

DISKI 195

**Motion Planning for Manipulators
with Many Degrees of Freedom –
The BB-Method**

Boris Baginski



infix

Motion Planning for Manipulators with Many Degrees of Freedom – The BB-Method

Boris Baginski



infix

Boris Baginski
Technische Universität München
Fakultät für Informatik
Orleansstr. 34
81667 München
baginski@in.tum.de
www.in.tum.de/~baginski

Vollständiger Abdruck der
von der Fakultät für Informatik
der Technischen Universität München
zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. M. Paul
Prüfer der Dissertation:
1. Univ.-Prof. Dr. H.-J. Siebert
2. Univ.-Prof. Dr. Dr. h.c. W. Brauer

Die Dissertation wurde am 24. Juni 1998 bei der Technischen Universität München eingereicht
und durch die Fakultät für Informatik am 31. August 1998 angenommen.

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Baginski, Boris:

Motion planning for manipulators with many degrees of freedom – the
BB-method / Boris Baginski. – Sankt Augustin : Infix, 1999
(Dissertationen zur künstlichen Intelligenz ; Bd. 195)
Zugl.: München, Techn. Univ., Diss., 1998
ISBN 3-89601-195-2

© 1999 Dr. Ekkehard Hundt, „infix“, Ringstr. 32, 53757 Sankt Augustin

Das Werk ist in allen seinen Teilen urheberrechtlich geschützt. Jede Verwertung ohne
ausdrückliche Zustimmung des Verlages ist unzulässig. Das gilt insbesondere für
Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung in und
Verarbeitung durch elektronische Systeme.

Reproduziert von einer Druckvorlage des Autors
Umschlaggestaltung: Art und Media, Bonn
Druck und Verarbeitung: Hundt Druck GmbH, Köln
Printed in Germany

ISSN 0941-5769
ISBN 3-89601-195-2

Preface

The topic of this book – automatic path and trajectory planning – is an important area of robotics. In this decade robotics – similar to some other areas of computer science – was shaped by striving for systems with more intelligence, more autonomy, and more freedom of choice. The conception of such inherently complex systems posed new challenges: in research as well as in the reorientation process of the often traditionally shaped application domains.

In industrial manufacturing these efforts were strengthened and supported by the target of an integrated computer support comprising all steps from design to product. Due to these efforts more and more products and their manufacturing production lines are completely modelled with CAD-tools.

Due to this, product properties cannot be determined only when a prototype is available, but already during the development process by using computer simulation. Examples from the car manufacture are driving characteristics, crashtests, design of manufacturing cells or digital mock-up. This extensive use of computers leads to cost reduction and shorter development cycles.

Within the industrial area we thus find robots, which operate in a well-known – because described by CAD-models – real or simulated environment. In the real environment up to today the robots usually still move only on predefined paths. Unexpected errors cause a halt in the production. In the future autonomy will be used to handle unexpected error situations. Among other tools path planning is mandatory for this. In the simulated environment, for example during the design of the manufacturing cells or within mock-up simulation, suitable collision-free paths have yet to be found. Here computer-aided path planning, for example with the algorithms developed by Mr. Baginski, is important.

Computer-aided path planning is helpful or even indispensable in many other application domains. In order to show the diversity of the domains, the following scenarios are mentioned:

- A multilegged and highly articulated robot executes inspection work in a sewer system (a partially well-known environment).
- A service robot in a hospital delivers meals. It knows the static objects and recognizes moving objects by sensors, e.g. a camera. The robot must find its way to the target and then move his arms to place the meal correctly.
- Virtual humans or other objects with joints, move in a virtual environment to a goal position to grasp an object. The movement of the human altogether and the movement of the individual parts of the body have to be calculated with the constraint "the movement must look natural".

In many scenarios pertinent to industrial manufacturing, paths for robots with six or more joints can now be calculated within a short time by the BB-algorithm presented in this book.

Nevertheless there are still many other path planning problems without satisfying algorithms to solve them. An example is path planning in very tight and complex obstacle spaces, which are typical for digital mock-up simulation. Another example is path planning for hyperredundant robots under many, also nongeometrical boundary conditions. In the last section of this book Mr. Baginski shortly points to some unsolved problems and his ideas for extending the BB-method to their solution.

I would be happy to see a rapid transfer of Mr. Baginski's algorithms for path planning into the industry and hope that many new impulses for research in path planning are triggered by this book.

Munich, October 26, 1998

Hans-Jürgen Siegert

Abstract

A new approach for manipulator motion planning - the BB-method - is introduced. A collision-free path is found by incrementally modifying an initial, usually colliding, path. The method described alters the robot's motion in its physical workspace, utilizing *virtual* rating functions that are based on *reducing* and *expanding* the robot's geometry model. The BB-method is able to plan motions for redundant and hyperredundant manipulators. With this, it stands in contrast to most other motion planning methods usually developed in an abstract configuration space. Due to their complexity, they are not applicable to real robot tasks.

The experimental experiences with the BB-method prove the main thesis of this work: an efficient and effective motion planning scheme can be build upon careful examination and utilization of the *spatial properties* of the motion of a robot manipulator. The algorithmic complexity of collision free path planning is only linear in the number of degrees of freedom, and the effective computational effort is not necessarily increased for an increased number of joints. The BB-method is local and heuristic, therefore it is an incomplete motion planning algorithm. Within this work, all major properties of the BB-method are characterized and analysed in detail.

This work is a basis for further explorations of the various properties, possibilities, extensions and further developements of the BB-method. But nonetheless, efforts were taken to identify all necessary requirements of a motion planning system and to introduce the paradigm of *virtual geometry modification* in a way all aspects can be handled. The existing prototype implementation allows real-time planning (up to a few seconds) of motions for typical six degree of freedom industrial manipulators in practical cases, as well as fast planning for multiple kinematic devices with several tens of joints. The planning allows the assertion of safety distances to handle control and modelling uncertainties, and a local length reduction of the path, yielding motions that require less time and less energy if they are executed in reality.

Kurzfassung

Das BB-Verfahren, ein neuartiges Verfahren zur Bewegungsplanung für Manipulatoren, wird vorgestellt. Eine kollisionsfreie Bahn wird gefunden, indem eine initiale, üblicherweise kollisionsbehaftete Bahn schrittweise verändert wird. Das dargestellte Verfahren modifiziert die Roboterbewegung im Arbeitsraum, unter Ausnutzung *virtueller* Bewertungsfunktionen, die durch *Verkleinerung* und *Vergrößerung* des Geometriemodells des Roboters bestimmt werden. Das BB-Verfahren ist auch für redundante und hyperredundante Manipulatoren anwendbar. Es unterscheidet sich damit ganz wesentlich von den meisten anderen aus der Literatur bekannten Ansätzen. Diese basieren überwiegend auf der Planung in einem abstrakten Konfigurationsraum und sind aufgrund ihrer Komplexität nicht für realistische Anwendungen geeignet.

Die Ergebnisse der Versuche mit dem BB-Verfahren belegen die Kernthese dieser Arbeit: ein leistungsfähiges und praktisch anwendbares Bewegungsplanungssystem kann durch geeignete Auswertung und Nutzbarmachung der *räumlichen Eigenschaften* der Bewegungen eines Manipulators entwickelt werden. Die algorithmische Komplexität der Planung kollisionsfreier Bahnen ist nur linear in der Anzahl der Freiheitsgrade, und der effektive Planungsaufwand wird durch Hinzufügung zusätzlicher Gelenke nicht notwendigerweise vergrößert. Das BB-Verfahren ist ein lokales und heuristisches Planungsverfahren, es ist daher unvollständig. In dieser Arbeit werden alle wesentlichen Eigenschaften des Verfahrens beschrieben und detailliert analysiert.

Diese Arbeit versteht sich als Basis für die weitere Untersuchung der Eigenschaften, Möglichkeiten, Erweiterungen und Ergänzungen des BB-Verfahrens. Nichtsdestotrotz wurde versucht, alle notwendigen Teilaspekte eines Bewegungsplanungssystems zu identifizieren und das Paradigma der *virtuellen Geometrie-verbormung* in einer Weise einzuführen, daß alle relevanten Aspekte berücksichtigt werden können. Die vorliegende prototypische Implementierung ermöglicht die Planung von Bewegungen für typische sechsgelenkige Industrieroboter in praktischen Anwendungsbeispielen in Echtzeit (maximal wenige Sekunden), und ebenso sehr schnelle Planung für komplexe (auch gegabelten) kinematische Strukturen mit beliebig vielen Freiheitsgraden. Dabei können Sicherheitsabstände eingehalten werden, die Modellierungs- und Steuerungsunsicherheiten berücksichtigen, und die Bahnen können lokal verkürzt werden, was in der Praxis zu verringerten Fahrzeiten und verringertem Energieaufwand führt.

Acknowledgements

First, I would like to thank Prof. Dr. H.-J. Siegert for his encouragement and support during my years at his chair. I enjoyed the hours of discussion very much.

I gratefully acknowledge the support of my colleagues and friends at the institute (in no specific order): Andrea Baumann, Andreas Koller, Oliver Schmid, both Stöhrs, Dr. Gerhard Schrott, Stefan Riesner, Thomas Weiser, Martin El-dracher, Till Brychcy, and Dr. Max Fischer.

Special thanks go to the students who spent months with me, listened to my ideas and realized them with enthusiasm: Quidong Ji, Alex Diebald, Matthias Riesch, Reinhard Haslbeck, Dirk Gottschling, Manfred Hockauf, Richard Lechner, Spyridon Kolssouzidis and the others whom I forgot to mention.

Kristina Degn deserves special thanks for her tireless work and skills in proof-reading this thesis.

Finally, I thank Toni for her patience, love, and inspiration of different points of view when discussing all aspects of this work.

LIST OF SYMBOLS

Formula Symbols

The following convention is used: vectors and sets of vectors are written in a lower case bold typeface ($\mathbf{v}$), matrices and vector-valued maps are written in an upper case bold typeface ($\mathbf{M}$), and scalar values are written in italics (d).

$\mathbf{u}, \mathbf{v}, \mathbf{w}$	Cartesian points
G_i	Geometry model of link i
G^{+d}	Model isotropically expanded by d
$\mathbf{R}$	Rotation matrix
$\mathbf{t}$	Translation vector
${}^i\mathbf{T}_j$	Homogeneous transform
${}^i\mathbf{T}_j(\mathbf{q})$	Configuration-dependent transform
${}^s\mathbf{T}_j$	Scaling transform, scaling factor s
n	Number of joints / links
n_i	Number of joints preceeding link i
v_i	Joint preceeding joint i / link preceeding link i
I_i	List of joints preceeding link i
$\mathbf{W}$	Workspace
$\mathbf{W}_{free}$	Free workspace
q_i	Joint value of joint i
$\mathbf{q}$	n -dimensional configuration
$\mathbf{q}_{start}, \mathbf{q}_{goal}$	Start/goal configurations
$\mathbf{C}$	Configuration space
$\mathbf{C}_{free}$	Free configuration space
$\mathbf{C}_{coll}$	Colliding configuration space
$\mathbf{P}$	Path
$\bar{\mathbf{P}}$	Linear path segment
$\hat{\mathbf{P}}$	Polygonal path
δ_{min}	Smallest displacement for path planning
δ_{max}	Largest displacement for path planning
$\Delta_i(\mathbf{q}_a, \mathbf{q}_b)$	Estimated cartesian motion for link i
$\mathbf{c}^a$	Replacement candidate set for configuration $\mathbf{q}_a$

f_{step}	Relative displacement step size
f_{divide}	Path segment subdivision factor
z_{random}	Maximum number of random subgoals
d_{max_i}	Desired safety distance for link i
$\mathbf{d}_{max}$	n -dimensional vector of desired safety distances
$\mathbf{d}_{\bar{p}}$	Safety distances of a path segment
ρ	Threshold ratio for path shortening
λ	Segment length threshold for path shortening
τ	Tolerance for collision detection
γ_{scale}	Approximation precision for scaling based rating
γ_{dist}	Approximation precision for expansion based rating
t_{scale}	Planning time for collision-free path
t_{dist}	Planning time for safety distances
t_{short}	Planning time for path length reduction
m_{dist}	Approximate quality of safety distances
m_{short}	Length ratio achieved by length reduction

Algorithms

To clarify the presentation, prototype algorithms are used throughout this work. A PASCAL-style notation is used for the control structures.

Q_{scale}	Scaling based path rating (p. 51)
ModifySegmentScale	Scaling based path modification (p. 61)
DivideSegmentScale	Path segment subdivision (p. 62)
PlanScale	Collision free path planning (p. 63)
PlanGlobal	Random global planning (p. 74)
Q_{dist}	Expansion based path rating (p. 79)
ModifySegmentDist	Expansion based path modification (p. 83)
DivideSegmentDist	Path segment subdivision (p. 85)
PlanDist	Planning safety distances (p. 87)
CutTriangle	Local path length reduction (p. 99)
PlanShort	Length reduction planning (p. 100)
SegmentColl	Collision test for a path segment (p. 109)
SweptObjColl	Collision test for a moving object (p. 114 and 118)
SweptStepColl	Collision test for an intermediate segment (p. 117)
ObjColl	Collision test for two objects (p. 118)
BisectScale	Approximation of scaling (p. 120)
SweptObjGetScale	Scaling of a moving object (p. 121)

CONTENTS

1	Introduction	1
1.1	Motivation	1
1.2	Objectives	2
1.2.1	Sample Robot Applications	2
1.2.2	The Context of Motion Planning	4
1.2.3	Objectives and Requirements	8
1.3	Approach of the BB-Method	10
1.4	Overview of this Work	11
1.5	Summary of Results	13
2	State of Research	15
2.1	Introduction	15
2.2	Theoretical Results	16
2.3	Attempting a Classification	17
2.4	Resolution Complete Global Planning	18
2.5	Random Global Planning	20
2.6	Opportunistic Resolution Complete Global Planning	21
2.7	Opportunistic Random Global Planning	23
2.8	Heuristic Planning	25
2.9	Summary	28
3	Terms, Objectives, and Approach	29
3.1	Basic Terms	29
3.1.1	Geometry Models	29
3.1.2	Homogeneous Transforms	31
3.2	Manipulator Systems	33
3.2.1	Modelling Joints Between Objects	33
3.2.2	Modelling Manipulator Systems	33
3.2.3	Collision of a Manipulator System	35
3.3	Workspace and Configuration Space	36
3.3.1	Workspace	36
3.3.2	Configuration Space	37
3.3.3	Paths	39
3.4	Objectives	39

3.4.1	O1 – Plan Collision-Free Paths	39
3.4.2	O2 – Keep Safety Distances	40
3.4.3	O3 – Prefer Short Paths	41
3.5	Approach – Path Rating and Modification	42
3.6	Illustration Examples	43
4	O1 – Plan a Collision-Free Motion	47
4.1	Rating Function Q_{scale} – Reducing the Robot’s Size	47
4.1.1	Intention and Approach	47
4.1.2	Scaling of Links and Manipulator Systems	49
4.1.3	Rating a Path Segment	50
4.2	Path Modification – Local Planning	52
4.2.1	Intention and Approach	52
4.2.2	Candidate Creation	54
4.2.3	Replacement Selection	59
4.2.4	Path Refinement	61
4.2.5	Planning Algorithm	63
4.2.6	Termination	64
4.3	Planning Examples	64
4.4	Failure of Local Planning	70
4.4.1	“Local Maximum” of the Rating Function	70
4.4.2	“Loss” of Degrees of Freedom	71
4.4.3	Parameter Selection	73
4.5	Global Planning with Random Subgoals	73
4.6	Complexity and Computational Effort	74
5	O2 – Planning Safety Distances	77
5.1	Rating Function Q_{dist} – Expanding the Robot	77
5.1.1	Intention and Approach	77
5.1.2	Rating a Path Segment	78
5.2	Path Modification	80
5.2.1	Intention and Approach	80
5.2.2	Candidate Creation	82
5.2.3	Replacement Selection	82
5.2.4	Path Refinement	83
5.2.5	Planning Algorithm	86
5.2.6	Termination	87
5.3	Planning Examples	88
5.4	Complexity and Computational Effort	93
6	O3 – Planning Short Paths	95
6.1	Rating Function Q_{short}	95
6.2	Path Modification	96

6.2.1	Intention and Approach	96
6.2.2	Candidate Creation	98
6.2.3	Replacement Selection	98
6.2.4	Planning Algorithm	99
6.2.5	Termination	100
6.3	Planning Examples	101
6.4	Complexity and Computational Effort	105
7	Swept Volume Collision Detection and Path Rating	107
7.1	Intention and Approach	107
7.2	Discretizing a Path Segment	110
7.3	Sweeping Polyhedral Objects	114
7.3.1	Principle Algorithm	114
7.3.2	Expanded Geometry Models	114
7.3.3	Collision Detection for Swept Edges	115
7.3.4	Collision Detection for Swept Polyhedral Objects	118
7.4	Realization of the Rating Functions	118
7.4.1	Intention and Approach	118
7.4.2	Realization of Q_{scale}	120
7.4.3	Realization of Q_{dist}	121
7.5	Complexity and Computational Effort	122
7.5.1	Manipulator Collision Detection at a Single Configuration	122
7.5.2	Collision Detection for Path Segments	123
7.5.3	Hierarchical Modelling	125
8	Experimental Results	127
8.1	Practical Aspects	127
8.2	Examples	129
8.2.1	6-DOF Manipulator with Difficult Load	129
8.2.2	7-DOF Redundant Manipulator	134
8.2.3	7-DOF Redundant Manipulator on 3-DOF Gantry (10-DOF)	136
8.2.4	7-DOF Redundant Manipulator on 3-DOF Gantry with 4- Fingered Hand (26-DOF)	138
8.2.5	16-DOF Snake Manipulator	139
8.2.6	31-DOF Snake Manipulator	140
8.2.7	Planning for a Mobile Manipulator	143
8.3	Comparison	146
8.3.1	Problems of Comparison	146
8.3.2	Comparison with <i>RPP</i> – Randomized Path Planner	146
8.3.3	Comparison with parallelized A*-Search in C-Space	149
8.3.4	Comparison with <i>ZZ</i> -Planning with Slidesteps	151

9 Conclusion	153
9.1 Discussion of Results	153
9.1.1 Requirements	153
9.1.2 Objectives	155
9.1.3 Extensions	157
9.2 Suggestions for Future Work	158
References	161
Index	169

LIST OF FIGURES

1.1	Embedding of Motion Planning	5
3.1	Sample Manipulators with 2 and 8 Joints	34
3.2	Workspace and Configuration Space for 2-DOF Manipulator . . .	38
3.3	Sample Task for 2-DOF Manipulator	43
3.4	Sample Task for 6-DOF Manipulator	44
3.5	Sample Task for 16-DOF Manipulator	45
4.1	Scaling Link 2 of the 2-DOF Manipulator	48
4.2	Scaling Examples for Different Robots	50
4.3	Planning for the 2-DOF Manipulator	53
4.4	Planned Collision Free Motion for the 2-DOF Manipulator	53
4.5	Examples for Candidate Calculation	58
4.6	Principle Alternative Paths	59
4.7	Principle Path Refinement	62
4.8	Planning for the 6-DOF Manipulator	65
4.9	Planned Collision Free Motion for the 6-DOF Manipulator	67
4.10	Planning for the 16-DOF Manipulator	68
4.11	Planned Collision Free Motion for the 16-DOF Manipulator	69
4.12	Example for Local Maximum	71
4.13	Failure Situation for 6-DOF PUMA-type Manipulator	72
4.14	Failure Situation for Planar 8-DOF Manipulator	72
5.1	Expanding Link 2 of the 2-DOF Manipulator	78
5.2	Planning Safety Distance for the 2-DOF Manipulator	81
5.3	Planned Motion with Safety Distance for the 2-DOF Manipulator	81
5.4	Excluding Segments from Planning	84
5.5	Planning Safety Distances for the 6-DOF Manipulator	89
5.6	Planned Motion for the 6-DOF Manipulator	90
5.7	Planning Safety Distances for the 16-DOF Manipulator	91
5.8	Planned Motion for the 16-DOF Manipulator	92
6.1	Path Length Reduction for the 2-DOF Manipulator	97
6.2	Planned Motion for the 2-DOF Manipulator	97
6.3	Path Length Reduction for the 6-DOF Manipulator	102
6.4	Planned Motion for the 6-DOF Manipulator	103

6.5	Path Length Reduction and Motion for the 16-DOF Manipulator	104
7.1	Principle Idea of Swept Volume Collision Detection	109
7.2	C-Step Size Conditions	111
7.3	Polyhedral Models Expanded by Vertex Displacement	115
7.4	Convex Hull of Swept Edge	116
7.5	Illustration of Required Test at Intermediate Configuration	117
8.1	Tasks for the 6-DOF Manipulator	130
8.2	7-DOF Redundant Manipulator	135
8.3	7-DOF Redundant Manipulator on 3-DOF Gantry	137
8.4	7-DOF Redundant Manipulator on 3-DOF Gantry with 4-Fingered Hand (26-DOF)	138
8.5	31-DOF Snake Manipulator	141
8.6	Scenery ROMAN1	143
8.7	Scenery ROMAN2	145
8.8	Solutions for Sceneries ROMAN1 and ROMAN2	145
8.9	Scenery HOLE IN THE WALL	147
8.10	Scenery TWO SLOTS	149
8.11	Four Benchmark Sceneries	150

LIST OF TABLES

7.1	Relative Effort of Swept Volume Collision Detection	124
8.1	Results for 6-DOF Manipulator – Task TABLE1	131
8.2	Results for 6-DOF Manipulator – Task TABLE2	132
8.3	Results for 6-DOF Manipulator – Task TABLE3	132
8.4	Results for 6-DOF Manipulator – Random Tasks, Local Planning	133
8.5	Results for 6-DOF Manipulator – Random Tasks, Global Planning	134
8.6	Results for 7-DOF Manipulator – Different Tasks	134
8.7	Results for 10-DOF Manipulator – Different Tasks	136
8.8	Results for 26-DOF Manipulator (10-DOF and 4-Fingered Hand)	139
8.9	Parameter Variations for the 16-DOF Manipulator	139
8.10	Results for 31-DOF Manipulator – Scenery ONE GATE	140
8.11	Results for 31-DOF Manipulator – Scenery THREE GATE	142
8.12	Results for 31-DOF Manipulator – Scenery COLUMNS	142
8.13	Results for Scenery ROMAN1	144
8.14	Results for Scenery ROMAN2	146
8.15	Results for Scenery HOLE IN THE WALL	148
8.16	Results for Scenery TWO SLOTS	148
8.17	Planning Times for Benchmark Sceneries	151
8.18	Comparison with ZZ-Planning	152