
16.11	A* Algorithm	267
16.12	References	268
17	Maze Exploration	271
17.1	Micro Mouse Contest	271
17.2	Maze Exploration Algorithms	273
17.3	Simulated versus Real Maze Program	281
17.4	References	282
18	Map Generation	283
18.1	Mapping Algorithm	283
18.2	Data Representation	285
18.3	Boundary-Following Algorithm	286
18.4	Algorithm Execution	287
18.5	Simulation Experiments	289
18.6	Robot Experiments	290
18.7	Results	293
18.8	References	294
19	Real-Time Image Processing	297
19.1	Camera Interface	297
19.2	Auto-Brightness	299
19.3	Edge Detection	300
19.4	Motion Detection	302
19.5	Color Space	303
19.6	Color Object Detection	305
19.7	Image Segmentation	310
19.8	Image Coordinates versus World Coordinates	312
19.9	References	314
20	Robot Soccer	317
20.1	RoboCup and FIRA Competitions	317
20.2	Team Structure	320
20.3	Mechanics and Actuators	321
20.4	Sensing	321
20.5	Image Processing	323
20.6	Trajectory Planning	325
20.7	References	330
21	Neural Networks	331
21.1	Neural Network Principles	331
21.2	Feed-Forward Networks	332
21.3	Backpropagation	337
21.4	Neural Network Examples	342
21.5	Neural Controller	343
21.6	References	344

Contents

22	Genetic Algorithms	347
22.1	Genetic Algorithm Principles	348
22.2	Genetic Operators	350
22.3	Applications to Robot Control	352
22.4	Example Evolution	353
22.5	Implementation of Genetic Algorithms	357
22.6	Starman	361
22.7	References	363
23	Genetic Programming	365
23.1	Concepts and Applications	365
23.2	Lisp	367
23.3	Genetic Operators	371
23.4	Evolution	373
23.5	Tracking Problem	374
23.6	Evolution of Tracking Behavior	377
23.7	References	381
24	Behavior-Based Systems	383
24.1	Software Architecture	383
24.2	Behavior-Based Robotics	384
24.3	Behavior-Based Applications	387
24.4	Behavior Framework	388
24.5	Adaptive Controller	391
24.6	Tracking Problem	395
24.7	Neural Network Controller	396
24.8	Experiments	398
24.9	References	400
25	Evolution of Walking Gaits	403
25.1	Splines	403
25.2	Control Algorithm	404
25.3	Incorporating Feedback	406
25.4	Controller Evolution	407
25.5	Controller Assessment	409
25.6	Evolved Gaits	410
25.7	References	413
26	Automotive Systems	415
26.1	Autonomous Automobiles	415
26.2	Automobile Conversion for Autonomous Driving	418
26.3	Computer Vision for Driver-Assistance Systems	420
26.4	Image Processing Framework	421
26.5	Lane Detection	422
26.6	Vehicle Recognition and Tracking	429
26.7	Automatic Parking	433
26.8	References	436
27	Outlook	439

APPENDICES

A	Programming Tools	443
B	RoBIOS Operating System	453
C	Hardware Description Table	495
D	Hardware Specification	511
E	Laboratories	519
F	Solutions.	529
	Index	533