

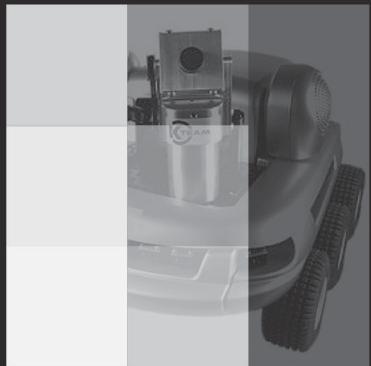
Sebastian Thrun, Wolfram Burgard, and Dieter Fox

# [確率ロボティクス]

セバスチャン・スラン、ウルラム・バーガード、ディーター・フォックス 著 上田隆一 訳

本書は、学生や研究者、エンジニアなど、ロボットの実装を行う全ての人に向けて書かれた学術書です。ベイズ則とその拡張を数学的背景に、『確率ロボティクス』の主要なアルゴリズムを、擬似コードによる実装例を交えて詳細に解説・議論しています。物理環境やロボットシステム自身に存在する『不確実さ』に対処し、ロバストなロボットシステムを実現するためのステップとなる、大変有用な一冊です。





# 確率ロボティクス

Sebastian Thrun, Wolfram Burgard, and Dieter Fox

訳：上田隆一

## Probabilistic robotics

ISBN0262201623 by Sebastian Thrun , Wolfram Burgard , Dieter Fox

©2006 Massachusetts Institute of Technology

Japanese-language Edition copyright © 2007 , 2015 Mynavi Corporation.

All rights reserved.

Japanese translation right with Massachusetts Institute of Technology

through The English Agency (Japan Ltd.).

●本書は

『確率ロボティクス』(2007年10月 弊社刊)

を元にしたオンドマンド版/電子版です。

内容は変更されておりません。

●各章の扉画像は、下記に掲載する機関のご厚意により提供いただいております。

本扉 : Koaka / K-Team (Compliments of K-Team)

目次 : Albert / Autonomous Intelligent Systems Lab. at the University of Freiburg (Compliments of Prof. Burgard)

まえがき : MINERVA / the Stanford AI Lab. (Compliments of Prof. Thrun)

1章 : Koaka / K-Team (Compliments of K-Team)

2章 : Ameria / the Stanford AI Lab. (Compliments of Prof. Thrun)

3章 : PatrolBot / MobileRobots Inc. (Compliments of MobileRobots Inc.)

4章 : Albert / Autonomous Intelligent Systems Lab. at the University of Freiburg (Compliments of Prof. Burgard)

5章 : groundhog robot / the Stanford AI Lab. (Compliments of Prof. Thrun)

6章 : Seeker / MobileRobots Inc. (Compliments of MobileRobots Inc.)

7章 : Seeker / MobileRobots Inc. (Compliments of MobileRobots Inc.)

8章 : 2D+3D SLAM/VSLAM & Perception System / MobileRobots Inc. (Compliments of MobileRobots Inc.)

9章 : LMS200 / SICK,Inc. (Compliments of SICK,Inc)

10章 : Seeker / MobileRobots Inc. (Compliments of MobileRobots Inc.)

11章 : Seeker / MobileRobots Inc. (Compliments of MobileRobots Inc.)

12章 : PIONEER P3-DX / MobileRobots Inc. (Compliments of MobileRobots Inc.)

13章 : Cye / Educational Robot company (Compliments of Educational Robot company)

14章 : Zora / Autonomous Intelligent Systems Lab. at the University of Freiburg (Compliments of Prof. Burgard)

15章 : Hero / the Stanford AI Lab. (Compliments of Prof. Thrun)

16章 : Cye / Educational Robot company (Compliments of Educational Robot company)

17章 : MINERVA / the Stanford AI Lab. (Compliments of Prof. Thrun)

あとがきにかえて : MINERVA / the Stanford AI Lab. (Compliments of Prof. Thrun)

参考文献 : Seeker / MobileRobots Inc. (Compliments of MobileRobots Inc.)

索引 : MobileRobots Inc. (Compliments of MobileRobots Inc.)

- 本書に記載された内容は、情報の提供のみを目的としております。したがって、本書を用いての運用はすべてお客様自身の責任と判断において行ってください。

- 著者や出版社のいずれも、本書の内容に関してなんらかの保証をするものではなく、内容に関するいかなる運用結果についてもいっさいの責任を負いません。あらかじめご了承ください。

- 本書中の会社名や商品名は、該当する各社の商標または登録商標です。



## もくじ

## Contents

謝辞	XV
<b>第Ⅰ部 基礎</b>	001
<b>1 緒言</b>	003
1.1 ロボティクスにおける不確実さ	003
1.2 確率ロボティクス	004
1.3 他の枠組みとの関係	009
1.4 本書の構成	010
1.5 確率ロボティクスの講義	010
1.6 関連する文献	011
<b>2 再帰的状態推定</b>	013
2.1 はじめに	013
2.2 確率の基本的な概念	014
2.3 ロボットと環境の相互作用	018
2.3.1 状態	018
2.3.2 環境との相互作用	021
2.3.3 確率的発生法則	022
2.3.4 信念分布	024
2.4 ベイズフィルタ	025
2.4.1 ベイズフィルタアルゴリズム	025
2.4.2 例	026
2.4.3 ベイズフィルタの数学的導出	029

2.4.4 マルコフ性	031
2.5 表現と計算	031
2.6 まとめ	032
2.7 関連する文献	033
2.8 練習問題	033
<b>3 ガウシアンフィルタ</b>	<b>037</b>
3.1 はじめに	037
3.2 カルマンフィルタ	038
3.2.1 線形ガウス型モデル	038
3.2.2 カルマンフィルタアルゴリズム	040
3.2.3 例	041
3.2.4 カルマンフィルタ (KF) の数学的導出	042
3.3 拡張カルマンフィルタ	050
3.3.1 なぜ線形化するのか？	050
3.3.2 テイラー展開による線形化	053
3.3.3 EKF アルゴリズム	055
3.3.4 EKF の数学的導出	055
3.3.5 実用上配慮すべき点	057
3.4 アンセンテッドカルマンフィルタ	060
3.4.1 アンセンテッド変換による線形化	060
3.4.2 UKF アルゴリズム	064
3.5 情報フィルタ	066
3.5.1 カノニカルパラメータ化	066
3.5.2 情報フィルタアルゴリズム	067
3.5.3 情報フィルタの数学的導出	069
3.5.4 拡張情報フィルタアルゴリズム	070
3.5.5 拡張情報フィルタの数学的導出	071
3.5.6 実用に関する考察	072
3.6 まとめ	073
3.7 関連する文献	075
3.8 練習問題	075

<b>4 ノンパラメトリックフィルタ</b>	<b>079</b>
4.1 ヒストグラムフィルタ	080
4.1.1 離散ベイズフィルタアルゴリズム	080
4.1.2 連続状態空間の場合	081
4.1.3 ヒストグラム近似の数学的導出	083
4.1.4 離散化の手法	085
4.2 静的状態に対するバイナリベイズフィルタ	086
4.3 パーティクルフィルタ	089
4.3.1 基本アルゴリズム	089
4.3.2 重み付きサンプリング	092
4.3.3 パーティクルフィルタの数学的導出	094
4.3.4 パーティクルフィルタの実用上の性質と考慮すべき点	096
4.4 まとめ	104
4.5 関連する文献	104
4.6 練習問題	105
<b>5 ロボットの動作</b>	<b>107</b>
5.1 はじめに	107
5.2 事前準備	108
5.2.1 運動学におけるコンフィギュレーション	108
5.2.2 確率的運動学	109
5.3 速度動作モデル	110
5.3.1 解析的な計算	111
5.3.2 サンプリングアルゴリズム	112
5.3.3 速度動作モデルの数学的導出	114
5.4 オドメトリ動作モデル	120
5.4.1 解析的な計算	121
5.4.2 サンプリングアルゴリズム	123
5.4.3 オドメトリ動作モデルの数学的導出	125
5.5 動作と地図	127
5.6 まとめ	130
5.7 関連する文献	131
5.8 練習問題	131

<b>6 ロボットの知覚</b>	135
6.1 はじめに	135
6.2 地図	138
6.3 レンジファインダのビームモデル	139
6.3.1 基本計測アルゴリズム	139
6.3.2 モデル固有のパラメータの調整	143
6.3.3 ビームモデルの数学的導出	147
6.3.4 実用に関する考察	151
6.3.5 ビームモデルの限界	152
6.4 レンジファインダの尤度場	152
6.4.1 基本アルゴリズム	152
6.4.2 拡張	155
6.5 相関ベースの計測モデル	157
6.6 特徴ベースの計測モデル	158
6.6.1 特徴抽出	158
6.6.2 ランドマーク計測	159
6.6.3 対応関係が既知の場合のセンサモデル	160
6.6.4 姿勢のサンプリング	161
6.6.5 さらに考慮すべき点	163
6.7 実用に関する考察	164
6.8 まとめ	165
6.9 関連する文献	166
6.10 練習問題	166
<b>第 II 部 ローカライゼーション</b>	169
<b>7 移動ロボットの位置推定：マルコフ位置推定とガウス分布表現</b>	171
7.1 位置推定問題の分類	173
7.2 マルコフ位置推定	176
7.3 マルコフ位置推定の図解	178
7.4 EKF 位置推定	180
7.4.1 図解	180
7.4.2 EKF 位置推定アルゴリズム	181

7.4.3 EKF 位置推定の数学的導出	182
7.4.4 実環境を想定した実装	188
7.5 対応関係の推定	193
7.5.1 対応関係が未知の場合の EKF 位置推定	193
7.5.2 最尤データ対応付け手法の数学的導出	193
7.6 複数仮説追従	195
7.7 UKF 位置推定	197
7.7.1 UKF 位置推定の数学的導出	197
7.7.2 図解	200
7.8 実用に関する考察	204
7.9 まとめ	207
7.10 関連する文献	208
7.11 練習問題	209
<b>8 移動ロボットの位置推定：格子表現とモンテカルロ法による表現</b>	<b>213</b>
8.1 はじめに	213
8.2 格子位置推定	214
8.2.1 基本アルゴリズム	214
8.2.2 格子の解像度	216
8.2.3 計算量の考慮	218
8.2.4 図解	220
8.3 モンテカルロ位置推定	225
8.3.1 図解	225
8.3.2 MCL アルゴリズム	225
8.3.3 実世界を想定した実装	227
8.3.4 MCL の性質	228
8.3.5 ランダムパーティクル MCL：誤りからの回復	229
8.3.6 提案分布の改良	234
8.3.7 KLD サンプリング：適応的なパーティクル数の調節	236
8.4 動的環境における位置推定	240
8.5 実用に関する考察	245
8.6 まとめ	246
8.7 関連する文献	246
8.8 練習問題	248

<b>第 III 部 地図生成</b>	251
<b>9 占有格子地図の生成</b>	253
9.1 はじめに	253
9.2 占有格子地図の生成アルゴリズム	256
9.2.1 マルチセンサフュージョン	263
9.3 逆計測モデルの学習	264
9.3.1 計測モデルの逆モデル化	264
9.3.2 順モデルからのサンプリング	265
9.3.3 誤差関数	266
9.3.4 例, そして考慮すべき点	268
9.4 最大事後確率 (MAP) 推定を用いた占有地図の作成	269
9.4.1 独立性の仮定が問題になる場合	269
9.4.2 順モデルを利用した占有格子地図の生成	270
9.5 まとめ	273
9.6 関連する文献	274
9.7 練習問題	275
<b>10 姿勢と地図の同時推定</b>	279
10.1 はじめに	279
10.2 拡張カルマンフィルタによる SLAM	282
10.2.1 準備と仮定	282
10.2.2 対応関係が既知の場合の SLAM	282
10.2.3 EKF SLAM の数学的導出	286
10.3 対応関係が未知の場合の EKF SLAM	289
10.3.1 一般 EKF SLAM アルゴリズム	289
10.3.2 例	292
10.3.3 特徴の選択方法と地図の管理	296
10.4 まとめ	298
10.5 関連する文献	299
10.6 練習問題	302

<b>11</b>	<b>GraphSLAM アルゴリズム</b>	<b>305</b>
11.1	はじめに	305
11.2	直観的な説明	307
11.2.1	グラフの構築	308
11.2.2	推定	310
11.3	GraphSLAM アルゴリズム	313
11.4	GraphSLAM の数学的導出	318
11.4.1	完全 SLAM の事後推定	319
11.4.2	負の対数事後確率	320
11.4.3	テイラー展開	321
11.4.4	情報形式の構築	322
11.4.5	情報形式の縮小	322
11.4.6	軌跡と地図の復元	325
11.5	GraphSLAM におけるデータ対応付け	327
11.5.1	対応関係が未知の場合の GraphSLAM アルゴリズム	328
11.5.2	対応関係のテストの数学的導出	330
11.6	計算効率の向上	332
11.7	実世界での実装	333
11.8	他の最適化手法	338
11.9	まとめ	341
11.10	関連する文献	343
11.11	練習問題	344
<b>12</b>	<b>疎拡張情報フィルタ</b>	<b>347</b>
12.1	はじめに	347
12.2	直観的な理解	349
12.3	SEIF SLAM アルゴリズム	352
12.4	SEIF の数学的導出	356
12.4.1	動作更新	356
12.4.2	計測更新	359
12.5	疎化	359
12.5.1	概念	359
12.5.2	SEIF における疎化	360

12.5.3 精化ステップの数学的導出	361
12.6 地図の近似復元のための分割計算	362
12.7 SEIF はどれだけ疎であるべきか？	364
12.8 データの逐次対応付け	367
12.8.1 逐次的アプローチにおけるデータ対応付けの確率計算	367
12.8.2 実用に関する考察	369
12.9 分枝限定法によるデータの対応付け	372
12.9.1 再帰的探索	373
12.9.2 任意のデータ対応付けに対する確率計算	375
12.9.3 同値制約	377
12.10 実用に関する考察	378
12.11 複数ロボットによる SLAM	381
12.11.1 地図の融合	381
12.11.2 地図統合の数学的導出	384
12.11.3 対応関係の確立	385
12.11.4 例	386
12.12 まとめ	388
12.13 関連する文献	390
12.14 練習問題	391
<b>13 FastSLAM アルゴリズム</b>	<b>393</b>
13.1 基本アルゴリズム	395
13.2 事後推定の因子分解	396
13.2.1 因子分解された事後推定の数学的導出	397
13.3 対応関係既知の FastSLAM	399
13.4 提案分布の改善	403
13.4.1 新しい姿勢のサンプリングによる推定軌跡の延長	405
13.4.2 観測された特徴の推定値の更新	407
13.4.3 重みの計算	408
13.5 対応関係が未知の場合	410
13.6 地図の管理	412
13.7 FastSLAM アルゴリズム	413
13.8 高効率な実装方法	413
13.9 特徴ベースの地図を作成するための FastSLAM	420

13.9.1 実験例と考察	420
13.9.2 ループの綴じ込み	422
13.10 格子ベース FastSLAM	426
13.10.1 アルゴリズム	426
13.10.2 実験例と考察	427
13.11 まとめ	429
13.12 関連する文献	431
13.13 練習問題	432
<b>第 IV 部 計画と制御</b>	<b>435</b>
<b>14 マルコフ決定過程</b>	<b>437</b>
14.1 動機付け	437
14.2 行動選択における不確かさ	440
14.3 値値反復	444
14.3.1 終端状態と報酬	444
14.3.2 完全観測可能な場合における最適制御方策の発見	448
14.3.3 値値関数の計算	449
14.4 ロボット制御への応用	451
14.5 まとめ	455
14.6 関連する文献	456
14.7 練習問題	458
<b>15 部分観測マルコフ決定過程</b>	<b>461</b>
15.1 動機付け	461
15.2 例示	463
15.2.1 設定	463
15.2.2 行動選択	464
15.2.3 センシング	466
15.2.4 予測	470
15.2.5 長い区間での計画と枝刈り	472
15.3 有限 POMDP アルゴリズム	475
15.4 POMDP の数学的導出	477
15.4.1 信念空間における値値反復	477

15.4.2 値値関数の表現	479
15.4.3 値値関数の計算	479
15.5 実用上考慮すべき点	482
15.6 まとめ	486
15.7 関連する文献	487
15.8 練習問題	489
<b>16 POMDP の近似手法</b>	<b>491</b>
16.1 動機付け	491
16.2 QMDP	492
16.3 拡張マルコフ決定過程	493
16.3.1 拡張状態空間	493
16.3.2 AMDP アルゴリズム	494
16.3.3 AMDP の数学的導出	497
16.3.4 移動ロボットのナビゲーションへの適用	498
16.4 モンテカルロ POMDP	501
16.4.1 パーティクルセットの使用	501
16.4.2 MC-POMDP アルゴリズム	502
16.4.3 MC-POMDP の数学的導出	504
16.4.4 実用上考慮すべき点	505
16.5 まとめ	506
16.6 関連する文献	507
16.7 練習問題	507
<b>17 探査</b>	<b>511</b>
17.1 はじめに	511
17.2 基本探査アルゴリズム	513
17.2.1 情報利得	513
17.2.2 グリーディー手法	514
17.2.3 モンテカルロ探査	515
17.2.4 多段階手法	516
17.3 能動的位置推定	516
17.4 占有格子地図を学習するための探査	521
17.4.1 情報利得の計算	521

17.4.2 利得の伝播	524
17.4.3 複数ロボットへの拡張	527
17.5 SLAM のための探査	532
17.5.1 SLAM におけるエントロピー分解	532
17.5.2 FastSLAM における探査	533
17.5.3 実地での性質	536
17.6 まとめ	538
17.7 関連する文献	540
17.8 練習問題	542
あとがきにかえて	545
参考文献	547
インデックス	572





## まえがき

### Preface

この本は、確率ロボティクスという新しい研究分野を紹介するために執筆されました。確率ロボティクスは知覚と制御に関するロボティクスの一分野であり、情報の記述や行動決定のために確率・統計を駆使するところに特徴があります。ロボット技術を利用する際に避けることのできない「不確実性」の問題に、そのようなアプローチで対処することが確率ロボティクスの目指すべきこととなります。確率的な手法を応用したアルゴリズム設計は、近年、ロボティクスにおける有力な枠組みの一つとなりました。本書は、この分野における主要な手法を包括的に紹介するはじめての学術書となります。

本書では、アルゴリズムの説明に特に重点が置かれています。この本のすべてのアルゴリズムはただ一つの基礎的な理論：ベイズ則と、それを時系列データの利用のために拡張したベイズフィルタ、に基づいています。このように数学的な背景が同じであることは、本書の確率的アルゴリズムの重要な共通点です。

私たちは本書において、手法の詳細をなるべく完璧に記述しようと試みました。各章には、一つ、あるいはいくつかの主要なアルゴリズムが記述されています。各アルゴリズムの説明は、次の四つの事項から構成されています。

- (1) 疑似コードによる実装例の記述。
- (2) 公理からの数式展開。これはアルゴリズム実装の裏にある様々な前提を明確に説明するために必要です。
- (3) なるべく多くの実験結果。これらは読者がアルゴリズムをより詳しく理解するために役立つことでしょう。
- (4) 各アルゴリズムに対する、実装する人の立場からの利点、欠点の細かい議論。

上記全ての事項を各アルゴリズムに対して記述することは、実に骨の折れる作業でした。そ

してその作業の結果、斜め読みをしようとする読者には、時々難しく感じられる内容になつたかもしれません（もちろん、数式展開を読み飛ばすことはできますが！）。私たちは熱心な読者が、本書からこの分野に関して深い理解を得られるように願っています。うわべだけで数学的ではない、どのような解説からよりもずっと—。

本書は、私たちや私たちの学生、そしてこの分野の多くの研究者たちの、10年以上にわたる研究成果です。私たちは本書を書き始めたのは1999年のことでした。その時は数か月以内に執筆が終わるくらいに考えていました。そして5年が経ってしまいました、初期の下書きがほとんど残らないくらい、内容が変わってしまいました。その間、執筆を通じて思っていたよりもずっと多くの情報や行動決定に対する理論を学ぶことができました。私たちが学んだ多くのことをこの本に反映しあ伝えできることを、大変幸せに感じています。

この学術書は、学生、研究者、その他ロボットの実装を行う人々に向けて書かれています。私たちは、ロボットの製作者はソフトウェア開発を必ず行うものと信じていますので、本書の内容が、すべてのロボット研究者に興味のあるものであると考えています。また、ロボティクス以外の分野でも、応用統計学者や実世界のセンサを扱う人々に興味を持っていただけるものと思っています。異なる技術的背景を持つ様々な読者を対象とするために、本書の内容はなるべく本書のみで理解できるように記述することを心がけました。線形代数や基本的な確率統計の知識は読み進める上で有用だとは思いますが、本書では確率の基本的な法則についての解説を行い、また高度な数学の使用を避けるようにしています。

また、本書は講義でも利用できるように書かれています。各章ではいくつかの設問や参加型の演習問題を用意しています。講義で使用するときには、各章を1、2回分の講義で利用すると良いでしょう。各章を順番に講義する必要はなく、適宜飛ばしたり入れ替えたりすることもできます。実際、私たちが講義で本書を使用する際は、たいてい7章から始めています。本書での学習と共に、各章おわりで紹介されている演習を参考にして実践的で参加型の実験を行うことをお勧めしたいと思います。自分で手を動かす以上にロボティクスで重要なことはありません！

最善は尽くしましたが、本書の内容にはまだ技術的な不備があるかと思います。そこで、私たちは本書に関するサイト

[www.probabilistic-robotics.org](http://www.probabilistic-robotics.org)

を立ち上げ、訂正を掲載しています。このサイトからは本書の関連情報も得ることができます。この本を皆さんに楽しんでもらえることを願っています。

Sebastian Thrun  
Wolfram Burgard  
Dieter Fox

## 謝辞

数多くの友人、家族、学生、そしてこの分野の研究者の皆様の助けや支えがなければ、本書は完成しませんでした。ここに名前が書ききれないくらいの方々にお世話になりました。

この本の多くの内容は、私たちの過去や現在の学生やポスドクとの共同作業の成果です。特に、研究室に在籍したすべてのメンバーと共に、Rahul Biswas, Matthew Deans, Frank Dellaert, James Diebel, Brian Gerkey, Dirk Hähnel, Johnathan Ko, Cody Kwok, John Langford, Lin Liao, David Lieb, Benson Limketkai, Michael Littman, Yufeng Liu, Andrew Lookingbill, Dimitris Margaritis, Michael Montemerlo, Mark Moors, Mark Paskin, Joelle Pineau, Charles Rosenberg, Nicholas Roy, Aaron Shon, Jamie Schulte, Dirk Schulz, David Stavens, Cyrill Stachniss, そして Chieh-Chih Wang に感謝を述べたいと思います。Greg Armstrong, Grinnell More, Tyson Sawyer, そして Walter Steiner は研究室のロボットが継続して使用できるように、数年にわたってメンテナンスを続けてくれました。

私たちの研究の大部分は、私たちがピッツバーグ州のカーネギーメロン大学に在籍していましたときに行われました。CMU 時代の同僚や友人と刺激的な議論を多くできたことに感謝いたします。また、私たち 3 人と一緒に研究グループに招いてくださいり、すばらしい共同作業のきっかけを与えてくださったボン大学の Armin Cremers に感謝の意を表したいと思います。

研究を進めていく間、私たちは実に多くの研究者から有益な意見や明察を受けました。特に Gary Bradski, Howie Choset, Henrik Christensen, Hugh Durrant-Whyte, Nando de Freitas, Zoubin Gharamani, Geoffery Gordon, Steffen Gutmann, Andrew Howards, Leslie Kaelbling, Daphne Koller, Kurt Konolige, Ben Kuipers, John Leonard, Tom Mitchell, Kevin Murphy, Eduardo Nebot, Paul Newman, Andrew Y.Ng, Reid Simmons, Satinder Singh, Gaurav Sukhatme, Juan Tardós, Ben Wegbreit, そして Alex Zelinsky には、長年のフィードバックをいただいたことに感謝を申し上げます。

Anita Araneda, Gal Elidan, Udo Frese, John Leonard, Benson Limketkai, Rudolph van der Merwe, Anna Petrovskaya, Bob Wang, そして Stefan Williams といった方々には、この本の初期の下書きに対して広範なコメントをいただきました。それらは実に有用でした。Chris Manning は親切にも、この本のために LaTeX のマクロをくださいました。Bob Prior はこの本を出版する手助けをしてくださいました。

本書の出版は、多くの機関や企業からの専門的なアドバイス・惜しみのない資金援助によって達成することができました。特に私たちは、多くの研究プログラム (TMR, MARS, LAGR, SDR, MICA, そして CoABS) を通じての DARPA の援助に感謝しております。また CAREER, ITR, 様々な CISE 助成プログラムを通じての国立科学財団からの援助、ドイツ研究協会や欧州委員会からの助成にも感謝しています。数多くの企業スポンサーと個人か

らの寄付もいただきました。Android, Bosch, Daimler Chryslter, Intel, Google, Microsoft, Mohr Davidow Ventures, Samsung, そして Volkswagen of America からの寄付をいただきました。John Blitch, Doug Gage, Sharon Heise, James Hendler, Larry Jackel, Alex Krott, Wendell Sykes, そして Ed van Reuth に対しては、数年にわたり豊富な課題や手引きをいただきましたことに特に感謝したいと思います。当然のことながら、本書に含まれる知見や結論は著者に帰属するものであります。必ずしもそれらの知見や結論が、あるスポンサーの方針であるとか、あるスポンサーが賛成していることであるとかというように解釈されるべきではありません。

最後に、私たちが最も感謝すべきは私たちの家族です。彼らの愛や献身が、この大きなプロジェクトの完成へと私たちを導いてくれました。特に、Petra Dierkes-Thrun, Anja Gross-Burgard と Carsten Burgard, そして Luz, Sofia, Karla Fox の愛とサポートに感謝いたします。

確率ロボティクス 第Ⅰ部

# 基礎





# 1 緒言

## Introduction

### 1.1 ロボティクスにおける不確実さ

ロボティクスは、コンピュータで制御された機器を通じて、物理世界を知覚し操作することを扱う科学分野である。惑星探査のための移動台車、生産ラインでの産業用ロボットアーム、自動走行する自動車、手術に用いられるマニピュレータなどは、成功しているロボットシステムの例であろう。このようなロボットシステムは物理世界の中にあり、センサを通じて環境の情報を知覚し、そして物理的な力を介してなんらかの操作を行うという性質を持っている。

ロボティクスの多くの部分は未だ初期の段階にあるが、「知的な」操作機器という着想は世の中を大きく変える可能性を秘めている。もしすべての自動車が安全に自動走行するようになり、交通事故が過去のものになってしまえば、それは最高なことではないだろうか？もしチェルノブイリのような核汚染地帯でロボットが無人で清掃をしてくれると、それはすばらしいことではないだろうか？我が家にロボットが住み、家の補修や管理をすべて引き受けてくれると、あなたは最高だと思わないだろうか？

上記のような仕事をする場合、ロボットは物理世界に存在する非常に多くの不確実さに対処しなければならない。ロボットに関する不確実さには、実に数多くの要因が関係している。

何よりもまず、**ロボット環境**<sup>1.1</sup> は本質的に予測不可能である。組立ラインのように良く構成された環境では不確実さが小さいのに対し、ハイウェイや個人の家のような環境はよく物が動き、いろいろな意味で予測不可能である。人間に近いところで作業をするロボットにとって、そのような不確実さは特に大きい。

---

►1.1 robot environment

**センサ**は何でも感知できるわけではなく、いくつかの要因から感知できるものに制限がある。センサの有効範囲や分解能は物理的限界に支配されている。例えば、カメラでは壁の向こう側を見ることはできないし、得られる画像の空間分解能には限界がある。センサは、他に雑音の影響も被る。雑音はセンサの計測値に予測不能な誤差を与えるため、センサから得られる情報の精度を制限してしまう。そして、センサは壊れる場合がある。センサの欠陥を検知することは、実は極めて難しい問題である。

**ロボットの動き**は、モータによって実現されるものであるが、モータには少なくともある程度、予測不可能な要素がある。例えば制御ノイズや消耗、故障のような要因から不確実さが生じる。高負荷の産業用ロボットアームは極めて正確で信頼性のある動きをするが、その他のロボット、例えば安価な移動ロボットのようなものの動きは極めて雑である。

ロボットに実装されたソフトウェアも、時にある種の不確実さをもたらす。ソフトウェア中で実世界をモデル化した**内部モデル**は、すべてに近似がともなう。モデルは実世界を抽象化したものにすぎない。さらに言うなら、ロボットやロボットの周囲の環境に発生する物理過程の一部を抽象化したものにすぎない。このようなモデル誤差は、ロボティクスでは無視されることの多い不確実さの原因の一つである。実際、最新鋭のロボットシステムの内部モデルは、実はずいぶん粗削りである。

不確実さは、**アルゴリズム上の近似**によっても発生する。ロボットは実時間システムである。そのことが、実行可能な計算量に制限をかけてしまう。よく知られているアルゴリズムの多くは近似計算であり、正確さを犠牲にして高い実時間応答性を得ている。

不確実さの程度は、実用の段階で決まる。組立ラインのようなロボットの応用例では、人間は巧みにシステムを組んで、最低限まで不確実さを減らすことに成功している。対照的に、住居や他の惑星で動作するロボットは、かなりの不確実さに対処しなければならない。そのようなロボットには、センサや内部モデルが正しい決定を行うだけの十分な情報を与えてくれなくとも、なにかしらの行動をすることが求められる。ロボティクスは今、開いた世界に進出しようとしているが、有用なロボットシステムを設計する上で、不確実さの問題は最も主要な障害となってきたのである。不確実さへの対処は、ロバストな実世界のロボットシステムの実現に向けて、最も重要なステップになるであろう。

それが、この本を執筆した所以である。

## 1.2 確率ロボティクス

本書は、**確率ロボティクス**の概観を包括的に示すものである。確率ロボティクスはロボットの知覚と行動に関する不確実さに特段の注目を置くという、ロボティクスでは比較的新しい試みである。確率ロボティクスで鍵となるアイデアは、確率論の演算を用いて不確実さを陽に表現することである。換言すれば確率ロボティクスでは、ある推定に対して結果を「最も妥当な」一つの推測値で表すのではなく、推測値の空間全体での確率分布として、推定情報が表される。そうすることによって、曖昧さや信頼度を数学的にまつとうな方法で表すことができる。確率ロボティクスでは、残っている不確

実さを考慮することで行動を慎重にしたり、さらには、もしそれが効果的であれば、不確実さを減らすようにロボットに行動させることも可能である。このように、確率的アルゴリズムは、ロボットが不確実さに直面したときに、きわどい行動を回避させることができる。その結果、多くの実世界におけるアプリケーションにおいて、他の手法をしのぐ性能を発揮する。

ここで二つの好例を挙げる。一方はロボットの知覚に関する例、もう一方は計画と制御に関する例である。

最初の例は、**移動ロボットの自己位置推定**である。ロボットの自己位置推定は、ある外界の座標系に対して、ロボット座標を推定する問題である。ロボットは環境の地図を有するが、センサの出力を利用して地図中での自身の位置を求めなければならない。図1.1は、そのような状況を描いたものである。環境には、区別できない三つのドアがあることをロボットは知っている。ロボットのタスクは、センシングと動きによって、自分がどこにいるのかを見出すことである。

この種の問題は、**大域的自己位置推定**として知られている。大域的自己位置推定では、ある既知の環境中のどこかにロボットが置かれ、そのロボットは全く位置が分からず状態から位置を求めなければならない。確率的な方法論では、ロボットのある時点での**信念**<sup>1,2</sup>が、全ての位置から成る空間中における確率密度関数として表現される。図1.1(a)には、そのような信念のグラフが描かれている。図中のグラフは、全ての位置に広がっている一様分布を表している。ここで、ロボットが最初のセンサデータを得て、ドアのそばにいることが分かったとしよう。確率的手法は、この情報をを利用して信念を更新する。その「事後」の信念は、図1.1(b)に描かれている。この図では、ドア付近の確率は増加し、それ以外の場所の確率は低下している。この確率分布が、三点においてピークを持っていることに注意しよう。これの各ピークは、環境中の見分けのつかないドア一つ一つに対応している。このままでは、決してロボットは自身の位置を**知**ることはできない。その代わり、この時点でロボットは、それぞれ同等に尤もらしい、三個の仮説をセンサから得たことになる。また、ロボットがドアに近くない場所にも、ゼロでない確率を割り当てていることにも注意しよう。この確率の割り当ては、センシングの不確実さに基づいた当然の結果である。というのは、小さいがゼロではない確率で、ロボットがドアがないのにドアがあると勘違いすることがあるからである。非ゼロの小さな確率を常に保持していることは、ロバスト性を実現するためには重要である。

ここで、ロボットが移動したとしよう。図1.1(c)は、移動の結果が確率密度関数に与える影響を描いている。確率密度関数は、ロボットが動いたのと同じ方向にシフトしている。また、関数のピークはそれぞれ広がっており、これはロボットの移動の不確実さを反映している。図1.1(d)には、ロボッ

▶1.2 訳注：この表現には違和感を覚える読者もいるであろうが、これはベイズ統計学の用語である。ベイズ統計学（ベイズ主義）では、確率はあくまでなんらかの主観的な根拠に基づいて計算されるものであり、計算された確率分布を「信念」、ある事象に対する確率を「信念の度合い」と呼ぶ。以後、「信念」とある場合は確率分布、あるいは確率密度関数のことを指していると考えて読み進めるとよい。

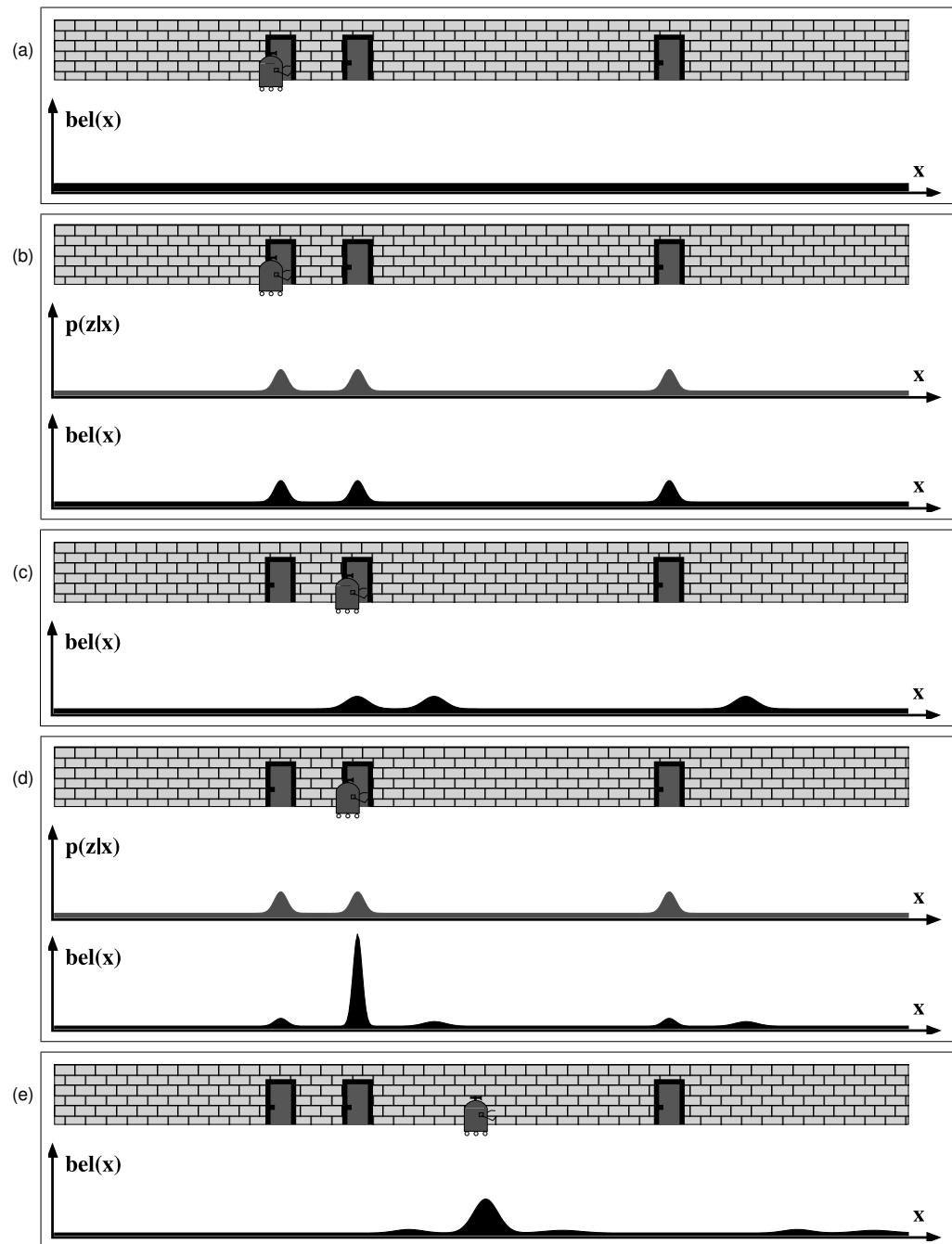


図 1.1

マルコフ位置推定の基本的なアイデア。一台の移動ロボットが大域的位置推定を行っている。マルコフ位置推定は 7 章、8 章で扱われる。

トがこの後にもう一回ドアを観測した後の確率密度関数が描かれている。この観測により、確率が一つのドア付近に密集する。そして、ロボットはこの時点で、自身の位置に関して高い確信を持つ。最後の図1.1(e)は、ロボットが廊下を進んだ後の確率密度関数を表している。

この例は、確率的な枠組みの多くの面を表現している。確率論的に述べると、この問題は状態推定問題であり、この自己位置推定の問題には**ベイズフィルタ**として知られるアルゴリズムが、ロボット位置の空間全域に対する事後推定に用いられている。情報は確率密度関数として表されており、この関数の変更は、センサの計測値から得られる情報、あるいはロボットの不確実さを増加させるような出来事によって起こる情報の損失を表現している。

二番目の例は、ロボットの行動計画や制御の分野における例を示している。前述のように確率的なアルゴリズムは、ある時点におけるロボットの不確実さを計算することができる。しかしそれだけではなく、確率的なアルゴリズムは将来生じる不確実さを予測することもでき、好みの行動を選ぶときにそのような不確実さの予測を考慮することができる。そのようなアルゴリズムの一つに**コースタルナビゲーション**<sup>1.3</sup>と呼ばれるものがある。コースタルナビゲーションの例を図1.2に示す。この図には、実際の建物の平面地図が見られる。上図では、ロボットが移動した後に推定された移動経路と、実際の移動経路が比較されている。推定経路がそれてしまつたのは、我々が先ほど言及したロボットの移動の不確実さが原因である。ここで、興味深い洞察ができる。それは、「全ての経路で、このように大きな推定の誤りが生じるわけではない」ということである。図1.2(a)の場合では、ロボットは比較的開けた場所を移動しており、正しく位置推定を続けられるほどの特徴あるデータを得られていない。図1.2(b)には、別の移動経路が見られる。この例では、ロボットは特徴のある一つのかどを探し、その後、正しく位置推定し続けるために壁沿いを「這つて」いる。当然のことながら、(a)と(b)では(b)の経路の方が不確実さが抑えられる。そのため、目的位置まで正確に到着する確率を上げることができる。

ロボットの制御に不確実さを適切に考慮する方法は多く存在し、この例はそのうちの一つを表している。経路推定が不確かになることを予測することで、ロボットはその不確実さを減らすためだけに遠回りな(b)の経路を選んでいる。(b)の経路は、目的地点まで正しく到達する可能性をずっと高くするという観点からは(a)よりも適切である。実は、(b)の経路を通ることは、ロボットが能動的に情報収集する例である。ロボットは確率的な思考から、目的地点を求めるために最良の行動選択が経路沿いに情報を探すことであることを発見している。このように確率的な行動計画手法は、不確実さを予測して情報収集の行動を計画することができる。そしてそのような計画結果は、確率的な制御手法によって実際のロボットの動作として実現される。

---

►1.3 訳注：航海用語で沿岸航法のことである。

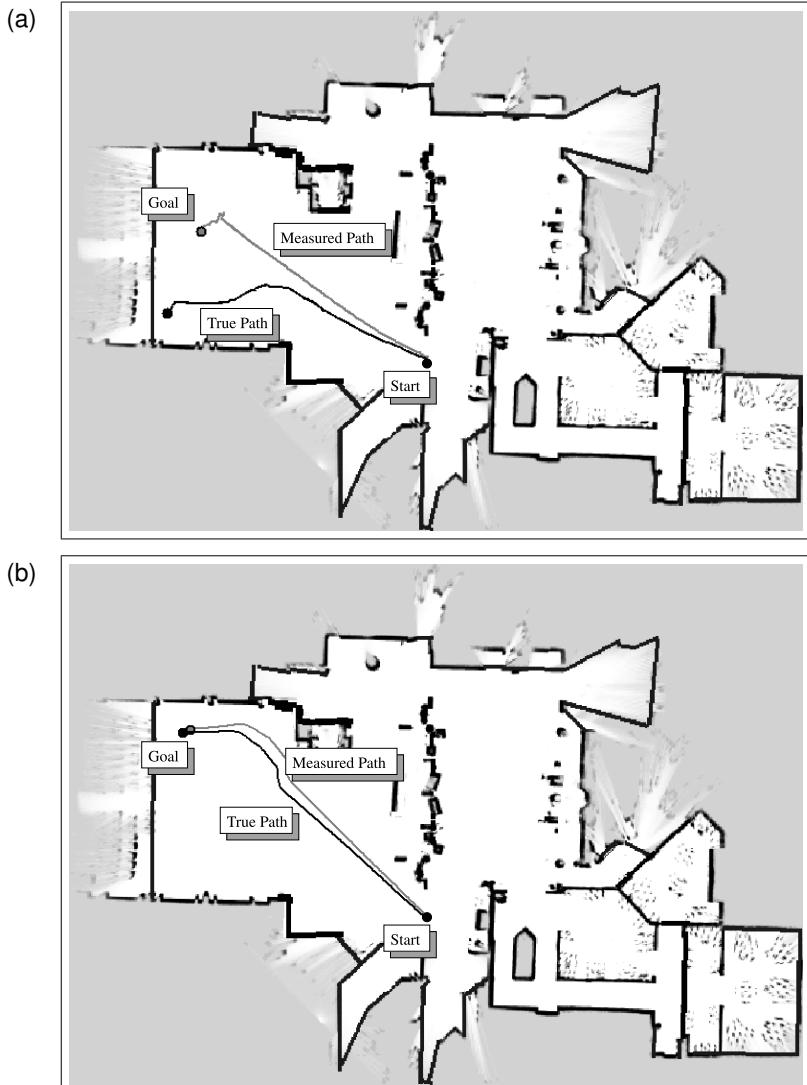


図 1.2

上図：特徴のない広い空間を移動するロボットは、自身の位置を見失いがちである。下図：そのような問題は、既知の障害物に近づいて移動することで回避することができる。これらの図はコースタルナビゲーションと呼ばれるアルゴリズムの適用前後の結果を比較するものである。コースタルナビゲーションについては、16 章で扱う。図は MIT の Nicholas Roy から提供された。

### 1.3 他の枠組みとの関係

確率ロボティクスは、モデルをセンサデータとシームレスに統合し、両方の限界を克服する方法である。この考え方は下位レベルな制御だけに限らず、上位から下位まであらゆるレベルのロボットのソフトウェアに適用できる。

モデルベースの動作計画方法や反射的なビヘービアベースドアプローチなど、従来のロボティクスにおけるプログラミング手法と異なり、確率的なアプローチはセンサの限界やモデルの限界に対して、よりロバストであるという傾向がある。確率ロボティクスによって、以前の枠組みよりも、複雑な実世界の環境をずっと良く計測できるようになる。このとき、不確実さの考慮ということがより重要な意味を持つ。実際、前述の自己位置推定の問題、とても広い環境の正確な地図を作成する問題のような、推定に関するロボティクスの難問に対しては、一部の確率的アルゴリズムだけが現在のところ实用に耐えうることが知られている。

従来のモデルベース手法と比較すると、確率的アルゴリズムでは正確なロボットのモデルがあまり要求されない。それによりプログラマーは、正確なモデルを構築するという徒労から解放される。また、一瞬一瞬のセンサ入力だけを制御に用いる多くのリアクティブな手法よりも、センサが正確であることの必要性が小さい。確率論の立場で見ると、**ロボットの学習問題**は長時間にわたる推定問題である。そのため、確率的アルゴリズムは、多種多様なロボット学習に対して、理にかなった方法論を提供できる。

しかしながらこれらの長所には、それ相当の代償がともなう。最も指摘される確率的アルゴリズムの制約は、**計算複雑性と近似の必要性**である。確率的アルゴリズムは、確率的でないライバル手法よりも本質的に効率が良くない。それは、確率的アルゴリズムが、一つの推定値ではなくて、推定値の空間中の確率密度全てを考慮することに起因する。近似については、ほとんどのロボットの世界が連續的であるために必要になる。事後の確率分布を正確に求める計算は、コンピュータにとっては扱いにくい場合が多い。コンパクトなパラメトリックモデル（ガウス分布など）で、きっちり不確実さが近似できる幸運な場合がある。しかし、そうでない場合、そのようなモデルを用いると計算が大雑把になりすぎる。そのときには、もっと複雑な確率の表現方法が必要となる。

近年のコンピュータの進歩によって、かつてないほどの計算量をバーゲン価格で利用できるようになった。この進歩は、確率ロボティクスの分野を確実に手助けしてきた。さらに近年の研究は、ロボティクスに関する難問（本書ではそれらの多くを徹底解説する）を解決するために確率的アルゴリズムの計算効率を向上させてきた。しかしながら、それにもかかわらず計算量の課題は残されている。本書の多くの部分では、各々の確率的な解法の有利な点、不利な点を調査することで計算量の問題を取り扱う。

## 1.4 本書の構成

本書は、四部構成であり、以下の章から構成されている。

- 2～4章ではいくつかの基本的なアルゴリズムと共に、本書のアルゴリズムに共通する基本的な数学的枠組みを導入する。これらの章は、本書を読む上で数学的な基礎となる。
- 5, 6章では、移動ロボットの確率モデルを提示する。これらの章はいろいろな意味で、古典的なロボティクスのモデルの、確率への一般化である。
- 移動ロボットの自己位置推定問題については7, 8章で論じる。これらの章では、5, 6章で設計したモデルに、推定アルゴリズムが組み込まれる。
- 9～13章では、ロボットによる地図生成の問題を扱う。これらの章で扱うアルゴリズムも依然として基本的なアルゴリズムに基づいたものである。しかし、地図生成の問題は非常に複雑であり、それを克服するために様々な技法を駆使する。
- 確率的な計画、制御手法は、14～17章で扱われる。このパートでは、いくつかの基本的な手法の解説から説明を始める。その後、確率的手法でロボットを制御するための、いくつかの実用的なアルゴリズムの説明に分岐する。

本書は、1章から順に読み進めるのが一番良い。しかしながら、我々はなるべく各章が独立した説明になるように試みた。最初に読むときには、今後頻繁に出てくる「…の数学的導出」という項を、本書の話の流れを見失うことなく安全に読み飛ばすことができる。

## 1.5 確率ロボティクスの講義

講義で本書を使うときには、学生が抽象的な数学の概念をしっかりと理解していない限り、各章を順番に教えていくことを**お勧めしない**。パーティクルフィルタは、ガウシアンフィルタよりも教えやすいし、学生たちは抽象的なフィルタのアルゴリズムよりも、具体的な移動ロボットの自己位置推定の話のほうに興味をかきたてられる傾向があるからである。我々の講義の場合ではたいてい2章から講義を始めて、その後すぐに7章、8章に移動する。自己位置推定を教えている途中、必要に応じて3～6章の内容に戻る。また、講義の初期の段階で学生を計画と制御の問題に触れさせるため、14章は早いうちに教えるようにしている。

講義では、本書の様々なアルゴリズムを説明するために、

[www.probabilistic-robotics.org](http://www.probabilistic-robotics.org)

で公開しているスライドやムービーを気兼ねなく使ってほしい。あなたの講義のウェブサイトや他の人が確率ロボティクスを教える上で有用な資料があれば、気兼ねなくその場所を知らせてほしい。本

書の内容は、手を動かす課題と共に教えることが最も良い。実機のためのプログラミングよりも、良いロボティクスの教育方法はない。そして、ロボティクスの落とし穴や困難さについて、自然界よりも多くを語るものはないのである。

## 1.6 関連する文献

ロボティクスの分野は、ソフトウェアデザインにまつわる一連の理論的枠組みと共に進んできた。最初の重要な概念は1970年代半ばに発生した。それは現在、モデルベースドアプローチとして知られているものである。このアプローチは、連続空間で高自由度のマニピュレータを制御することの難しさを示す、多くの研究から発生した[Reif 1979]。そして、[Schwartz et al. 1987]におけるロボットの運動に対する複雑性の解析、[Canny 1987]における指数オーダーの問題に対する最初の一般的な動作計画アルゴリズム、そして、モデルベースドの動作計画の分野に関する、Latombeによる強力な入門書[Latcombe 1991]に発展した（その後の画期的な研究については14章で述べる）。これら初期の研究では、Kavrakiらの動作計画手法[Kavraki et al. 1996]のように乱数を用いる研究は広範囲に行われていたものの、不確実さの問題は概して無視されていた。そして、正確な環境モデルが既知で、ロボットの動きも決定論的であると仮定されていた。残った不確実さを下位レベルの制御コントローラで吸収できるくらいに小さくなるように、モデルは十分に正確でなければならなかった。ポテンシャル法[Khatib 1986]やナビゲーション関数[Koditschek 1987]のように不慮の事態に（もしそれが観測できるならば）対応する方法を提供するアイデアもあったが、ほとんどの行動計画手法は単にマニピュレータを制御するための一つの経路を生成するだけのものであった。これらの初期の手法に応用例があったとしても、その環境では各種の不確実さは工夫して取り除かれるか、あるいは十分正確にセンシングできるか確認する必要があった。

1980年代半ば、ロボティクスの分野は劇的な変化を経験した。その時代、感覚フィードバックの欠如という問題が、ロボット研究者のコミュニティー全体において興味の中心になっていた。そのような中、ビヘイビアベースドロボティクスは強い信念の下、内部モデルを用いる考え方を排除した。その代わり、複雑なロボットの動きを創り出すのは、**状況に埋め込まれたエージェント**<sup>1.4</sup> の物理的環境との相互作用であるとした（この現象はしばしば**創発的行動**と呼ばれる[Steels 1991]）。そのような考え方から、ビヘイビアベースドロボティクスではセンシングが主要な役割を演じ、内部モデルを利用する考え方を受け入れられなかつた[Brooks 1990]。

ビヘイビアベースドロボティクスのブームは、初期の段階で、従来のモデルベースドの動作計画ではなしえなかつたいくつかの成功例が得られたことにより火がついた[Brooks 1986]。Brooksによる6脚ロボット“Genghis”は、その例の一つである。この例では、比較的単純な有限状態オートマトンがゴツゴツした地面上でもロボットの歩容を制御できることを示した。このような手法の成功の鍵は、センシングにあった。制御は完全に環境との相互作用に基づいており、環境との相互作用はロボットのセンサから知覚される。環境フィードバックを利用し、一見複雑なロボットを創るという点で、初期のいくつかの研究は印象的である[Connell 1990]。最近になって、この概

---

►1.4 situated agent

念は掃除ロボット [iRobots Inc.] として商業的に成功している。この掃除ロボットのソフトウェアは、ビヘイビアベースドの考えに基づいて作成されている。

ただし、内部モデルが欠如していることや、単純な制御機構に焦点を当てていることから、ほとんどのビヘイビアベースドなロボットシステムでは、その時々のセンサ情報で正しい行動決定ができる比較的単純なタスクでのみでしか動作が保証されない。このような限界が分かってきて、この分野の最近の研究では、ハイブリッド制御アーキテクチャが取り入れられている [Arkin 1998]。このアーキテクチャでは、ビヘイビアベースドは下位レベルの制御に用いられ、一方で上位の抽象的なレベルでは、モデルベースドな計画器がロボットの行動をつかさどる。今日のロボティクスでは、そのようなハイブリッドアーキテクチャは一般的なものとなっている。また、Gat による三層アーキテクチャに関する独創的な研究 [Gat 1998] も、ハイブリッドアーキテクチャとは無縁ではない。Gat のこの研究は、“Shakey the Robot” [Nilsson 1982] に起源がある。

現在の確率ロボティクスは、1990 年代半ばに出現した。もっとも、その起源はカルマンフィルタ [Kalman 1960] の発明までさかのぼることができる。いろいろな意味で、確率ロボティクスはモデルベースド手法とビヘイビアベースド手法の間に位置している。確率ロボティクスでは、モデルが使用される。しかし、そのモデルは不完全であり、制御には不十分のものであると仮定される。モデルとセンサ測定の両方を統合することにより、制御動作が考え出される。確率ロボティクスでは、統計学によってモデル構築とセンサ測定が数学的に結び付けられる。

以後の章では、確率ロボティクスの分野における主要な進歩の多くが議論される。この分野におけるいくつかの礎には、Smith と Cheeseman による、高次元の知覚問題に対するカルマンフィルタ手法の出現 [Smith and Cheeseman 1986]、占有格子地図の考案 [Elfes 1987; Moravec 1988]、そして Kaelbling らによる部分観測計画手法の再導入 [Kaelbling et al. 1998] が含まれる。この 10 年は手法が急増した。パーティクルフィルタが非常に流行するようになり [Dellaert et al. 1999]、ベイズ的な情報処理に注目した新しい計画手法が開発してきた [Thrun 2000b; Lebeltel et al. 2004; Park et al. 2005]。これらの手法の発展は、確率的アルゴリズムで動作するロボットシステムの開発と共に進んできた。[Durrant-Whyte 1988] による荷役用の産業用機械、博物館のエンタテイメントロボット [Burgard et al. 1999a; Thrun et al. 2000a; Siegwart 2003]、介護施設用ロボット [Pineau et al. 2003d] がある。確率的手法を多用する移動ロボット制御のためのオープンソースソフトウェアパッケージが、Montemerlo らによって記述されている [Montemerlo et al. 2003a]。

商用のロボットも、転換期を迎えている。国連と国際ロボット連盟による 2004 年のロボットマーケット市場報告書<sup>1.5</sup> では、世界のロボット市場規模は年間 19% 拡大していることを算出している。さらに目を見張るべきことは、産業応用からサービスロボットや市販品への、市場区分の変化である。

---

#### ►1.5 World Robotics Survey



## 2 再帰的状態推定 Recursive State Estimation

### 2.1 はじめに

確率ロボティクスの核心は、センサデータから状態推定するという考え方である。状態推定とは、推論できるが直接観測できない値をセンサデータから推定することである。ほとんどのロボット技術の応用では、何をすべきか決定することは、**確かな値**さえ分かれば比較的簡単である。例えば移動ロボットの移動は、そのロボットの正確な位置と、周囲の障害物が分かれれば比較的簡単である。残念ながら、それらの変数は直接測定できない。その代わり、ロボットはセンサを頼りにその情報を収集しなければならない。各センサは、それらの変数に対して、部分的な情報しか与えてくれない。そしてセンサによる計測は雑音によって乱れてしまう。状態推定はそのような乱れたデータから状態変数を正しく得ることである。確率的状態推定アルゴリズムでは、環境中で得る状態全体に対して、ロボットの信念確率分布が計算される。確率的状態推定の例は、「移動ロボットの位置推定問題」としてすでに本書に登場している。

本章の目的は、基本的な語彙と、センサデータから状態を推定するための数学の技法を紹介することである。

- 2.2 節では、本書を通じて使用される基本的な確率の概念を導入する。
- 2.3 節では、本書を通じて用いられる重要な専門用語を説明しながら、ロボットと環境の相互作用に対する我々の形式モデルについて説明する。
- 2.4 節では、ベイズフィルタを導入する。ベイズフィルタは、本書で紹介されるほぼ全ての技法の基礎をなす、状態推定のための再帰的なアルゴリズムである。
- 2.5 節では、ベイズフィルタを実装するときに生じる、数式表現上、計算上の問題を議論する。

## 2.2 確率の基本的な概念

本節では、本書で使用される基本的な表記法や確率の事実について、読者に慣れ親しんでもらう。確率ロボティクスでは、センサ計測値や制御やロボットと環境の状態のような量は、全て確率変数としてモデル化される。確率変数は多値をとる<sup>2.1</sup> ことができ、ある特定の確率法則に従った場合、多値をとる。確率的推論は、他の確率変数や観測されたデータから展開される、確率法則を用いた確率変数の計算過程のことである。

$X$  を一つの確率変数、 $x$  を、 $X$  ではないかと考えられるある特定の値とする。確率変数のスタンダードな例にコイン投げがある。この例では  $X$  は表か裏の 2 値をとることができる。もしこの例のように、 $X$  がとれる全ての値の空間が離散的であれば、我々は、

$$p(X = x) \quad (2.1)$$

というように、確率変数  $X$  が値  $x$  をとる確率を表現する。例えば、細工のないコインが  $p(X = \text{表}) = p(X = \text{裏}) = \frac{1}{2}$  で特徴づけられるとする。離散的確率の和は 1 である。つまり、

$$\sum_x p(X = x) = 1 \quad (2.2)$$

となる。確率は常に非負である。つまり  $p(X = x) \geq 0$  となる。

簡単のために、我々は可能ならば確率変数の明示的な記述を省略し、よく用いられる略記  $p(x)$  を  $p(X = x)$  の代わりに用いる。本書の手法のほとんどは、連続空間での推定や行動決定についてのものである。連続空間は、連続値をとれるいくつかの確率変数で特徴付けられる。特に断りなしに、全ての連続的な確率変数には確率密度関数 (probability density functions, PDFs) が定義できることを仮定する。平均  $\mu$ 、分散  $\sigma^2$  の一次元正規分布は、よく知られている密度関数である。正規分布の PDF は、次のようなガウス関数：

$$p(x) = (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\frac{(x - \mu)^2}{\sigma^2}\right\} \quad (2.3)$$

で与えられる。正規分布は、本書において重要な役割を演じる。我々は、正規分布を頻繁に  $\mathcal{N}(x; \mu, \sigma^2)$  と略記する。ここで、 $x, \mu, \sigma^2$  はそれぞれ確率変数の値とその平均と分散を表す。

正規分布の式 (2.3) では  $x$  がスカラ値であると仮定されているが、今後  $x$  はしばしば多次元ベクトルになる。そのようなベクトルに対する正規分布は多変量正規分布と呼ばれる。多変量正規分布は、次のような形式の密度関数：

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right\} \quad (2.4)$$

---

▶2.1 訳注：0, 1 以外の中間値をとることができるという意味。

で表される。ここで  $\mu$  は平均ベクトル,  $\Sigma$  は半正定値対称行列であり, 共分散行列と呼ばれる。上付き文字  $T$  は転置を表す。この PDF の指数部は  $x$  の二次式であり,  $\mu$  と  $\Sigma$  がこの二次関数のパラメータである。

ここで読者は, 式 (2.4) が式 (2.3) の厳密な一般化になっていることに留意すべきであろう。この二つの式は,  $x$  がスカラで  $\Sigma = \sigma^2$  ならば等価である。

式 (2.3) と式 (2.4) は, PDF の例である。離散系の確率分布の合計が必ず 1 になるように, PDF を積分すると

$$\int p(x) dx = 1 \quad (2.5)$$

のように 1 になる。しかしながら, 离散系の確率とは異なり, PDF の上限は 1 ではない。本書を通じて, 我々は確率, 確率密度, 確率密度関数という用語を交互に用いる。我々は暗に, 全ての確率変数は可測であり, 全ての連続な確率分布は必ず密度が定義できることを仮定する。

二つの確率変数  $X$  と  $Y$  の結合確率は,

$$p(x, y) = p(X = x \text{ and } Y = y) \quad (2.6)$$

で与えられる。この表現は, 確率変数  $X$  が値  $x$  をとり, かつ確率変数  $Y$  が値  $y$  をとるという事象が起こる確率を意味している。もし  $X$  と  $Y$  が独立であれば,

$$p(x, y) = p(x) p(y) \quad (2.7)$$

となる。しばしば, ある確率変数が他の確率変数の情報を有していることがある。ここで  $Y$  の値が  $y$  であることを我々はすでに知っていると仮定しよう。そしてその条件下で, 我々が  $X$  の値が  $x$  である確率を知りたいとする。そのような確率は,

$$p(x | y) = p(X = x | Y = y) \quad (2.8)$$

と表され, 条件付き確率と呼ばれる。もし  $p(y) > 0$  であれば, この条件付き確率は,

$$p(x | y) = \frac{p(x, y)}{p(y)} \quad (2.9)$$

で定義される。もし  $X$  と  $Y$  が互いに独立ならば, 我々は

$$p(x | y) = \frac{p(x) p(y)}{p(y)} = p(x) \quad (2.10)$$

を得る。言い換えると, もし  $X$  と  $Y$  が互いに独立ならば,  $Y$  は  $X$  の値についてなにも教えない。この場合,  $X$  に関心があるならば,  $Y$  を知るメリットは全くない。独立性と, その一般化として知られる条件付き独立性は本書を通じて重要な役割を演じる。

ところで, 全確率の定理と呼ばれる,

$$p(x) = \sum_y p(x|y)p(y) \quad (\text{離散系の場合}) \quad (2.11)$$

$$p(x) = \int p(x|y)p(y) dy \quad (\text{連続系の場合}) \quad (2.12)$$

という興味深い事実がある。これは、条件付き確率の定義と確率測度の公理から導かれる。ただし、 $p(x|y)$ あるいは $p(y)$ がゼロであれば、残りの因数の値にかかわらず $p(x|y)p(y)$ はゼロであると定義する。

条件付き確率 $p(x|y)$ を、その「逆」である $p(y|x)$ と関連づける**ベイズ則**は、同様に重要な法則である。この法則は、 $p(y) > 0$ という条件の下、

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\sum_{x'} p(y|x')p(x')} \quad (\text{離散系}) \quad (2.13)$$

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\int p(y|x')p(x') dx'} \quad (\text{連続系}) \quad (2.14)$$

のように記述される。ベイズ則は、確率ロボティクス（と確率的推論全般において）支配的な役割を果たす。もし $x$ の値を $y$ から推測したいとき、確率<sup>2.2</sup> $p(x)$ の分布は**事前確率分布**と呼ばれ、 $y$ は**データ**（例えば一つのセンサ測定値）と呼ばれる。確率 $p(x)$ の分布は、 $y$ が考慮される前の $X$ に関する知識を集約している。 $p(x|y)$ の分布は、 $X$ 上の**事後確率分布**と呼ばれる。式(2.14)が示すように、ベイズ則は、「逆」の条件付き確率 $p(y|x)$ を使い、 $p(x)$ から事後確率 $p(x|y)$ を計算する便利な方法を提供する。言い換えると、センサデータ $y$ から $x$ を推測したいとき、逆確率を通じてそれを可能にすると言える。逆確率は、 $x$ であると仮定した場合にデータ $y$ が得られる確率を特定するものである。ロボティクスでは、確率 $p(y|x)$ はしばしば**生成モデル**と呼ばれる。それは、状態変数 $X$ がどのようにセンサ計測値 $Y$ の**原因になる**かをある程度の抽象化レベルで記述されたものだからである。

ベイズ則の分母 $p(y)$ が $x$ に依存しないことは、重要な注目事項である。実際、式(2.13)と式(2.14)中の $p(y)^{-1}$ は、事後確率 $p(x|y)$ 中の $x$ の値にかかわらず一定である。このような理由から、 $p(y)^{-1}$ はしばしば**ベイズ則の正規化**の記号と呼ばれ、一般的に $\eta$ と記述される。 $\eta$ を用いると、ベイズ則は

$$p(x|y) = \eta p(y|x)p(x) \quad (2.15)$$

と表現できる。このような記法の利点は、その簡潔さにある。正規化のための式を厳密に表記する代わりに、最終的に計算結果が1に正規化されることを示す正規化の記号 $\eta$ を使う。厳密に数式を記述すると、ある種の数式展開では式がすぐに巨大化してしまう。本書を通じて、この種の正規化変数は $\eta$ （あるいは $\eta', \eta'', \dots$ ）で表される。**重要：**今後、たとえ違う値を表していても、異なる式で断りなく $\eta$ を用いることがある。

ここで、今まで議論してきたどの法則に対して、例えば変数 $Z$ などの別の確率変数で条件付けし

▶2.2 訳注：連続系では密度になるので適宜読み替えること。

ても問題ないことに気づく。例えば、 $Z = z$  の条件でベイズ則を当てはめると、 $p(y|z) > 0$  ならば、

$$p(x|y,z) = \frac{p(y|x,z)p(x|z)}{p(y|z)} \quad (2.16)$$

が得られる。

同様に、独立確率変数の結合確率の式 (2.7) を他の変数ベクトル  $z$  で、

$$p(x,y|z) = p(x|z)p(y|z) \quad (2.17)$$

と条件づけることができる。このような関係は、**条件付き独立性**として知られている。読者は簡単に確認できると思うが、式 (2.17) は、

$$p(x|z) = p(x|z,y) \quad (2.18)$$

$$p(y|z) = p(y|z,x) \quad (2.19)$$

と等価である。条件付き独立性は確率ロボティクスで重要な役割を担う。変数  $x$  の値を知る上で、変数の値  $z$  が分かっている場合に、変数  $y$  がなにも情報をもたらさない場合にはいつもこの性質が現れる。条件付き独立性は（絶対的な）独立性を意味しない。つまり、

$$p(x,y|z) = p(x|z)p(y|z) \neq p(x,y) = p(x)p(y) \quad (2.20)$$

となる。その反対もまた一般的に正しくない。つまり、絶対的な独立性は、条件付き独立性を意味しない。つまり

$$p(x,y) = p(x)p(y) \neq p(x,y|z) = p(x|z)p(y|z) \quad (2.21)$$

となる。ただし特別な場合は、条件付きと絶対的な独立性は一致する。

多くの確率的アルゴリズムでは、確率分布の特徴量や統計を得る必要がある。確率変数  $X$  の**期待値**は、

$$E[X] = \sum_x x p(x) \quad (\text{離散系}) \quad (2.22)$$

$$E[X] = \int x p(x) dx \quad (\text{連続系}) \quad (2.23)$$

で与えられる。確率変数が全て有限の期待値を有するわけではないが、そのような例外は本書で扱う事例とは関係がない。

期待値は、確率変数に対して線形である。特に、任意の数値  $a, b$  に対して、

$$E[aX + b] = aE[X] + b \quad (2.24)$$

が成り立つ。 $X$  の共分散は、

$$\text{Cov}[X] = E[X - E[X]]^2 = E[X^2] - E[X]^2 \quad (2.25)$$

のように得られる。共分散は、平均値からの予想される偏差の二乗を測るものである。上記で述べたように、多変量正規分布  $\mathcal{N}(x; \mu, \Sigma)$  の平均は  $\mu$ 、共分散は  $\Sigma$  である。

最後に、本書で重要な概念にエントロピーがある。確率分布のエントロピーは、次式

$$H_p(x) = E[-\log_2 p(x)] \quad (2.26)$$

で与えられる。この式は、

$$H_p(x) = - \sum_x p(x) \log_2 p(x) \quad (\text{離散系}) \quad (2.27)$$

$$H_p(x) = - \int p(x) \log_2 p(x) dx \quad (\text{連続系}) \quad (2.28)$$

となる。

エントロピーの概念は情報理論に由来する。エントロピーは、 $x$  の値が持つと期待される情報量のことである。離散系の場合は、 $p(x)$  が  $x$  の出現確率と仮定すると、 $-\log_2 p(x)$  は最適な符号化を用いた場合に必要なビット数である。本書では、ロボットが情報収集するとき、特定の行動を実行するとロボットが受けるであろう情報量を表現するためにエントロピーを用いる。

## 2.3 ロボットと環境の相互作用

図 2.1 は、ロボットと、その環境の相互作用を表している。環境、あるいは世界は、内部状態を持つ力学系である。ロボットは自身のセンサを用いて、環境の情報を取得することができる。しかしながらセンサは雑音を被る。また、たいてい直接知覚することができない物も多く存在する。結果としてロボットは、図の左側に描かれたような環境の状態に関する内部信念を維持・更新していくことになる。

また、ロボットは自身のアクチュエータを介して環境に影響を与えることができる。その効果は、ある程度予測不能であることが多い。それゆえ、各制御動作は環境の状態とロボットの内部信念の両方に影響を与える。

以下では、この相互作用をもっと形式的に記述する。

### 2.3.1 状態

環境は、状態で特徴付けられる。本書の内容では、ロボットと環境の、将来に影響する全ての局面として状態を考えることが便利である。あるいくつかの状態変数は、ロボットの近くにいる人々の位置のように、時間経過と共に変化する性質を持つ。他の状態変数は、(ほとんどの) 建物の壁のように静的である。変化する状態は動的状態と呼ばれ、静的状態や不变状態とは区別される。姿勢、速度、セ

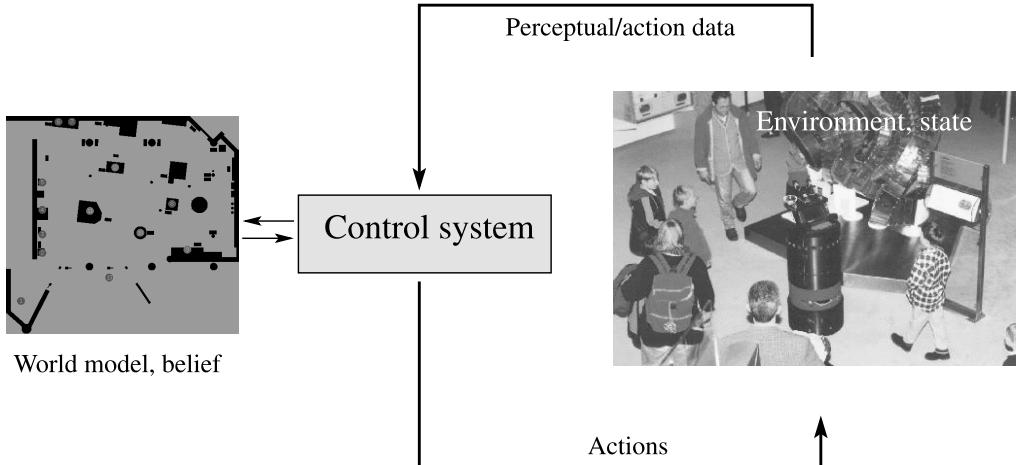


図 2.1

ロボットと環境の相互作用

ンサが正しく動作しているかどうかなど、ロボット自身に関する変数も状態に含まれる。

本書を通じて、状態は  $x$  で表される。 $x$  に含まれる状態変数は問題に応じて変化する。時刻  $t$  の状態は  $x_t$  と表記される。本書で扱われる典型的な状態は、以下のものである。

- **ロボットの姿勢。** これは、グローバル座標系に対するロボットの位置、向きで構成される。剛体の移動ロボットはそのような変数を 6 個持ち、そのうち 3 個はデカルト座標系のもので、残り 3 個は角度方向のもの（ロール、ピッチ、ヨー角）である。平面上を移動する剛体の移動ロボットにとっては、姿勢は通常 3 個の変数で表される。そのうちの 2 個は平面上の位置、残りは向いている方角（ヨー角）である。
- マニピュレータの場合、姿勢は**アクチュエータのコンフィギュレーション**に関する変数を含む。例えば、回転ジョイントの関節角を含むかもしれない。ロボットアームの各自由度は、ある時刻での一次元コンフィギュレーションで特徴付けられる。コンフィギュレーションの各次元は、ロボットの運動状態を表すための不可欠な要素である。マニピュレータのコンフィギュレーションは、しばしば**運動学的状態**と呼ばれる。
- **ロボットの速度と関節の速度**は、一般に**力学的状態**と言われる。空間を移動する剛体のロボットは、6 個の速度変数で特徴づけられる。それらは、各姿勢の変数と関連している。力学的状態は本書ではありません扱われない。
- **環境中の物体の位置と特徴**も、状態変数である。物体は、樹木であったり、壁であったり、ある壁面、床面に印された点かもしれない。そのような物体は、それらの外観（色や模様）で特徴付けられる。

厳密に状態をモデル化すると、ロボット環境は数十から数十億の状態変数を持つことになる（あ

るいはそれ以上かもしれない). いったいどれぐらいたくさんビット数があなたの環境を正確に表現するために必要か想像してほしい！

本書で扱われる多くの問題では、環境中の物体の位置は静的である。いくつかの問題では、ある物体が**ランドマーク**とみなされる。ランドマークになり得る物体は、環境中で目立つ不変の特徴を持ち、確実に認識できるものである。

- **移動物体と人の位置や速度**も、状態変数となり得る。環境中で動くものがロボットだけでないということは、よくあることである。ロボットの他の移動体は、めいめいに運動学的、力学的状態を持つ。
- ロボットの動作に影響を与える状態変数は、他にも数多く存在する。例えば、センサが壊れていかどうかも状態変数になり得る。電池で動くロボットにとっては、電池の残量も状態変数になる。そのような潜在的な状態変数を枚挙すると全くきりがないのである！

状態  $x_t$  が将来を予測するために最善のものであるとき、その状態は**完備**であると言われる。別の表現をすると、完備性とは、その状態ベクトルに、過去の状態や計測値、制御といった変数を加えても、将来を予測するために有益な情報が増えないことである。我々の定義する完備性では、将来の状態の推移が**決定論的な**関数でなくてもよいことに留意することは重要である。次のような条件「将来の状態は確率的に遷移する、しかし、将来に対して影響を与える変数が  $x_t$  の他にない、あつたとしても状態  $x_t$  に従属している」を満たす時間過程は**マルコフ連鎖**として一般に知られている。

状態の完備性の概念は、主に理論上において重要である。実際には、どのようなロボットシステムでも完備状態を特定することは不可能である。完備状態は、将来に影響する環境中の全ての特徴だけでなく、ロボット自身、ロボットのメモリの内容、周囲の人々の腹積もりなどを含む。これらのいくつかは、知ることが困難である。従って現実的な実装では、全ての状態変数から上記で述べたような状態変数を少し抜き出して状態を構成することになる。このような状態は、**不完備状態**と呼ばれる。

ほとんどのロボットアプリケーションでは、状態は連続である。つまり、 $x_t$  は連続体上で定義される。ロボットの姿勢は、連続状態空間の良い例である。ロボットの位置や向きは、外部の座標系で定義される。しかし時々、状態は離散的である。センサが壊れているかどうかをモデル化した（バイナリ）状態変数がその例である。連続と離散の両方の変数が含まれる状態空間は、**ハイブリッド状態空間**と呼ばれる。

興味の対象となるロボティクスの問題では、ほとんどの場合、状態は時間と共に変化する。本書では、時間は離散的に扱われる。つまり、興味の対象となる全ての事象は、離散的な時刻  $t = 0, 1, 2, \dots$  に起こる。ロボットがある時刻に行動を起こし始めたとき、その時刻は  $t = 0$  とされる。

### 2.3.2 環境との相互作用

ロボットとその周囲の環境との相互作用には、基本的な二種類のタイプがある。一方はロボットがアクチュエータによって環境の状態に影響を与えることであり、もう一方はセンサを通じて状態に関する情報を収集することである。この両者は同時に起こる。しかし、本書では解説の都合上、両者を分けて考えることとする。図 2.1 には、これらの相互作用が描かれている。

- **環境のセンサ計測。** 認識は、ロボットがセンサを用いて環境の状態に関する情報を獲得する過程のことである。例えば、ロボットは環境に関する情報を得るために、カメラで画像を撮ったり、測域センサを利用したり、触覚センサに問い合わせたりする場合があろう。そのような認識に関する相互作用の結果一つ一つは、**計測値**と呼ばれる。もっとも、我々はそのことを**観測**と呼んだり**知覚**と呼んだりもする。概して、センサの計測値は幾分かの遅れを伴って得られる。従って、計測値は少し前の時刻の状態に関する情報を与える。
- **制御動作**は世界の状態を変化させる。ロボットの環境に能動的に力を加えることで、そのような変化が起こる。**制御動作**の例には、ロボットの移動や物体のマニピュレーションが含まれる。状態はたいていの場合、ロボットがなにも行動しなくても変化する。従って、我々は理論の整合性のため、ロボットは**常に**制御動作を実行していると仮定する。たとえロボットがどのモータも動かさないことを選択していても、それはそのような制御動作であるとみなされる。実際、電源の入っているロボットは動かないでいる時も常に制御プログラムを実行しており、それと共に計測も行っていることが普通である。

ロボットが過去の全てのセンサ計測値や制御動作を記録しておくことができると仮定したとする。このとき、そのような記録を（実際に記憶されてようがなかろうが）我々は**データ**と呼ぶ。環境との相互作用が 2 種類あることから、二つの異なるデータが定義できる。

- **環境計測データ**は、各時刻の環境の状態に関する情報を与える。計測データの例としては、カメラ画像、測域センサの計測結果などが挙げられる。ほとんどの場合、我々は小さな時間のずれを単純に無視する。（例えば、レーザセンサは高速に環境をスキャンするものの微妙に時間差が発生するが、我々は計測結果のある時刻に一瞬で得られたものと単純に仮定する。）時刻  $t$  の計測データを  $z_t$  で表す。

表記上、本書のほとんどの部分では、ロボットがある時刻に得る計測データは一つであると仮定する。しかし、本書のほぼ全てのアルゴリズムは、ある時刻に複数の計測データが得られる場合にも簡単に拡張できる。次の表記

$$z_{t_1:t_2} = z_{t_1}, z_{t_1+1}, z_{t_1+2}, \dots, z_{t_2} \quad (2.29)$$

は、時刻  $t_1$  から  $t_2$  までに得られた全ての計測値の集合を表す ( $t_1 \leq t_2$ )。

- 制御データは、環境の状態の変化に関する情報を与える。移動ロボティクスでは、ロボットの速度は制御データの典型例である。5 [s], 10 [cm/s] と設定することは、このコマンドが発せられてから終了するまでに、ロボットが大体 50 [cm] 前方に移動することを示す。このように、制御は状態の変化に関する情報を伝えるのである。

速度の代わりにオドメータも制御データの源となる。オドメータはロボットの車輪の回転量を計測するセンサであり、走行量（オドメトリ）として状態の変化に関する情報を与える。オドメータはセンサであるが、制御動作の影響を計測するので我々はオドメトリを制御データとして扱う。

制御データは  $u_t$  で表される。変数  $u_t$  は時間  $(t-1; t]$  での状態の変化に対応する。計測データと同様、制御データは

$$u_{t_1:t_2} = u_{t_1}, u_{t_1+1}, u_{t_1+2}, \dots, u_{t_2} \quad (2.30)$$

で表される ( $t_1 \leq t_2$ )。たとえロボットがなにもしなくても環境の状態は変化するので、厳密に言えば時間が経過したということが制御データとなる。従って制御データは、時間ステップあたりに一つずつ存在し、そのなかには「なにもしない」という行動が含まれる。

計測と制御の区別は不可欠である。以後の本書の内容で両者は基本的に異なる役割を果たす。環境認識は、環境の状態に対する情報を提供する。従って環境認識はロボットの知識を増やす傾向を有する。一方行動は、ロボットの行動で発生する雑音やロボット環境の偶然性により、知識の損失を起こす傾向にある。行動と知覚の区別は、決してそれらが異なる時間に起こることを意味しているわけではない。むしろ、両者は同時に起こる。両者の区別は、完全に便宜上のものである。

### 2.3.3 確率的発生法則

状態や計測値の発生は確率の法則に支配されている。一般的に、状態  $x_t$  は状態  $x_{t-1}$  から確率的に発生する。 $x_t$  の発生の裏にある確率分布を特定することは、意味があろう。一見すると、状態  $x_t$  の発生という事象は過去の全ての状態や計測値、制御に従属しているように思われる。ゆえに、状態の発生の背後にある確率法則は、 $p(x_t | x_{0:t-1}, z_{1:t-1}, u_{1:t})$  という確率分布で与えられるように思われる。特に理由はない仮定だが、ここではロボットが制御動作  $u_1$  を行ってから計測  $z_1$  を行うこととする。

以下のことは重要である：状態  $x$  が完備ならば、 $x$  には過去に起こった全てのことが集約されている。特に、 $x_{t-1}$  は、その時点での過去の制御・計測、 $u_{1:t-1}$  と  $z_{1:t-1}$  の十分な統計になっていることは重要である。もし状態  $x_{t-1}$  が分かっていれば、 $p(x_t | x_{0:t-1}, z_{1:t-1}, u_{1:t})$  の全ての変数のなかで  $u_t$  のみが問題となる。

確率論では、この事実は次の等式

$$p(x_t | x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t | x_{t-1}, u_t) \quad (2.31)$$

で表現される。この等式で表現される性質は、条件付き独立性の例である。前述のように、条件付き

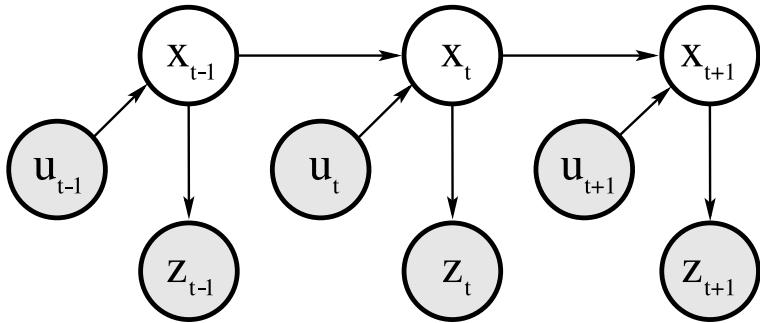


図 2.2

制御, 状態, 計測の推移の特性を表すダイナミックベイズネットワーク

独立性とは、いくつかの変数（条件付け変数）の値が分かっていると、特定のいくつかの変数が他の変数に対して独立となるという性質のことである。条件付き独立性は、本書で全面的に利用される性質である。そのことが、本書で紹介されるアルゴリズムのほとんどが計算機と相性が良い主な理由となっている。

さらに、計測で発生する過程をモデル化することも必要であろう。 $x_t$  が完備ならば、次の重要な条件付き独立性：

$$p(z_t | x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t | x_t) \quad (2.32)$$

が成り立つ。この式は、（ばらつきのあり得る）計測値  $z_t$  を予測するためには、 $x_t$  だけ分かれば良いということを意味している。過去の計測や制御や、過去の状態に関する知識は、 $x_t$  が完備であれば必要がない。

結局、得られた条件付き確率： $p(x_t | x_{t-1}, u_t)$  と  $p(z_t | x_t)$  は、いったい何なのであろうか。 $p(x_t | x_{t-1}, u_t)$  は、**状態遷移確率**である。状態遷移確率は、ロボットの制御  $u_t$  によって、どのように環境の状態が時間発展するかを規定するものである。ロボット環境は決定論的ではない。このことは  $p(x_t | x_{t-1}, u_t)$  が決定論的な関数ではなく、確率分布であることに反映されている。しばしば、状態遷移が時刻  $t$  と独立していることがある。この場合、我々は  $p(x_t | x_{t-1}, u_t)$  を  $p(x' | u, x)$  と記述する。ここで、 $x'$  は事後、 $x$  は事前の状態である。

確率  $p(z_t | x_t)$  は**計測確率**と呼ばれる。計測確率も時刻  $t$  に依存しないかもしれない。この場合、計測確率は  $p(z | x)$  と記述できる。計測確率は、環境の状態  $x$  からどの計測値  $z$  が得られるかということに関する確率法則を示す。計測値は、状態を雑音混じりで映し出すものであると考えることが適當である。

状態遷移確率と計測確率は、一対となってロボットとその環境に対する確率力学系を表現する。図 2.2 は、これらの確率で定義された状態と計測値の推移を表している。時刻  $t$  の状態は時刻  $t-1$  と制御  $u_t$  に、確率的に依存している。計測値  $z_t$  は時刻  $t$  の状態に確率的に依存している。このような時間発生

モデルは、**隠れマルコフモデル** (HMM)、あるいは**ダイナミックベイズネットワーク** (DBN) としても知られている。

### 2.3.4 信念分布

確率ロボティクスでもう一つ鍵となる概念に、**信念**がある。信念は、環境の状態に関するロボットの内部知識を反映するものである。我々はすでに、状態が直接計測できないものであるということを議論した。例えば、ロボットの姿勢が、あるグローバル座標系で  $x_t = \langle 14.12, 12.7, 45^\circ \rangle$  だったとしても、ロボットはたいていそれを知ることはできない。なぜなら姿勢を直接計測する方法がないからである（たとえ GPS があってもできない！）。その代わり、ロボットは自身の姿勢をデータから推測しなければならない。従って我々は、状態に関する内部の**信念**から、実際の状態を区別する。文中、信念の類義語として**状態の知識や情報状態**（後に出てくる情報ベクトルや情報行列と混同しないこと）が用いられる。

確率ロボティクスでは、条件付き確率分布によって信念が表現される。信念確率分布は、実際の状態に関して、あり得る全ての仮説に対して確率（あるいは密度値）を割りつける。信念確率分布は、得られたデータで条件づけられた事後確率分布であり、状態変数で構成される空間上で定義される。以後、ある状態の定義  $x_t$  上の信念を  $bel(x_t)$  で表す。この表現は、次の事後信念：

$$bel(x_t) = p(x_t | z_{1:t}, u_{1:t}) \quad (2.33)$$

を略記したものである。この事後信念は、過去全ての計測値  $z_{1:t}$  と制御  $u_{1:t}$  で条件づけられた、時刻  $t$  における状態空間上の確率分布である。

ここで読者は、信念が計測  $z_t$  を反映した**後の**ものとして暗に定義されたことに気づいたかもしれない。時折、制御  $u_t$  を実行した直後、 $z_t$  を反映する**前の**事後信念を計算する場合がある。このような事後信念は、

$$\overline{bel}(x_t) = p(x_t | z_{1:t-1}, u_{1:t}) \quad (2.34)$$

で表される。この確率分布は、確率的フィルタリングにおいてしばしば**予測**<sup>2.3</sup> と呼ばれるものである。この言葉の用法は、時刻  $t$  の計測が入る前に、 $\overline{bel}(x_t)$  が以前の事後信念に基づいて時刻  $t$  の状態を予測していることを反映している。 $\overline{bel}(x_t)$  から  $bel(x_t)$  を計算することは、**修正**<sup>2.4</sup>、あるいは**計測更新**<sup>2.5</sup> と呼ばれる。

---

▶2.3 prediction

▶2.4 correction

▶2.5 measurement update

```

1: Algorithm Bayes_filter( $bel(x_{t-1})$ ,  $u_t$ ,  $z_t$ ):
2:   for all  $x_t$  do
3:      $\bar{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$ 
4:      $bel(x_t) = \eta p(z_t | x_t) \bar{bel}(x_t)$ 
5:   endfor
6:   return  $bel(x_t)$ 

```

表 2.1

ベイズフィルタの基本アルゴリズム

## 2.4 ベイズフィルタ

### 2.4.1 ベイズフィルタアルゴリズム

信念を計算する最も一般的なアルゴリズムは**ベイズフィルタアルゴリズム**として与えられる。このアルゴリズムは計測や制御データから信念確率分布  $bel$  を計算する。ここでは、まず基本的なアルゴリズムを記述し、次に数値計算例と共にそのアルゴリズムを説明する。その後、これまでの仮定から、数学的にアルゴリズムを導く。

表 2.1 は、疑似コードでベイズフィルタを表したものである。ベイズフィルタは繰り返し演算である。つまり、時刻  $t - 1$  における信念  $bel(x_{t-1})$  から、時刻  $t$  における信念  $bel(x_t)$  が計算される。アルゴリズムへの入力は時刻  $t - 1$  における信念  $bel$  と最新の制御  $u_t$ 、最新の計測  $z_t$  である。アルゴリズムの出力は時刻  $t$  の信念  $bel(x_t)$  である。表 2.1 には、繰り返し演算のうちの一回分のみが記述されている。この一回分を**更新則**と呼ぶ。この更新則は、以前計算された信念  $bel(x_{t-1})$  から信念  $bel(x_t)$  を計算するために繰り返し適用される。

ベイズフィルタアルゴリズムは、二つの重要な過程から成り立っている。3 行目では、制御  $u_t$  が処理されている。状態空間全体の信念が、制御  $u_t$  と、直前の状態空間全体の信念に基づいて計算されている。ロボットが  $x_t$  に対して持つ信念  $\bar{bel}(x_t)$  は、二つの確率分布の積の積分（和）として得られている。一方は状態  $x_{t-1}$  に対する事前の信念であり、もう一方は制御  $u_t$  が引き起こす  $x_{t-1}$  から  $x_t$  への遷移に関する確率分布である。読者は、この過程が式 (2.12) と似ていることに気づくであろう。前述のように、この更新過程は制御更新あるいは**予測**と呼ばれる。

ベイズフィルタのもう一方の過程は、**計測更新**と呼ばれる。4 行目では、信念  $\bar{bel}(x_t)$  に計測  $z_t$  の得られる確率が乗算されている。この計算は、想定される各事後状態  $x_t$  に対して適用される。後でフィルタの式を展開するときに述べるが、この確率の積は一般に確率でなく、積分しても 1 にならない。そのため、正規化変数  $\eta$  によって正規化される。計測更新により、この更新則の最終的な解である信

念  $bel(x_t)$  が導かれ、6行目でこの信念が出力される。

事後信念を繰り返し計算するためには、時刻  $t = 0$  における初期信念  $bel(x_0)$  が境界条件として必要である。もし初期状態  $x_0$  が分かっているならば、 $bel(x_0)$  は真の状態  $x_0$  に全ての確率測度が集中し、他は確率ゼロの確率分布で初期化されるべきであろう。もし初期状態  $x_0$  に関して知識がなければ、 $bel(x_0)$  は  $x_0$  の定義域全域にわたる一様分布（あるいはディリクレ分布に属するような確率分布）で初期化されるであろう。初期状態  $x_0$  に対する部分的な知識は一様分布以外の確率分布で表現可能であるが、実際は初期状態が既知か未知かのどちらかであることがほとんどである。

ここまで述べたようなベイズフィルタのアルゴリズムは、ごく単純な推定問題に対してしか実装できない。具体的に言うと、3行目の積分と4行目の乗算を両方とも厳密に実行できるか、あるいは3行目の積分が（有限回の）足し算になるように状態空間を限定できる必要がある。

## 2.4.2 例

図 2.3 は、ロボットがカメラでドアの状態を推定しているところを表している。この例に基づいてベイズフィルタアルゴリズムを説明する。問題を簡単にするために、ドアは開いているか閉じているかの二つの状態のどちらかをとることにする。そして、このロボットだけがドアの状態を変えることができると仮定する。さらに、このロボットはドアの状態を最初知らないと仮定する。ドアの状態を与える代わりに、二つの状態に等しい事前確率

$$bel(X_0 = \text{open}) = 0.5$$

$$bel(X_0 = \text{closed}) = 0.5$$

を与える。

ここで、ロボットのセンサに雑音が生じると仮定しよう。この雑音は次のような条件付き確率：

$$p(Z_t = \text{sense\_open} | X_t = \text{is\_open}) = 0.6$$

$$p(Z_t = \text{sense\_closed} | X_t = \text{is\_open}) = 0.4$$

そして、

$$p(Z_t = \text{sense\_open} | X_t = \text{is\_closed}) = 0.2$$

$$p(Z_t = \text{sense\_closed} | X_t = \text{is\_closed}) = 0.8$$

で特徴づけられる。これらの確率は以下のことを表している。**閉じた** ドアを検知する場合にはロボットのセンサを比較的信頼でき、この場合に誤る確率は 0.2 である。しかしドアが開いている場合には、計測を誤る確率が 0.4 に達する。

最後に、ドアを押し開けるために、ロボットはマニピュレータを用いると仮定する。ドアがすでに

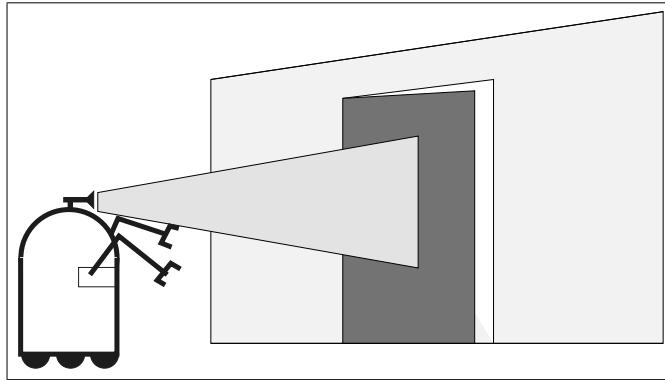


図 2.3

ドアの状態を推定する移動ロボット

開いているときには、ロボットがなにをしてもドアは空いたままになる。もし閉まっている場合には、ロボットが押してドアが開く確率は 0.8 である。つまり、

$$p(X_t = \text{is\_open} | U_t = \text{push}, X_{t-1} = \text{is\_open}) = 1$$

$$p(X_t = \text{is\_closed} | U_t = \text{push}, X_{t-1} = \text{is\_open}) = 0$$

$$p(X_t = \text{is\_open} | U_t = \text{push}, X_{t-1} = \text{is\_closed}) = 0.8$$

$$p(X_t = \text{is\_closed} | U_t = \text{push}, X_{t-1} = \text{is\_closed}) = 0.2$$

となる。ロボットはドアを押さない選択もできる。この場合状態は変化しない。このことは、次の条件付き確率

$$p(X_t = \text{is\_open} | U_t = \text{do\_nothing}, X_{t-1} = \text{is\_open}) = 1$$

$$p(X_t = \text{is\_closed} | U_t = \text{do\_nothing}, X_{t-1} = \text{is\_open}) = 0$$

$$p(X_t = \text{is\_open} | U_t = \text{do\_nothing}, X_{t-1} = \text{is\_closed}) = 0$$

$$p(X_t = \text{is\_closed} | U_t = \text{do\_nothing}, X_{t-1} = \text{is\_closed}) = 1$$

で表される。時刻  $t = 1$  で、ロボットは制御動作をしなかつたが開いたドアを検知した場合を考えてみる。その結果の事後信念は、事前信念  $bel(X_0)$ 、制御  $u_1 = \text{do\_noting}$ 、計測  $\text{sense\_open}$  からベイズフィルタによって計算される。状態空間が有限なので、3 行目の積分は、

$$\begin{aligned} \overline{bel}(x_1) &= \int p(x_1 | u_1, x_0) bel(x_0) dx_0 \\ &= \sum_{x_0} p(x_1 | u_1, x_0) bel(x_0) \\ &= p(x_1 | U_1 = \text{do\_nothing}, X_0 = \text{is\_open}) bel(X_0 = \text{is\_open}) \end{aligned}$$

$$+ p(x_1 | U_1 = \text{do\_nothing}, X_0 = \text{is\_closed}) bel(X_0 = \text{is\_closed})$$

のように有限個の足し算になる。ここで、とり得る二つの状態を状態変数  $X_1$  に置き換える。仮説  $X_1 = \text{is\_open}$  に対しては、

$$\begin{aligned} \overline{bel}(X_1 = \text{is\_open}) &= p(X_1 = \text{is\_open} | U_1 = \text{do\_nothing}, X_0 = \text{is\_open}) bel(X_0 = \text{is\_open}) \\ &\quad + p(X_1 = \text{is\_open} | U_1 = \text{do\_nothing}, X_0 = \text{is\_closed}) bel(X_0 = \text{is\_closed}) \\ &= 1 \cdot 0.5 + 0 \cdot 0.5 = 0.5 \end{aligned}$$

が得られ、同様に  $X_1 = \text{is\_closed}$  に対しては、

$$\begin{aligned} \overline{bel}(X_1 = \text{is\_closed}) &= p(X_1 = \text{is\_closed} | U_1 = \text{do\_nothing}, X_0 = \text{is\_open}) bel(X_0 = \text{is\_open}) \\ &\quad + p(X_1 = \text{is\_closed} | U_1 = \text{do\_nothing}, X_0 = \text{is\_closed}) bel(X_0 = \text{is\_closed}) \\ &= 0 \cdot 0.5 + 1 \cdot 0.5 = 0.5 \end{aligned}$$

が得られる。信念  $\overline{bel}(x_1)$  と事前信念  $bel(x_0)$  が同じであることは、特に驚くことではない。行動 `do_noting` が状態に対してなにも影響を与えていないし、この例では状態がひとりでに変化することはないので、これは当然のことである。

だが、計測が反映されると信念は変化する。前述のベイズフィルタアルゴリズムの 4 行目は、

$$bel(x_1) = \eta p(Z_1 = \text{sense\_open} | x_1) \overline{bel}(x_1)$$

となることを示している。 $X_1 = \text{is\_open}$ ,  $X_1 = \text{is\_closed}$  の二つの場合に対して、

$$\begin{aligned} bel(X_1 = \text{is\_open}) &= \eta p(Z_1 = \text{sense\_open} | X_1 = \text{is\_open}) \overline{bel}(X_1 = \text{is\_open}) \\ &= \eta 0.6 \cdot 0.5 = \eta 0.3 \end{aligned}$$

そして

$$\begin{aligned} bel(X_1 = \text{is\_closed}) &= \eta p(Z_1 = \text{sense\_open} | X_1 = \text{is\_closed}) \overline{bel}(X_1 = \text{is\_closed}) \\ &= \eta 0.2 \cdot 0.5 = \eta 0.1 \end{aligned}$$

が得られる。正規化の記号  $\eta$  は

$$\eta = (0.3 + 0.1)^{-1} = 2.5$$

と計算できる。従って、

$$bel(X_1 = \text{is\_open}) = 0.75$$

$$bel(X_1 = \text{is\_closed}) = 0.25$$

が求まる。この計算は次のステップでも繰り返せる。 $u_2 = \text{push}$ ,  $z_2 = \text{sense\_open}$  の場合,

$$\overline{bel}(X_2 = \text{is\_open}) = 1 \cdot 0.75 + 0.8 \cdot 0.25 = 0.95$$

$$\overline{bel}(X_2 = \text{is\_closed}) = 0 \cdot 0.75 + 0.2 \cdot 0.25 = 0.05$$

となり,

$$bel(X_2 = \text{is\_open}) = \eta 0.6 \cdot 0.95 \approx 0.983$$

$$bel(X_2 = \text{is\_closed}) = \eta 0.2 \cdot 0.05 \approx 0.017$$

となることは簡単に確かめられる。この時点で、ロボットは確率 0.983 でドアが開いていると信じていることになる。

一見、この確率は、ドアが開いているという仮説を採択し、それに応じて行動するためには十分高いように考えられる。しかし、そのような考え方をとると、不必要に高い代償を払うことになるかもしれない。もしドアが開いていないのに開いていると間違えると、損害が生じる（ロボットがドアと衝突するなど）。一方が起こりそうでなくとも、ロボットは両方の場合が起こり得ることを考慮して行動決定することが必要である。0.983 の確率で衝突しないと思いつつ自動操縦の飛行機に乗っていることを想像してほしい！

### 2.4.3 ベイズフィルタの数学的導出

ベイズフィルタアルゴリズムの正しさは、次のような帰納法で示すことができる。そのためには、事後確率分布  $p(x_t | z_{1:t}, u_{1:t})$  が<sup>3</sup>一つ前の事後確率分布  $p(x_{t-1} | z_{1:t-1}, u_{1:t-1})$  から計算できることを示す必要がある。それにより、時刻  $t=0$  の事前信念  $bel(x_0)$  を正確に初期化できるという前提の上で、帰納的にベイズフィルタアルゴリズムが正しいことが証明できる。

この数式展開では、2.3.1 節で定義したように、状態  $x_t$  が完備である必要がある。そして、制御がランダムに選択されることも前提として必要である。まず最初に

$$\begin{aligned} p(x_t | z_{1:t}, u_{1:t}) &= \frac{p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t | z_{1:t-1}, u_{1:t})}{p(z_t | z_{1:t-1}, u_{1:t})} \\ &= \eta p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t | z_{1:t-1}, u_{1:t}) \end{aligned} \quad (2.35)$$

のように、求めたい事後確率分布に対してベイズ則の式 (2.16) を適用する。ここで、状態が完備であるという仮定を利用する。2.3.1 節では、もし  $x_t$  の他に将来の状態遷移に影響を及ぼす変数がなければ、状態  $x_t$  は完備であると定義した。そして、もし（仮に）状態  $x_t$  が分かっており、計測  $z_t$  を予測したい場合、過去の計測や制御は追加情報をもたらさないことを述べた。これを数式で表すと、次のような条件付き確率：

$$p(z_t | x_t, z_{1:t-1}, u_{1:t}) = p(z_t | x_t) \quad (2.36)$$

で表現できる。この表現も、**条件付き独立性**の例である。上式により、式(2.35)を

$$p(x_t | z_{1:t}, u_{1:t}) = \eta p(z_t | x_t) p(x_t | z_{1:t-1}, u_{1:t}) \quad (2.37)$$

のように単純化することが可能となる。そして、

$$\overline{bel}(x_t) = \eta p(z_t | x_t) \overline{bel}(x_t) \quad (2.38)$$

となる。この等式は表2.1中のベイズフィルタアルゴリズムの4行目に記述されている。

次に、 $\overline{bel}(x_t)$ の項を、式(2.12)を用いて

$$\begin{aligned} \overline{bel}(x_t) &= p(x_t | z_{1:t-1}, u_{1:t}) \\ &= \int p(x_t | x_{t-1}, z_{1:t-1}, u_{1:t}) p(x_{t-1} | z_{1:t-1}, u_{1:t}) dx_{t-1} \end{aligned} \quad (2.39)$$

のように展開する。ここで再び、状態が完備であるという仮定を利用する。この仮定のもとでは、 $x_{t-1}$ が既知であれば、過去の計測や制御が状態 $x_t$ 以上に情報をもたらさない。このとき、

$$p(x_t | x_{t-1}, z_{1:t-1}, u_{1:t}) = p(x_t | x_{t-1}, u_t) \quad (2.40)$$

が成り立つ。この式の右辺には制御変数 $u_t$ が残っている。これは、状態 $x_{t-1}$ の前に $u_t$ が発生することがないからである。実際、読者は $p(x_t | x_{t-1}, u_t) \neq p(x_t | x_{t-1})$ となることにはすぐに気づくであろう。

最後に、 $p(x_{t-1} | z_{1:t-1}, u_{1:t})$  中にある条件付けの変数から、制御 $u_t$ を安全に取り除けることを述べておく。これにより、帰納的な更新式：

$$\overline{bel}(x_t) = \int p(x_t | x_{t-1}, u_t) p(x_{t-1} | z_{1:t-1}, u_{1:t-1}) dx_{t-1} \quad (2.41)$$

が得られる。容易に確かめられると思うが、この式は表2.1中のベイズフィルタアルゴリズムの3行目で実装されている。

要約すると、ベイズフィルタアルゴリズムは時刻 $t$ の計測と制御のデータで条件付けられた、状態 $x_t$ に対する事後信念を計算するものである。導出では世界がマルコフ性を有する、つまり状態が完備であることが仮定される。ベイズフィルタアルゴリズムの具体的な実装においては、3種類の確率分布：初期信念 $bel(x_0)$ 、計測確率 $p(z_t | x_t)$ 、状態遷移確率 $p(x_t | u_t, x_{t-1})$ が必要となる。我々はまだ実際のロボットシステムに対してこれらを求めるをしていないが、5章全体を $p(x_t | u_t, x_{t-1})$ 、6章全体を $p(z_t, x_t)$ のために割く。また、信念 $bel(x_t)$ の表現方法も必要であるが、これは3章で議論する。

#### 2.4.4 マルコフ性

マルコフ性、あるいは**状態の完備性**は本書の内容において実に基本的な役割を果たすため、言葉を整理する。マルコフ性とは、現在の状態  $x_t$  が既知ならば過去や未来のデータが独立していることを要求する仮定である。この仮定がいかに厳しいものであるか示すために、移動ロボットの自己位置推定の例を考えよう。移動ロボットの自己位置推定では、 $x_t$  がロボットの姿勢であり、ベイズフィルタは固定された地図に対するロボットの姿勢の推定に適用される。次の要因はセンサ信号に影響を与え、マルコフ性に反する事象を起こす。

- $x_t$  に含まれない環境中の、モデル化されないダイナミクス（自己位置推定の例では、動いている人々やそれらの人々がセンサ計測値に与える影響）
- $p(z_t | x_t)$  や  $p(x_t | u_t, x_{t-1})$  といった確率モデルが不正確（例：自己位置推定しているロボットのための地図）
- 信念を関数近似表現したときの近似誤差（例：後述の格子やガウス関数）、そして
- ロボット制御ソフトウェア内の、複数の制御に影響を与えるソフトウェア変数（例：「目標位置」のような変数は、一連の制御コマンド全体に影響を与える）。

原理上、これらの変数の多くは状態変数として状態の表現に組み込むことができる。しかし、不完備な状態は、より複雑な状態表現よりも、ベイズフィルタアルゴリズムの計算複雑性を減らすことが多い。実際には、ベイズフィルタはこのようなマルコフ性の破れに対して驚くほどロバストであることが知られている。しかし、経験から言って、状態  $x_t$  を定義するときには、モデル化されていない状態変数の影響がほぼ変量効果を持つ<sup>2.6</sup> ように注意すべきである。

### 2.5 表現と計算

確率ロボティクスでは、ベイズフィルタは様々な方法で実装される。次の二つの章で見ていくように、ベイズフィルタから派生した手法やアルゴリズムはかなり多く存在する。それぞれの手法は、計測確率や状態遷移確率、初期信念に関する異なる前提の上に成り立っており、前提の違いは、事後確率分布のタイプの違いで起こる。そして、それぞれの手法を実際に計算するためのアルゴリズムは、異なる様々な計算上の特性の違いを有する。経験から言うと、信念を厳密に計算する手法はかなり特別な場合にのみ存在する。通常のロボティクスの問題では、信念は近似される必要がある。アルゴリズムの複雑性の点で、近似の種類を選択することは重要な分岐点である。適した近似方法を見つけることは、たいてい挑戦的な問題であり、全てのロボティクスの問題に対して唯一で最適な答えはない。近似方法を選ぶときには、次のようなことを天秤にかけなければならない。

---

►2.6 訳注：考慮されていない変数による影響（雑音）が、平均値ゼロでランダムに発生すること。

- 計算効率.** 後で議論する線形ガウス近似のようないくつかの近似手法は、状態空間の次元に対して多項式時間で計算可能である。他の手法では指数時間が必要なことがある。後に議論するパーティクルを用いる手法には**任意時間特性**があり、近似の正確さと計算効率を調整することができる。
- 近似の正確さ.** いくつかの近似手法では、他よりも広範囲の確率分布をよりしっかりと近似できる。例えば、線形ガウス近似の使用はユニモーダルな分布の近似に限られる一方、ヒストグラム表現は、たとえ正確さに限りがあるあってもマルチモーダルな分布を近似できる。パーティクルによる表現では多様な分布が近似できるが、要求される正確さを達成するにはパーティクルの数が多くなってしまうことがある。
- 実装の簡単さ.** 確率的アルゴリズムの実装の困難さは、計測確率  $p(z_t | x_t)$  や状態遷移確率  $p(x_t | u_t, x_{t-1})$  の形状など、様々な要因に左右される。パーティクルによる表現を用いると、しばしば複雑な非線形系に対して驚くほど実装が簡単になる。これは、パーティクルによる表現が最近人気がある理由の一つである。

次の二つの章では、具体的でロボットに実装できるアルゴリズムを紹介する。これらのアルゴリズムでは、上記の評価指標に対してかなり異なった対応がなされている。

## 2.6 まとめ

本章では、環境とロボットの状態を推定する手段として、ロボティクスにおけるベイズフィルタの基本的な概念を紹介した。

- ロボットと環境の相互作用は、制御を選択することでロボットは環境を操作し、センサを通じてロボットが環境を知覚するという、二つの現象が結合した力学系としてモデル化される。
- 確率ロボティクスでは、ロボットと環境のダイナミクスは、次の二つの確率法則：状態遷移確率分布と計測確率分布、で定式化される。状態遷移確率分布は、ロボットの制御によって、状態がどのように時間変化するかを定式化したものである。計測確率分布は、どのように計測が状態によって左右されるかを定式化したものである。状態の遷移や計測に特有の不確かさを考慮し、両方の法則は共に確率的である。
- ロボットの**信念**は過去の全ての計測と制御から与えられる、(ロボットを含む) 環境の状態に対する事後確率分布である。**ベイズフィルタ**はロボティクスにおいて信念を計算するための原理的アルゴリズムである。ベイズフィルタは再帰的である。すなわち、時刻  $t$  の信念が時刻  $t - 1$  での信念から計算される。
- ベイズフィルタは、状態が過去の完全な集約であるという**マルコフ性**を仮定する。この仮定は、ロボットの過去の履歴を表現するためには、信念だけで十分であることを暗に示している。ロボティクスでは、マルコフ性の仮定は通常、単なる近似でしかない。本章では、マルコフ性が破れ

る条件を挙げた。

- ベイズフィルタは実際的なアルゴリズムではなくディジタルコンピュータ上には実装できないため、適切な近似が適用される。近似手法は、正確さ、計算の効率、実装の容易さに関する異なる基準に従って評価できそうである。

次の二つの章では、二つのよく知られている再帰的推定手法群について議論する。どちらの手法群も、ベイズフィルタから派生したものである。

## 2.7 関連する文献

本章における基本的な統計の内容は、ほとんどの確率統計の入門書でカバーされている。[DeGroot 1975; Subrahmaniam 1979; Thorp 1966] など、いくつかの古典的な教科書には、本書の内容の理解を助ける平易な解説が提供されている。さらに踏み込んだ解説は、[Feller 1968; Casella and Berger 1990; Tanner 1996] や [Devroye et al. 1996; Duda et al. 2000] に見出すことができる。ロボットと環境の相互作用の概念は、ロボティクスでは一般的である。このテーマは、Russell と Norvig によって AI の観点から議論されている [Russell and Norvig 2002]。

## 2.8 練習問題

- あるロボットが 0 [m] から 3 [m] まで計測できる距離センサを装備している場合を考える。簡単のため、計測対象は、0 [m] から 3 [m] の間にランダムに存在し、その存在確率はその区間の一様分布に従うこととする。残念ながら、このセンサは壊れることがあり、壊れたセンサは、実際の距離とは関係なく 1 [m] 未満の値を出力し続ける。我々は現在、センサが壊れている確率が  $p = 0.01$  であると知っていることとする。

ここで、ロボットが  $N$  回、センサから計測値を受け取ったとする。そして、全ての計測値が 1 [m] 未満だったとする。このときの確率モデルを定式化し、 $N = 1, 2, \dots, 10$  に対して、センサが壊れている事後確率を求めよ。

- 一日の天気が晴れ、曇り、雨の 3 通りのいずれかにはつきりと分類できる地域に住んでいると仮定する。毎日の天気の推移は、以下の表に従うマルコフ連鎖であるとする。

		明日の天気は…		
		晴れ	曇り	雨
今日の天気は…	晴れ	.8	.2	0
	曇り	.4	.4	.2
	雨	.2	.6	.2

- (a) 一日目が晴れであったとしよう. 「二日目：曇り, 三日目：曇り, 四日目：雨」となる確率を求めよ.
- (b) 上の表に従って「天気」を生成するシミュレータを書け.
- (c) 作成したシミュレータを用いて, このマルコフ連鎖の定常分布を求めよ. 定常分布とは, ある日が晴れ, 曇り, あるいは雨である確率を求めたものである.
- (d) 上記の表から得られる定常分布を計算するための方程式を記述して解くことは可能だろうか.
- (e) 定常分布のエントロピーを求めよ.
- (f) ベイズ則を用いて, 今日の天気から昨日の天気の確率を求めた表を計算せよ. (数値計算で解いても良いし, これまでの問題の解を用いて解いててもよい.)
- (g) この確率モデルに季節を加えることを考える. 上記の状態遷移関数は夏にしか適用できず, 冬, 春, 秋には別の状態遷移関数が必要となる. このことは, この確率過程のマルコフ性を破ることになるだろうか答えよ. 答えに対して説明をせよ.
3. 我々自身では天気を直接観測できず, センサで観測する場合を考える. 困ったことに, このセンサには雑音が混入する. 計測値のばらつきは, 次の計測モデルに従う.

		センサの出力は...		
		晴れ	曇り	雨
実際の天気は...	晴れ	.6	.4	0
	曇り	.3	.7	0
	雨	0	0	1

- (a) 一日目は晴れであったとする (これは事実として分かっているとする). そして, 二日目以降の4日間のセンサ計測値が「曇り, 曇り, 雨, 晴れ」であったとする. このセンサ計測値から, 五日目が実際に晴れである確率を求めよ.
- (b) (a) と同様, 一日目は晴れであったとする. 二日目から四日目まで, センサ計測値は「晴れ, 晴れ, 雨」であった. 二日目から四日目までのそれぞれの日について, 最も可能性の高い天気を求めよ. この問題に対して, 次の二種類の場合について答えよ: i) 答えを求める日までのデータだけを用いる, ii) 答えを求める日より未来のデータも用いることができる.
- (c) (b) と同じ状況 (一日目が晴れ, 二日目以降のセンサ計測値が「晴れ, 晴れ, 雨」) を考える. 二日目から四日目までの天気のシーケンスで, もっとも可能性の高いものを求めよ. そのシーケンスの確率を求めよ.
4. この問題では, ベイズ則をガウス関数に適用する. 我々が長い直線状の道の上にいる一台の移動ロボットであると想像せよ. 我々の位置  $x$  はこの道に沿った一次元変数である. 最初, 我々は自分の位置を  $x_{\text{init}} = 1,000 \text{ [m]}$  であると信じていたが, あるきっかけでこの推定値は不確かであると気づいた. その不確かさから, 我々は初期信念を分散  $\sigma_{\text{init}} = 900 \text{ [m}^2]$  のガウス関数でモデル化

した。

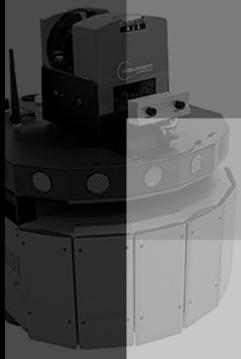
位置情報を得るために、我々は GPS 受信機に位置を聞いた。GPS 受信機は我々の位置が  $z_{\text{GPS}} = 1,100 \text{ [m]}$  であると応えた。この GPS 受信機の誤差は、分散  $100 \text{ [m}^2]$  であると分かっている。

- (a) 事前信念  $p(x)$ 、計測確率分布  $p(z|x)$  の確率密度関数を記述せよ。
- (b) ベイズ則を使うと、事後信念  $p(x|z)$  はどうなるか。 $p(x|z)$  がガウス関数であると証明できるかどうか答えよ。
- (c) 我々の事前信念と GPS 受信機の誤差に関する知識から考えると、計測値  $z_{\text{GPS}} = 1,100 \text{ [m]}$  が得られる確率はどれだけであったか。

ヒント：この問題は、二次形式を扱うための練習問題である。

5. 式 (2.17) とこれまで述べた確率の法則から、式 (2.18) と式 (2.19) を導出せよ。
6. 式 (2.25) を証明せよ。この式は何を意味しているか答えよ。





### 3

## ガウシアンフィルタ Gaussian Filters

### 3.1 はじめに

本章では、帰納的状態推定の重要な手法群の一つである**ガウシアンフィルタ**について述べる。歴史的に見て、ガウシアンフィルタは連続空間におけるベイズフィルタの最初の現実的な実装方法であり、(いくつかの短所はあるものの)間違いなく今まで最も実績のある手法群である。

ガウス関数を用いる手法は、多変量正規分布で信念を表現するという共通した方針を持っている。我々はすでに、式(2.4)で多変量正規分布の定義に触れた。もう一度書くと、多変量正規分布とは

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right\} \quad (3.1)$$

で表される確率分布である。この確率密度関数は二組のパラメータ：平均  $\mu$  と共分散行列  $\Sigma$  で特徴づけられる。平均  $\mu$  は、状態  $x$  と同じ次元を持つベクトルである。共分散行列は半正定値対称行列であり、一辺の要素数が  $x$  の次元の数だけある平方行列である。つまり、共分散行列内の要素数は、状態変数ベクトルの要素数の二乗となる。

事後信念をガウス分布で表現するという決まりは、大きな制約を生む。最も重要なことは、ガウス分布がユニモーダル、つまり一つの最大値しか持たない、ということである。そのような事後信念は、ロボティクスにおける多くの軌道追従問題では典型的である。これらの問題では、真の状態周辺の小さい領域の不確かさだけに対して、事後信念が計算される。ガウス分布で表現された事後信念は、大域的推定問題には適さない。大域的推定問題では多くの異なる仮説が存在し、状態空間において、それぞれの仮説がめいめいに高確率の領域を構成する。

平均と共分散でガウス分布を構成する方法は、**モーメント（積率）パラメータ化**と呼ばれる。それ

は、平均と共分散が、それぞれ確率分布の一次と二次のモーメントであるからである。正規分布では、他のモーメントはゼロとして扱われる。本章では、我々は代替のパラメータ化手法について議論する。その手法は、**カノニカル（正準）パラメータ化**と呼ばれたり、時々**ナチュラルパラメータ化**と呼ばれたりするものである。モーメントパラメータ化とカノニカルパラメータ化は数学的には等価であり、全単射な写像が存在する。しかしながら、これらから作成されるフィルタアルゴリズムは、コンピュータ上ではかなり異なる性質を持つ。後から分かるが、この両者は、双対なものとして考えると良い。つまり、一方のパラメータ化がコンピュータ上で難しければもう一方は易しくなる。

本章では、二つの基本的なガウシアンフィルタアルゴリズムを紹介する。

- 3.2 節では、カルマンフィルタについて説明を行う。カルマンフィルタは、モーメントパラメータ化で実装されたベイズフィルタであり、線形な力学や計測の関数で構成されるという、限られた部類の問題に対して適用される。
- 3.3 節では、カルマンフィルタを非線形な問題に適用するための、拡張カルマンフィルタについて述べる。
- 3.4 節では、アンセンティッドカルマンフィルタとして知られるもう一つの非線形カルマンフィルタについて述べる。
- 3.5 節では、情報フィルタについて述べる。情報フィルタは、ガウス分布をカノニカルパラメータ化して構成した、カルマンフィルタと双対なフィルタである。

## 3.2 カルマンフィルタ

### 3.2.1 線形ガウス型モデル

**カルマンフィルタ (KF)** は、ベイズフィルタの実装方法として我々の知る限り最も多く研究されたものである。カルマンフィルタは、Swerling と Kalman によって、**線形ガウス型モデル**<sup>3.1</sup> におけるフィルタリングや予測の手法として考案された [Swerling 1958; Kalman 1960]。線形ガウス型モデルについてはこのあと定義する。カルマンフィルタでは、連続状態に対して信念に関する計算が実装される。カルマンフィルタは、離散系やハイブリッドな状態空間には適用できない。

カルマンフィルタでは、モーメントパラメータ化によって信念が表現される。時刻  $t$  において、信念は平均  $\mu_t$  と共分散  $\Sigma_t$  で表される。ベイズフィルタの説明の際に述べたマルコフ性に加えて、次の三つの条件が満たされると、事後信念は**ガウス分布**となる。

1. 状態遷移確率  $p(x_t | u_t, x_{t-1})$  が、ガウス雑音を足した**線形**関数である必要がある。この条件は、次のような等式：

---

►3.1 linear Gaussian systems

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \quad (3.2)$$

で表される。ここで  $x_t$  と  $x_{t-1}$  は時刻  $t$  における状態ベクトルで、 $u_t$  は同時刻の制御ベクトルである。本書では、これらのベクトルは縦ベクトルである。

$$x_t = \begin{pmatrix} x_{1,t} \\ x_{2,t} \\ \vdots \\ x_{n,t} \end{pmatrix} \quad \text{そして} \quad u_t = \begin{pmatrix} u_{1,t} \\ u_{2,t} \\ \vdots \\ u_{m,t} \end{pmatrix} \quad (3.3)$$

と表現される。 $A_t$  と  $B_t$  は行列である。 $n$  が状態ベクトル  $x_t$  の次元であるとき、 $A_t$  は  $n \times n$  の正方行列である。 $B_t$  は  $n \times m$  行列であり、ここで  $m$  は制御ベクトル  $u_t$  の次元である。状態と制御をそれぞれ行列  $A_t$  と  $B_t$  で乗算して足すことで、この状態遷移関数は**線形**となる。カルマンフィルタは、このような線形系を想定している。

式(3.2)中の確率変数  $\varepsilon_t$  は、ガウス分布に従う乱数ベクトルであり、状態遷移の不確かさをモデル化するために足される。 $\varepsilon_t$  の次元は状態変数の次元に等しい。この乱数ベクトルの平均値はゼロベクトルで、共分散は  $R_t$  と記述される。式(3.2)で与えられる状態遷移確率は、線形な式にガウス雑音が加わっていることから、**線形ガウス型**であると呼ばれる。厳密に言えば、式(3.2)には定数項が足されることがあるが、後の議論に不必要なため、ここでは省略する。

式(3.2)は、状態遷移確率<sup>3.2</sup>  $p(x_t | u_t, x_{t-1})$ を定義している。この確率は、多変量正規分布の式(3.1)に、式(3.2)を当てはめることで得られる。事後状態の平均値は  $A_t x_{t-1} + B_t u_t$ 、共分散は  $R_t$  で与えられるので、状態遷移確率は

$$\begin{aligned} p(x_t | u_t, x_{t-1}) &= \det(2\pi R_t)^{-\frac{1}{2}} \\ &\exp\left\{-\frac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1} (x_t - A_t x_{t-1} - B_t u_t)\right\} \end{aligned} \quad (3.4)$$

で得られる。

2. 計測確率  $p(z_t | x_t)$  も**線形**であり、雑音はガウス雑音でなければならない。つまり、計測確率は

$$z_t = C_t x_t + \delta_t \quad (3.5)$$

で定式化できなければならない。ここで  $C_t$  は  $k \times n$  行列である。 $k$  は計測ベクトル  $z_t$  の次元である。ベクトル  $\delta_t$  は計測に関する雑音を表している。 $\delta_t$  の分布は平均ゼロ、共分散  $Q_t$  の多変量正規分布であるとする。計測確率は、次の多変量正規分布：

$$p(z_t | x_t) = \det(2\pi Q_t)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1} (z_t - C_t x_t)\right\} \quad (3.6)$$

►3.2 訳注：厳密に言えば、関数を指す場合は状態遷移確率の確率分布、数値を指す場合は確率密度。

```

1: Algorithm Kalman_filter( $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$ ):
2:    $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$ 
3:    $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$ 
4:    $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$ 
5:    $\mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t)$ 
6:    $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$ 
7:   return  $\mu_t, \Sigma_t$ 

```

表3.1  
線形ガウス型の状態遷移と計測のためのカルマンフィルタアルゴリズム

で与えられる。

3. 初期信念  $bel(x_0)$  が正規分布でなければならない。この分布の平均値を  $\mu_0$ , 共分散を  $\Sigma_0$  で表すこととする。このとき,

$$bel(x_0) = p(x_0) = \det(2\pi\Sigma_0)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x_0 - \mu_0)^T \Sigma_0^{-1} (x_0 - \mu_0)\right\} \quad (3.7)$$

となる。

上記の三事項は、事後信念  $bel(x_t)$  が任意の時刻  $t$  でガウス分布であるために十分な仮定である。この重要な事実は、後のカルマンフィルタの導出（3.2.4 項）で証明される。

### 3.2.2 カルマンフィルタアルゴリズム

表3.1にカルマンフィルタアルゴリズムを示す。カルマンフィルタは、時刻  $t$  の信念  $bel(x_t)$  を、平均値  $\mu_t$  と共分散  $\Sigma_t$  で表現する。カルマンフィルタへの入力は時刻  $t-1$  の信念であり、その信念は平均値  $\mu_{t-1}$  と共分散  $\Sigma_{t-1}$  で表されている。これらのパラメータを更新するために、入力として制御  $u_t$  と計測  $z_t$  も必要となる。カルマンフィルタの出力は、 $\mu_t$  と  $\Sigma_t$  である。

2行目と3行目では、 $z_t$  が反映される前の信念  $\overline{bel}(x_t)$  のパラメータ  $\bar{\mu}$  と  $\bar{\Sigma}$  が計算されている。この信念は、制御  $u_t$  を反映することで得られる。平均値  $\bar{\mu}$  は、式(3.2)の状態  $x_{t-1}$  を  $\mu_{t-1}$  で置き換えて、決定論的な部分のみを計算することで得られる。共分散  $\bar{\Sigma}$  は、事後状態が線形行列  $A_t$  の作用で事前状態から遷移することを反映して更新される。行列  $A_t$  は二回、共分散に掛け算される。

信念  $\overline{bel}(x_t)$  は、その後4~6行目で信念  $bel(x_t)$  に変換される。ここでは、計測  $z_t$  が反映される。4行目で計算される変数  $K_t$  は**カルマンゲイン**と呼ばれる。この変数は、計測を新たな状態推定にどの程度反映させるかを決定する。この変数の意味については3.2.4項で明らかにする。5行目では、平均



試し読みはお楽しみ  
いただけましたか？

ここからはManatee  
おすすめの商品を  
ご紹介します。

---

Tech Book Zone  
**Manatee** 

