

# Preface

## What Is Meant by “Planning Algorithms”?

Due to many exciting developments in the fields of robotics, artificial intelligence, and control theory, three topics that were once quite distinct are presently on a collision course. In robotics, motion planning was originally concerned with problems such as how to move a piano from one room to another in a house without hitting anything. The field has grown, however, to include complications such as uncertainties, multiple bodies, and dynamics. In artificial intelligence, planning originally meant a search for a sequence of logical operators or actions that transform an initial world state into a desired goal state. Presently, planning extends beyond this to include many decision-theoretic ideas such as Markov decision processes, imperfect state information, and game-theoretic equilibria. Although control theory has traditionally been concerned with issues such as stability, feedback, and optimality, there has been a growing interest in designing algorithms that find feasible open-loop trajectories for nonlinear systems. In some of this work, the term “motion planning” has been applied, with a different interpretation from its use in robotics. Thus, even though each originally considered different problems, the fields of robotics, artificial intelligence, and control theory have expanded their scope to share an interesting common ground.

In this text, I use the term *planning* in a broad sense that encompasses this common ground. This does not, however, imply that the term is meant to cover everything important in the fields of robotics, artificial intelligence, and control theory. The presentation focuses on *algorithm* issues relating to planning. Within robotics, the focus is on designing algorithms that generate useful motions by processing complicated geometric models. Within artificial intelligence, the focus is on designing systems that use decision-theoretic models to compute appropriate actions. Within control theory, the focus is on algorithms that compute feasible trajectories for systems, with some additional coverage of feedback and optimality. Analytical techniques, which account for the majority of control theory literature, are not the main focus here.

The phrase “planning and control” is often used to identify complementary issues in developing a system. Planning is often considered as a higher level process than control. In this text, I make no such distinctions. Ignoring historical connotations that come with the terms, “planning” and “control” can be used

interchangeably. Either refers to some kind of decision making in this text, with no associated notion of “high” or “low” level. A hierarchical approach can be developed, and either level could be called “planning” or “control” without any difference in meaning.

### **Who Is the Intended Audience?**

The text is written primarily for computer science and engineering students at the advanced-undergraduate or beginning-graduate level. It is also intended as an introduction to recent techniques for researchers and developers in robotics, artificial intelligence, and control theory. It is expected that the presentation here would be of interest to those working in other areas such as computational biology (drug design, protein folding), virtual prototyping, manufacturing, video game development, and computer graphics. Furthermore, this book is intended for those working in industry who want to design and implement planning approaches to solve their problems.

I have attempted to make the book as self-contained and readable as possible. Advanced mathematical concepts (beyond concepts typically learned by undergraduates in computer science and engineering) are introduced and explained. For readers with deeper mathematical interests, directions for further study are given.

### **Where Does This Book Fit?**

Here is where this book fits with respect to other well-known subjects:

**Robotics:** This book addresses the planning part of robotics, which includes motion planning, trajectory planning, and planning under uncertainty. This is only one part of the big picture in robotics, which includes issues not directly covered here, such as mechanism design, dynamical system modeling, feedback control, sensor design, computer vision, inverse kinematics, and humanoid robotics.

**Artificial Intelligence:** Machine learning is currently one of the largest and most successful divisions of artificial intelligence. This book (perhaps along with [382]) represents the important complement to machine learning, which can be thought of as “machine planning.” Subjects such as reinforcement learning and decision theory lie in the boundary between the two and are covered in this book. Once learning is being successfully performed, what decisions should be made? This enters into planning.

**Control Theory:** Historically, control theory has addressed what may be considered here as planning in continuous spaces under differential constraints. Dynamics, optimality, and feedback have been paramount in control theory. This book is complementary in that most of the focus is on open-loop control laws, feasibility as opposed to optimality, and dynamics may or may not be important.

Nevertheless, feedback, optimality, and dynamics concepts appear in many places throughout the book. However, the techniques in this book are mostly algorithmic, as opposed to the analytical techniques that are typically developed in control theory.

**Computer Graphics:** Animation has been a hot area in computer graphics in recent years. Many techniques in this book have either been applied or can be applied to animate video game characters, virtual humans, or mechanical systems. Planning algorithms allow users to specify tasks at a high level, which avoids having to perform tedious specifications of low-level motions (e.g., key framing).

**Algorithms:** As the title suggests, this book may fit under algorithms, which is a discipline within computer science. Throughout the book, typical issues from combinatorics and complexity arise. In some places, techniques from computational geometry and computational real algebraic geometry, which are also divisions of algorithms, become important. On the other hand, this is not a pure algorithms book in that much of the material is concerned with characterizing various decision processes that arise in applications. This book does not focus purely on complexity and combinatorics.

**Other Fields:** At the periphery, many other fields are touched by planning algorithms. For example, motion planning algorithms, which form a major part of this book, have had a substantial impact on such diverse fields as computational biology, virtual prototyping in manufacturing, architectural design, aerospace engineering, and computational geography.

## Suggested Use

The ideas should flow naturally from chapter to chapter, but at the same time, the text has been designed to make it easy to skip chapters. The dependencies between the four main parts are illustrated in Figure 1.

If you are only interested in robot motion planning, it is only necessary to read Chapters 3–8, possibly with the inclusion of some discrete planning algorithms from Chapter 2 because they arise in motion planning. Chapters 3 and 4 provide the foundations needed to understand basic robot motion planning. Chapters 5 and 6 present algorithmic techniques to solve this problem. Chapters 7 and 8 consider extensions of the basic problem. If you are additionally interested in nonholonomic planning and other problems that involve differential constraints, then it is safe to jump ahead to Chapters 13–15, after completing Part II.

Chapters 11 and 12 cover problems in which there is sensing uncertainty. These problems live in an *information space*, which is detailed in Chapter 11. Chapter 12 covers algorithms that plan in the information space.

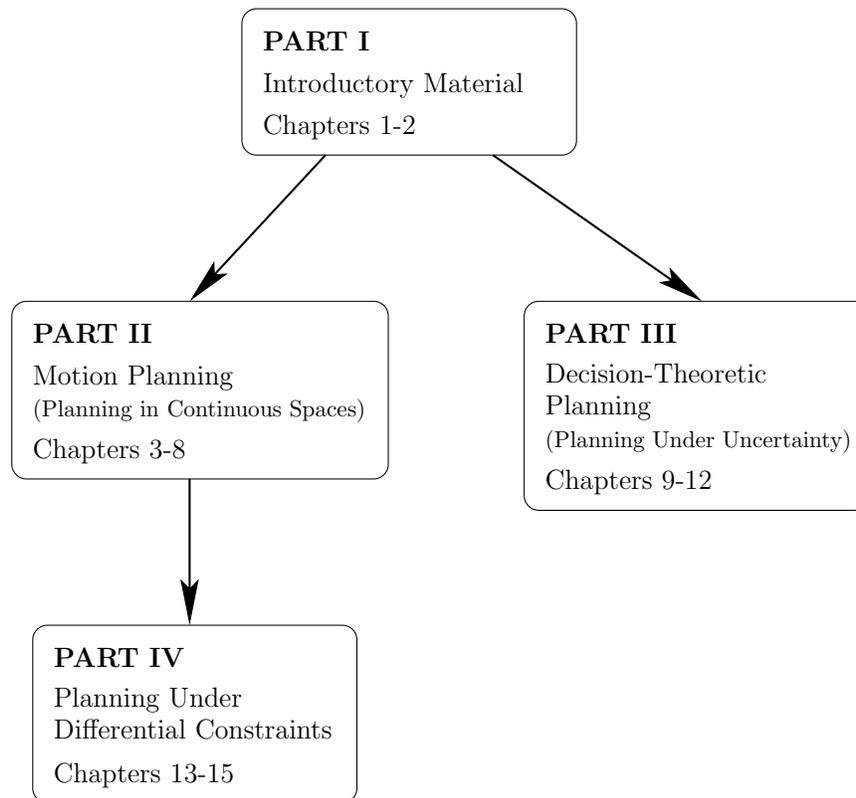


Figure 1: The dependencies between the four main parts of the book.

If you are interested mainly in decision-theoretic planning, then you can read Chapter 2 and then jump straight to Chapters 9–12. The material in these later chapters does not depend much on Chapters 3–8, which cover motion planning. Thus, if you are not interested in motion planning, the chapters may be easily skipped.

There are many ways to design a semester or quarter course from the book material. Figure 2 may help in deciding between core material and some optional topics. For an advanced undergraduate-level course, I recommend covering one core and some optional topics. For a graduate-level course, it may be possible to cover a couple of cores and some optional topics, depending on the initial background of the students. A two-semester sequence can also be developed by drawing material from all three cores and including some optional topics. Also, two independent courses can be made in a number of different ways. If you want to avoid continuous spaces, a course on discrete planning can be offered from Sections 2.1–2.5, 9.1–9.5, 10.1–10.5, 11.1–11.3, 11.7, and 12.1–12.3. If you are interested in teaching some game theory, there is roughly a chapter’s worth of material in Sections 9.3–9.4, 10.5, 11.7, and 13.5. Material that contains the most prospects for future research appears in Chapters 7, 8, 11, 12, and 14. In particular, research on information spaces is still in its infancy.

**Motion planning**

Core: 2.1-2.2, 3.1-3.3, 4.1-4.3, 5.1-5.6, 6.1-6.3

Optional: 3.4-3.5, 4.4, 6.4-6.5, 7.1-7.7, 8.1-8.5

**Planning under uncertainty**

Core: 2.1-2.3, 9.1-9.2, 10.1-10.4, 11.1-11.6, 12.1-12.3

Optional: 9.3-9.5, 10.5-10.6, 11.7, 12.4-12.5

**Planning under differential constraints**

Core: 8.3, 13.1-13.3, 14.1-14.4, 15.1, 15.3-15.4

Optional: 13.4-13.5, 14.5-14.7, 15.2, 15.5

Figure 2: Based on Parts II, III, and IV, there are three themes of core material and optional topics.

To facilitate teaching, there are more than 500 examples and exercises throughout the book. The exercises in each chapter are divided into written problems and implementation projects. For motion planning projects, students often become bogged down with low-level implementation details. One possibility is to use the Motion Strategy Library (MSL):

<http://msl.cs.uiuc.edu/msl/>

as an object-oriented software base on which to develop projects. I have had great success with this for both graduate and undergraduate students.

For additional material, updates, and errata, see the Web page associated with this book:

<http://planning.cs.uiuc.edu/>

You may also download a free electronic copy of this book for your own personal use.

For further reading, consult the numerous references given at the end of chapters and throughout the text. Most can be found with a quick search of the Internet, but I did not give too many locations because these tend to be unstable over time. Unfortunately, the literature surveys are shorter than I had originally planned; thus, in some places, only a list of papers is given, which is often incomplete. I have tried to make the survey of material in this book as impartial as possible, but there is undoubtedly a bias in some places toward my own work. This was difficult to avoid because my research efforts have been closely intertwined with the development of this book.

**Acknowledgments**

I am very grateful to many students and colleagues who have given me extensive feedback and advice in developing this text. It evolved over many years through the development and teaching of courses at Stanford, Iowa State, and the University of Illinois. These universities have been very supportive of my efforts.

Many ideas and explanations throughout the book were inspired through numerous collaborations. For this reason, I am particularly grateful to the helpful insights and discussions that arose through collaborations with Michael Branicky, Francesco Bullo, Jeff Erickson, Emilio Frazzoli, Rob Ghrist, Leo Guibas, Seth Hutchinson, Lydia Kavraki, James Kuffner, Jean-Claude Latombe, Rajeev Motwani, Rafael Murrieta, Rajeev Sharma, Thierry Siméon, and Giora Slutzki. Over years of interaction, their ideas helped me to shape the perspective and presentation throughout the book.

Many valuable insights and observations were gained through collaborations with students, especially Peng Cheng, Hamid Chitsaz, Prashanth Konkimalla, Jason O’Kane, Steve Lindemann, Stjepan Rajko, Shai Sachs, Boris Simov, Benjamin Tovar, Jeff Yakey, Libo Yang, and Anna Yershova. I am grateful for the opportunities to work with them and appreciate their interaction as it helped to develop my own understanding and perspective.

While writing the text, at many times I recalled being strongly influenced by one or more technical discussions with colleagues. Undoubtedly, the following list is incomplete, but, nevertheless, I would like to thank the following colleagues for their helpful insights and stimulating discussions: Pankaj Agarwal, Srinivas Akella, Nancy Amato, Devin Balkcom, Tamer Başar, Antonio Bicchi, Robert Bohlin, Joel Burdick, Stefano Carpin, Howie Choset, Juan Cortés, Jerry Dejong, Bruce Donald, Ignacy Duleba, Mike Erdmann, Roland Geraerts, Malik Ghallab, Ken Goldberg, Pekka Isto, Vijay Kumar, Andrew Ladd, Jean-Paul Laumond, Kevin Lynch, Matt Mason, Pascal Morin, David Mount, Dana Nau, Jean Ponce, Mark Overmars, Elon Rimon, and Al Rizzi.

Many thanks go to Karl Bohringer, Marco Bressan, John Cassel, Stefano Carpin, Peng Cheng, Hamid Chitsaz, Ignacy Duleba, Claudia Esteves, Brian Gerkey, Ken Goldberg, Björn Hein, Sanjit Jhala, Marcelo Kallmann, James Kuffner, Olivier Lefebvre, Mong Leng, Steve Lindemann, Dennis Nieuwenhuisen, Jason O’Kane, Neil Petroff, Mihail Pivtoraiko, Stephane Redon, Gildardo Sanchez, Viktor Schmidt, Fabian Schöfeld, Robin Schubert, Sanketh Shetty, Mohan Sirchabesan, James Solberg, Domenico Spensieri, Kristian Spoerer, Tony Stentz, Morten Strandberg, Ichiro Suzuki, Benjamin Tovar, Zbynek Winkler, Anna Yershova, Jingjin Yu, George Zaines, and Liangjun Zhang for pointing out numerous mistakes in the on-line manuscript. I also appreciate the efforts of graduate students in my courses who scribed class notes that served as an early draft for some parts. These include students at Iowa State and the University of Illinois: Peng Cheng, Brian George, Shamsi Tamara Iqbal, Xiaolei Li, Steve Lindemann, Shai Sachs, Warren Shen, Rishi Talreja, Sherwin Tam, and Benjamin Tovar.

I sincerely thank Krzysztof Kozłowski and his staff, Joanna Gawecka, Wirginia Król, and Marek Lawniczak, at the Politechnika Poznańska (Technical University of Poznan) for all of their help and hospitality during my sabbatical in Poland. I also thank Heather Hall for managing my U.S.-based professional life while I lived in Europe. I am grateful to the National Science Foundation, the Office of

Naval Research, and DARPA for research grants that helped to support some of my sabbatical and summer time during the writing of this book. The Department of Computer Science at the University of Illinois was also very generous in its support of this huge effort.

I am very fortunate to have artistically talented friends. I am deeply indebted to James Kuffner for creating the image on the front cover and to Audrey de Malmazet de Saint Andeol for creating the art on the first page of each of the four main parts.

Finally, I thank my editor, Lauren Cowles, my copy editor, Elise Oranges, and the rest of the people involved with Cambridge University Press for their efforts and advice in preparing the manuscript for publication.

Steve LaValle  
Urbana, Illinois, U.S.A.