

ランダム探索に力学フィルタを融合させた
リアルタイム障害物回避飛行誘導アルゴリズム

平成 28 年 1 月

伊多倉 京士朗

目次

概要	1
第 1 章 序言	5
1.1 研究背景と目的	5
1.1.1 無人航空機	5
1.1.1.1 概要	5
1.1.1.2 発展の歴史	5
1.1.1.3 市場の拡大と課題	7
1.1.2 火星探査航空機開発プロジェクト	9
1.1.2.1 火星探査の歴史	9
1.1.2.2 プロジェクトの概要	12
1.1.2.3 機体システムの課題	16
1.1.2.4 自律航行システムの課題	21
1.1.2.5 本研究の目的	22
1.1.3 有翼ロケット実験機	23
1.1.3.1 ロケット開発の歴史	23
1.1.3.2 プロジェクトの概要	27
1.1.3.3 開発状況	30
1.1.3.4 実験システム	33
1.1.3.5 本研究の目的	35
1.2 自律誘導アルゴリズムの先行研究	36
1.2.1 運動モデルを考慮した最適軌道生成の研究例	37
1.2.2 経路探索アルゴリズムを用いた最短経路生成の研究例	39
1.2.3 先行研究のまとめと本研究の意義	43
1.3 研究方針	45
1.3.1 障害物回避誘導に適したランダム探索の構築	45
1.3.2 航空宇宙機に適した力学フィルタの構築	45
1.3.3 ランダム探索と力学フィルタを用いた軌道プランナ	46
1.3.4 飛行試験によるオンボードリアルタイム誘導アルゴリズムの評価	46
1.4 論文構成	47
第 2 章 航空宇宙機の誘導に適した経路探索手法の構築	48
2.1 はじめに	48
2.2 経路探索アルゴリズムの比較検討	48

2.2.1	A* (A-Star)	49
2.2.2	PRM (Probabilistic Roadmap Method)	52
2.2.3	RRT (Rapidly-Exploring Random Trees)	53
2.2.4	軌道プランナに適用する経路探索法の選定	54
2.3	RRT の軌道プランナへの適用	55
2.3.1	疑似乱数モデル	55
2.3.2	基本アルゴリズム	56
2.3.3	衝突回避アルゴリズム	59
2.3.4	二点間結合アルゴリズム	61
2.3.5	経路簡略化アルゴリズム	64
2.4	ランダム探索のシミュレーション	67
2.5	本章のまとめ	70
第 3 章	力学フィルタと基礎的な軌道プランナの構築	71
3.1	はじめに	71
3.2	力学フィルタの設計概念	72
3.3	対象モデル	73
3.3.1	運動方程式の導出	73
3.3.2	機体諸元	77
3.3.2.1	火星探査航空機	77
3.3.2.2	有翼ロケット実験機	78
3.4	力学フィルタの構築	80
3.4.1	運動方程式の線形化	80
3.4.2	逆運動学	82
3.4.3	逆運動学を用いた力学フィルタ	83
3.5	ランダム探索と力学フィルタを組み合わせた基礎的な軌道プランナの構築	87
3.5.1	軌道プランナの概要	87
3.5.2	アルゴリズムの流れ	87
3.6	基礎的な軌道プランナの軌道生成シミュレーション	91
3.6.1	シミュレーション条件	91
3.6.2	シミュレーション結果	93
3.6.2.1	成功例	93
3.6.2.2	計算時間	98
3.6.2.3	失敗例と課題	99
3.7	本章のまとめ	105
第 4 章	ランダム探索と力学フィルタによる融合型軌道プランナの提案	106

4.1	融合型軌道プランナ	106
4.1.1	融合型軌道プランナの概要	106
4.1.2	アルゴリズムの構成	107
4.1.2.1	概念図	107
4.1.2.2	フローチャート	114
4.1.3	ベジエ曲線の適用	115
4.1.3.1	ベジエ曲線概念と定義	115
4.1.3.2	制御点の配置	117
4.1.3.3	比例定数 λ の決定	118
4.2	軌道生成シミュレーション	129
4.2.1	火星探査航空機	129
4.2.1.1	シミュレーション条件	129
4.2.1.2	シミュレーション結果	131
4.2.2	有翼ロケット実験機	141
4.2.2.1	シミュレーション条件	141
4.2.2.2	シミュレーション結果	143
4.3	本章のまとめ	153
第5章	リアルタイム誘導飛行シミュレーション	155
5.1	はじめに	155
5.2	6自由度運動方程式と制御則	156
5.3	誘導飛行シミュレーション	159
5.3.1	火星探査航空機	159
5.3.1.1	シミュレーション条件	159
5.3.1.2	シミュレーション結果	161
5.3.2	有翼ロケット実験機	171
5.3.2.1	シミュレーション条件	171
5.3.2.2	シミュレーション結果	173
5.4	本章のまとめ	183
第6章	UAVによるリアルタイム誘導計算飛行実験	184
6.1	はじめに	184
6.2	実験システム	184
6.2.1	システム概要	184
6.2.2	機体システム	185
6.2.2.1	機体諸元	185
6.2.2.2	搭載機器	186

6.2.2.3	空力係数	191
6.2.3	テレメトリシステム	193
6.2.4	コントローラ	194
6.3	実験条件.....	195
6.3.1	実験場所	195
6.3.2	障害物配置.....	195
6.3.3	軌道プランナのパラメータ	196
6.4	実験結果.....	198
6.4.1	パターン A.....	198
6.4.2	パターン B.....	201
6.4.3	パターン C.....	203
6.4.4	リアルタイム性能の評価	207
6.5	本章のまとめ	208
第 7 章	結言	209
	謝辞	211
	参考文献.....	212
	付録	215

图目录

Fig. 1.1.1 Sperry Aerial Torpedo	6
Fig. 1.1.2 Firebee	6
Fig. 1.1.3 RQ-2 Pioneer	6
Fig. 1.1.4 RQ-4 Global Hawk	6
Fig. 1.1.5 MQ-1 Predator.....	6
Fig. 1.1.6 FAZER.....	7
Fig. 1.1.7 D-SEND	7
Fig. 1.1.8 Mars3	9
Fig. 1.1.9 Viking Orbiter	9
Fig. 1.1.10 Viking Lander.....	10
Fig. 1.1.11 Mars Global Surveyor	10
Fig. 1.1.12 Mars Pathfinder.....	10
Fig. 1.1.13 PLANET-B.....	11
Fig. 1.1.14 MELOS Mission	11
Fig. 1.1.15 Olympus Mons.....	12
Fig. 1.1.16 Valles Marineris	12
Fig. 1.1.17 ARES.....	12
Fig. 1.1.18 ARES Mission Sequence	14
Fig. 1.1.19 High-Altitude Flight Test Using Balloon.....	15
Fig. 1.1.20 All Composite Wing Structure.....	18
Fig. 1.1.21 Lift Coefficient of Two-Dimensional Airfoil	19
Fig. 1.1.22 PIV of NACA0012 Airfoil	19
Fig. 1.1.23 Multi Objective Optimization of Mars Aircraft Airfoil.....	20
Fig. 1.1.24 Data Mining of Optimized Airfoil.....	20
Fig. 1.1.25 Dust Devil.....	21
Fig. 1.1.26 Orion Launch Experiment.....	23
Fig. 1.1.27 SpaceShipOne	24
Fig. 1.1.28 OREX	25
Fig. 1.1.29 HYFLEX	25
Fig. 1.1.30 ALFLEX.....	25
Fig. 1.1.31 Gliding Test	27
Fig. 1.1.32 Reentry Flight Test	27
Fig. 1.1.33 Artist Concept of WIRES	28

Fig. 1.1.34 Airframe and Engine Collaborative Design Meeting	29
Fig. 1.1.35 WIRES Development Plan.....	31
Fig. 1.1.36 Development History of WIRES Project	31
Fig. 1.1.37 Flight Experiment Setup	34
Fig. 1.2.1 Trajectory Optimization using MILP	37
Fig. 1.2.2 Trajectory Optimization using GA	38
Fig. 1.2.3 Real-Time Path Planning Using A* Algorithm	39
Fig. 1.2.4 Real-Time Path Planning using D* Algorithm.....	40
Fig. 1.2.5 Real-Time Path Planning using A*-EC.....	41
Fig. 2.2.1 A* Algorithm.....	50
Fig. 2.2.2 Simulation Results of A* Algorithm.....	51
Fig. 2.2.3 PRM	52
Fig. 2.2.4 Simulation Results of PRM	52
Fig. 2.2.5 Simulation Result of RRT.....	53
Fig. 2.3.1 Simulation Result of Mersenne Twister	56
Fig. 2.3.2 Flow Chart of Basic RRT	57
Fig. 2.3.3 Basic RRT Algorithm	57
Fig. 2.3.4 Simulation of Basic RRT.....	59
Fig. 2.3.5 Flow Chart of RRT with Collision Detection Algorithm	60
Fig. 2.3.6 Collision Detection Algorithm	60
Fig. 2.3.7 Simulation Result of RRT with Collision Detection Algorithm	61
Fig. 2.3.8 Flow Chart of RRT with Connection Algorithm	62
Fig. 2.3.9 Connection Algorithm	62
Fig. 2.3.10 Simulation Result of RRT with Connection Algorithm.....	63
Fig. 2.3.11 Flow Chart of RRT with Shortcut Algorithm	65
Fig. 2.3.12 Shortcut Algorithm	65
Fig. 2.3.13 Simulation Result of RRT with Shortcut Algorithm	66
Fig. 2.4.1 Simulation Results of Random Search	69
Fig. 2.4.2 Iteration Number of Each Obstacle Environment	69
Fig. 3.1.1 Conceptual Diagram of Dynamics Filter	71
Fig. 3.2.1 Problem of Trajectory Generation and Guidance Command.....	72
Fig. 3.2.2 Dynamics Filter and Guidance Command	72
Fig. 3.3.1 Coordinate System Definition	73
Fig. 3.3.2 Longitudinal Dyanmics.....	74
Fig. 3.3.3 Mars Exploration Aircraft Model	77
Fig. 3.3.4 Lift and Drag Coefficients of Mars Exploration Aircraft	78

Fig. 3.3.5 WIRES#014 Model	78
Fig. 3.3.6 Lift and Drag Coefficients of WIRES	79
Fig. 3.4.1 Forward Kinematics and Inverse Kinematics	83
Fig. 3.4.2 Block Diagram of Dynamics Filter	83
Fig. 3.5.1 Conceptual Diagram of Basic Trajectory Planner	87
Fig. 3.5.2 Tree Construction of Basic Trajectory Planner	88
Fig. 3.5.3 Dynamics Filter Process of Basic Trajectory Planner	89
Fig. 3.5.4 Distance Error Estimation Algorithm	89
Fig. 3.5.5 Terminal Condition of Dynamics Filter	90
Fig. 3.6.1 Simulation Result of Basic Trajectory Planner	94
Fig. 3.6.2 Guidance Commands	97
Fig. 3.6.3 Velocity and Distance Error	97
Fig. 3.6.4 Computer Calculation Time	98
Fig. 3.6.5 Trajectory with Collisions	100
Fig. 3.6.6 Incompletion of Trajectory Generation	101
Fig. 3.6.7 Trajectory Misgeneration	102
Fig. 3.6.8 Initial Angle Mismatch	103
Fig. 3.6.9 Indeterminate Terminal Trajectory Direction	104
Fig. 4.1.1 Conceptual Diagram of Fused Trajectory Planner	107
Fig. 4.1.2 Algorithm Flow of Coordinated Trajectory Planner (to be concluded) ...	113
Fig. 4.1.3 Algorithm Flow Chart	114
Fig. 4.1.4 Conceptual Diagram of Trajectory Planner with Bezier Curve	115
Fig. 4.1.5 Bezier Curve Drawing Method	116
Fig. 4.1.6 Control Point Arrangement Issue	117
Fig. 4.1.7 Control Points Allocation	118
Fig. 4.1.8 Bezier Curve and Resultant Trajectory of λ	120
Fig. 4.1.9 Evaluation Value E of λ	121
Fig. 4.1.10 Bezier Curve and Resultant Trajectory of λ	122
Fig. 4.1.11 Evaluation Value E of λ	123
Fig. 4.1.12 Bezier Curve and Resultant Trajectory of λ	125
Fig. 4.1.13 Evaluation Value E of λ	126
Fig. 4.1.14 Bezier Curve and Resultant Trajectory of λ	127
Fig. 4.1.15 Evaluation Value of λ	128
Fig. 4.2.1 Simulation Result (Obstacle Number 10)	133
Fig. 4.2.2 Command Time Histories	134
Fig. 4.2.3 Reference and Response State Variables	134

Fig. 4.2.4 Simulation Result (Obstacle Number 30)	135
Fig. 4.2.5 Command Time Histories	137
Fig. 4.2.6 Reference and Response States Variables	137
Fig. 4.2.7 Simulation Result (Obstacle Number 50)	138
Fig. 4.2.8 Command Time Histories	140
Fig. 4.2.9 Reference and Response State Variables	140
Fig. 4.2.10 Simulation Result (Obstacle Number 5)	144
Fig. 4.2.11 Command Time Histories	146
Fig. 4.2.12 Reference and Response State Variables	146
Fig. 4.2.13 Simulation Result (Obstacle Number 10)	147
Fig. 4.2.14 Command Time Histories	149
Fig. 4.2.15 Reference and Response State Variables	149
Fig. 4.2.16 Simulation Result (Obstacle Number 15)	150
Fig. 4.2.17 Command Time Histories	152
Fig. 4.2.18 Reference and Response State Variables	152
Fig. 5.3.1 Valles Marineris Image	159
Fig. 5.3.2 Terrain Map Model	159
Fig. 5.3.3 6 DOF Simulation Result without Trajectory Update	161
Fig. 5.3.4 Collision with Obstacle	162
Fig. 5.3.5 Reference and Response State Variables without Trajectory Update...	164
Fig. 5.3.6 Control Surface Angle Profile without Trajectory Update	165
Fig. 5.3.7 Angular Velocity Profile without Trajectory Update	165
Fig. 5.3.8 6 DOF Simulation Result with Trajectory Update	166
Fig. 5.3.9 Near-Miss Flight Situation	167
Fig. 5.3.10 Reference and Response State Variables with Trajectory Update	169
Fig. 5.3.11 Control Surface Angle Profile with Trajectory Update	170
Fig. 5.3.12 Angular Velocity with Trajectory Update	170
Fig. 5.3.13 Terrain Map at Van Horn / Texas	171
Fig. 5.3.14 6 DOF Simulation Result without Trajectory Update	173
Fig. 5.3.15 No-Fly Zone Violation	174
Fig. 5.3.16 Reference and Response State Variables without Trajectory Update.	175
Fig. 5.3.17 Control Surface Angle Profile without Trajectory Update	177
Fig. 5.3.18 Angular Velocity Profile without Trajectory Update	177
Fig. 5.3.19 6 DOF Simulation Result with Trajectory Update	178
Fig. 5.3.20 Reference and Response State Variables with Trajectory Update	181
Fig. 5.3.21 Control Surface Angle Profile with Trajectory Update	182

Fig. 5.3.22 Angular Velocity Profile with Trajectory Update	182
Fig. 6.2.1 Flight Test and Onboard System	185
Fig. 6.2.2 UAV	185
Fig. 6.2.3 System Arrangement	186
Fig. 6.2.4 Navigation and Control Board	186
Fig. 6.2.5 Guidance Board.....	187
Fig. 6.2.6 Absolute Pressure Sensor (Altitude)	187
Fig. 6.2.7 Differential Pressure (Velocity)	188
Fig. 6.2.8 Differential Pressure.....	188
Fig. 6.2.9 Calibration of Pressure Altimeter	189
Fig. 6.2.10 Calibration of Angle of Attack Using Pitot Tube	189
Fig. 6.2.11 Calibration of Side Slip Angle Using Pitot Tube	190
Fig. 6.2.12 GPS/INS Sensor	190
Fig. 6.2.13 Brushless Motor	191
Fig. 6.2.14 Wind Tunnel Test for Body Drag Measurement.....	192
Fig. 6.2.15 Lift and Drag Coefficients of UAV.....	193
Fig. 6.2.16 Communication Module.....	193
Fig. 6.2.17 Monitoring Software	193
Fig. 6.2.18 Radio Controller.....	194
Fig. 6.3.1 Flight Test Area.....	195
Fig. 6.3.2 Obstacle Pattern	196
Fig. 6.4.1 Real Time Calculation Flight Test	198
Fig. 6.4.2 Trajectory Generation Calculation of Pattern A.....	199
Fig. 6.4.3 Trajectory Generation Calculation of Pattern B	201
Fig. 6.4.4 Trajectory Generation Calculation of Pattern C	203
Fig. 6.4.5 Command Time Histories for Case 2 of Pattern C.....	205
Fig. 6.4.6 Reference and Response Histories for Case 2 of Pattern C	205
Fig. 6.4.7 Incomplete Trajectory Generation	206
Fig. 6.4.8 Average Calculation Time for Three Obstacle Patterns.....	207

表目次

Table 1.1.1 Planetary Environment of Mars and Earth.....	16
Table 1.1.2 Project Team	29
Table 1.2.1 Feature of Research.....	43
Table 3.3.1 Mars Exploration Aircraft Specifications.....	77
Table 3.3.2 WIRES Specifications.....	79
Table 3.6.1 Searching Environment	91
Table 3.6.2 Control Gain of Dynamics Filter	91
Table 3.6.3 Command Limiters.....	92
Table 3.6.4 Initial Parameter.....	92
Table 3.6.5 Computational Environment	92
Table 4.1.1 Initial States and Control Gains of Mars Aircraft.....	119
Table 4.1.2 Initial and Terminal Trajectory States for Evaluating Direction Angle	119
Table 4.1.3 Initial and Terminal Trajectory States for Evaluating Flight Path Angle	122
Table 4.1.4 Initial States and Control Gains of WIRES#015.....	124
Table 4.1.5 Initial and Terminal Trajectory States for Evaluating Direction Angle	124
Table 4.1.6 Initial and Terminal Trajectory States for Evaluating Flight Path Angle	126
Table 4.2.1 Simulation Conditions for Mars Aircraft	129
Table 4.2.2 Trajectory Planner Parameters for Mars Aircraft	130
Table 4.2.3 Initial States for Simulation.....	130
Table 4.2.4 Control Gains.....	130
Table 4.2.5 Command Limiters.....	131
Table 4.2.6 Simulation Conditions for WIRES	141
Table 4.2.7 Trajectory Planner Parameters for WIRES	142
Table 4.2.8 Initial States for Simulation.....	142
Table 4.2.9 Control Gains.....	143
Table 4.2.10 Command Limiters.....	143
Table 5.3.1 Initial and Terminal Trajectory Conditions	160
Table 5.3.2 Control Gains and Trim Angles for 6DOF Simulation	160
Table 5.3.3 Initial and Terminal Trajectory Conditions	172

Table 5.3.4 Control Gains and Trim Angles for 6 DOF Simulation	172
Table 6.2.1 Specification of UAV	185
Table 6.2.2 Navigation and Control CPU Specification	186
Table 6.2.3 Guidance CPU Specification	187
Table 6.2.4 Specification of Altitude Pressure Sensor	187
Table 6.2.5 Specification of Velocity Pressure Sensor.....	188
Table 6.2.6 Specification of Angle of Attack and Side Slip Angle Pressure Sensors	188
Table 6.2.7 Specification of GPS/INS Sensor	190
Table 6.2.8 Specification of Brushless Motor	191
Table 6.3.1 Trajectory Planner Setup	196
Table 6.3.2 Control Gains.....	197
Table 6.3.3 Command Limiters.....	197

概要

本論文では、障害物や飛行禁止区域が多数存在する複雑で制約の大きい飛行環境を想定し、航空宇宙機用の高速かつ厳密に衝突回避軌道を生成する誘導手法を提案する。

近年、無人航空機の研究開発が世界中で行われている。開発目的は、軍事用の偵察機や民間向けの農薬散布など多岐にわたる。また、学術目的としても盛んに開発が行われ、九州工業大学宇宙システム研究室では、宇宙探査と宇宙旅行を最終目標とする無人機開発プロジェクトを推進している。前者は、宇宙航空研究開発機構（JAXA/ISAS）が主導している火星探査航空機開発プロジェクトであり、後者は、本研究室が主体となり産官学連携で研究開発を行う有翼ロケット開発プロジェクトである。

火星探査航空機開発プロジェクトは、世界でも前例のない航空機による惑星探査を目指している。航空機を用いた惑星探査は、従来の高高度からの人工衛星による観測よりも高精細に、地上を走行するローバーよりも短時間で広範囲を観測できるという大きなメリットがある。火星の飛行環境は、地球の $1/3$ の重力と $1/100$ の大気密度であり従来の航空機よりも軽量の構造と高性能な翼型が要求されるため、工学的な観点においても極めて意義の大きな挑戦となる。さらに、同プロジェクトにおいて誘導システムは、ミッションの成立性を左右する重要な研究テーマの一つである。火星での飛行探査の際には、高度数キロにも及ぶ山々を避けて飛び、また砂塵嵐などの外乱の影響で目標軌道から大きく外れた場合にも機体自らが迅速に軌道修正を行う自律誘導アルゴリズムが必要とされる。

有翼ロケット開発プロジェクトは、地上から打上げて高度 100km の宇宙空間に到達し地球へ再突入後、滑空飛行して目的地に帰還する完全再使用型宇宙輸送システムの実現を目指している。開発目的としては、高層大気の観測や大陸間高速輸送、宇宙旅行などが想定されている。航空機のように繰り返し使える宇宙輸送システムが実現すると、宇宙開発のコストは劇的に削減され、宇宙産業の大幅な活性化が期待される。同プロジェクトにおいても自律誘導システムの研究は、安全に任務を遂行するために極めて重要となる。有翼ロケットの飛行中に機器の故障などの不具合の発生その他、諸事情による飛行中断の判断が下された場合、その場から目標の着陸地点まで飛行禁止区域を避けながら無事に帰還できるリアルタイム誘導アルゴリズムが要求される。

このように、航空宇宙機自らが飛行軌道を決定する自律誘導アルゴリズムは、今後の無人機市場の発展に伴いますます需要が高まってくると思われる。本論文では、無人航

空宇宙機が迅速に複雑な障害物や飛行禁止区域を避けて飛行し、目的地まで到達することを可能にするリアルタイム自律誘導アルゴリズムの実現を目指して以下の研究を行った。

ランダム探索と力学フィルタを用いた自律誘導アルゴリズムの構築

自律誘導アルゴリズムの従来研究には、最短軌道を求める問題など軌道の最適性に主眼が置かれているものが多い。このような最適化計算を必要とする場合、膨大なケースの軌道を評価し、より良い解を選択するための収束計算が行われるため、一般的には解を得るのに膨大な計算コストがかかる。一方、ロボティクスの分野において経路計画法としての研究が盛んな経路探索アルゴリズムを航空機の誘導に応用して高速に飛行経路を生成する研究例も存在する。しかし、それらは機体の運動を考慮しない、もしくは予め決められたパラメータに基づき、飛行経路を円弧や直線などの近似経路として表現することがほとんどである。近似経路の場合、運動の時定数と比べて十分に大きな経路であれば問題とはなりにくい。しかし、近距離の経路探索を行う場合、実際に航空機が飛行できる軌道との誤差が無視できず、障害物との衝突の大きな原因と成り得る。これらの従来研究の課題を踏まえて、本研究では、機体の運動モデルを考慮し、かつ高速な軌道生成法の確立を目指してアルゴリズムの構築を行った。

本研究は、主に二つの概念により構成される。一つは、障害物を回避する経路を広域に探索するための手法（ランダム探索）であり、もう一つは、運動モデルを考慮して飛べる軌道を与えるための手法（力学フィルタ）である。本稿では、これらを包括した誘導アルゴリズムを軌道プランナと呼ぶ。

まず、前者の障害物の回避経路を探索するアルゴリズムとして、代表的な経路探索法の特徴を比較し検討した。その中でも特に、環境に依存せず探索領域を均一に探索でき、かつ確実に解を得ることができる **Rapidly-Exploring Random Tree (RRT)** をランダム探索法として適用した。また、**RRT** の基本アルゴリズムに、衝突回避や二点間を結合する機能、無駄な経路を省く機能などを付加し、航空宇宙機の誘導に適した工夫を行った。さらに、障害物の数を変動させて3次元経路探索シミュレーションを行った。その結果、障害物の数を0個から100個まで増加しても衝突しない探索結果を得られることが示され、また障害物数の増加に対する探索の反復回数の相関関係も明らかとなった。

次に、力学フィルタの手法を検討した。この力学フィルタとは、前述の探索経路まわりに機体運動モデルの微分方程式を数値計算することにより厳密な軌道を獲得する方法である。従来の誘導システムの多くが座標データのみを生成していたのに対し、制御システムでの扱いが容易となるように、飛行経路を生成する推力や姿勢角を得られるように工夫した。また、障害物回避などの所望の軌道を獲得するため、ロボティクスの分野で研究される逆運動学理論を用いた力学フィルタを提案した。同手法は、モデルの常微分方程式の微小運動変化をヤコビアンを用いて表現し、出力項に所望の微小変化量を与えて入力項について解くと、必要な入力量が逆算できる手法である。

これらのランダム探索と力学フィルタを組み合わせて基礎的な軌道プランナを構築した。まず、ランダム探索により障害物回避経路を発見し、その目標経路に対して力学フィルタにより飛べる軌道へと変換した。シミュレーションの結果より、飛行に適した滑らかな軌道を生成することに成功した。しかし、障害物回避性を十分に保障できないことや、意図しない方向に誤誘導が発生するなど多くの課題が見つかった。

そこで、これらの課題を克服するため、力学フィルタをランダム探索に融合させる新しい軌道プランナ(融合型軌道プランナ)を考案した。前述の基礎的な軌道プランナは、ランダム探索により障害物回避経路を探索した後に力学フィルタ処理を行ったが、本手法では、ランダム探索と力学フィルタを同時に実行する。本アルゴリズムのシミュレーション結果では、火星探査航空機と有翼ロケット実験機の両モデルにおいて、障害物の数を増加させても厳密な回避軌道が生成できることが証明された。また、基礎的な軌道プランナでは困難であった、初期および終端の軌道角度を考慮した軌道生成にも成功し、目的の進入角度を持った軌道を獲得できることが明らかとなった。さらに融合型軌道プランナを用いて、実際のミッションを模擬した6自由度飛行シミュレーションによる仮想リアルタイム誘導飛行実証を行った。シミュレーションの結果より、リアルタイムに軌道更新を行うことで目的地まで誘導飛行が可能であることを証明した。

小型無人飛行機を用いたリアルタイム誘導計算飛行実験

融合型軌道プランナの有効性を検証するため、主翼スパン約2メートルの小型無人飛行機を開発した。機体は、自己位置や姿勢を観測する各種センサを搭載しており、将来的に航法誘導制御システムにより自律飛行が可能となる。同機に研究用のコンピュータを搭載し、オンボードでのリアルタイム誘導計算実験を実施した。障害物は、ソフトウ

エア内で仮想で配置し、3種類の障害物パターンで軌道生成を行った。実験の結果より、機体の現在位置および状態量を考慮して、目的地点までの障害物回避軌道を生成できることが確認できた。さらに、計算速度についても平均して約 0.3 秒とリアルタイム誘導として十分に高速なアルゴリズムであることが実証された。

まとめ

本研究の独自性は、軌道計算の高速性と厳密な障害物回避性を併せ持つことである。従来の誘導手法では、障害物回避軌道を生成するために最適化計算を行うものが多く、軌道計算の高速性が課題となった。また、高速に障害物回避経路が得られる手法では、幾何的に経路が表現されることにより障害物の回避性が保障されなかった。そこで本研究では、最適性を追わないことで計算の高速化を図り、さらにモデルのダイナミクスに基づく軌道計算により高い障害物回避性を獲得する手法を提案した。

また、本手法における新規性は、障害物回避経路を探索するアルゴリズム（ランダム探索）に軌道計算の処理（力学フィルタ）を融合させたことにある。自走ロボットのような高い機動力を持つモデルは、ランダム探索で得られる経路上を容易に走行できるが、航空宇宙機は、急激な加減速や旋回が困難であるためランダム探索の経路に従うことは難しい。そこで本手法では、目標値をランダムに与え、その目標に向かって運動モデルを追従させる着想により、ランダムに探索しながら飛行可能な軌道を描くことを可能にした。

第 1 章 序言

1.1 研究背景と目的

本論文では、九州工業大学が参画している 2 つの無人宇宙機開発プロジェクトへの適用を目指した自律誘導アルゴリズムを研究テーマとして取り扱う。本節では、はじめに一般的な無人航空機の概要や歴史的背景について触れ、主題である火星探査航空機および有翼ロケット実験機の両開発プロジェクトについて発足背景と概要を説明する。

1.1.1 無人航空機

1.1.1.1 概要

人が搭乗することなく、遠隔操縦または自動により飛行する航空機を一般に無人航空機 (UAV: Unmanned Aerial Vehicle または Drone) と呼ぶ。UAV は、人間が不得意として回避すべき、「3D 任務 (Dull, Dangerous, Dirty)」への適用に非常に適していると言われる^[1]。Dull は、「人間の疲労が蓄積する長時間の単調な任務」、Dangerous は、「人命を危険に晒す危険な任務」、Dirty は、「汚染された環境での (汚い) 任務」などを意味する。このような観点から、主に軍需を背景に研究開発が進められてきた。

1.1.1.2 発展の歴史

第一次世界大戦中の 1918 年 3 月、米海軍のカーチス N-9 練習機を改造し無線操縦化した「スペリー空中魚雷 (Sperry Aerial Torpedo)^[2] (Fig. 1.1.1)」が初飛行に成功した。これが UAV の歴史の始まりと言われている^[3]。UAV が軍事戦略に本格的に投入されたのは、1960 年代のベトナム戦争である。当時、最も広く任務に用いられた Firebee^[4] (Fig. 1.1.2) と呼ばれる機体は、当初攻撃の対象となるために開発された無人標的機であったが、多くの改修が行われ、カメラ映像による偵察、広告の散布、地対空ミサイル用レーダーの探知などを行う無人偵察機として発展し、様々な任務を果たした。ベトナム戦争における UAV の実戦投入に成功したことを発端に、主にアメリカとイスラエルにおいて UAV の研究開発が加速し、1980 年代後半に開発された RQ-2 Pioneer^[5] (Fig. 1.1.3) は、小型で安価な機体として湾岸戦争で活躍した。さらに、2000 年代に入るとアフガニスタン紛争やイラク戦争において、高高度長時間滞空型の無人偵察機 RQ-4 Global Hawk^[6] (Fig. 1.1.4) や、レーザー誘導対戦車ミサイルを搭載した無人戦闘機 MQ-1 Predator^[7] (Fig. 1.1.5) が実戦投入され、偵察、諜報、攻撃を目的とする UAV

は、現在戦略体系の重要な位置付けとなっている。

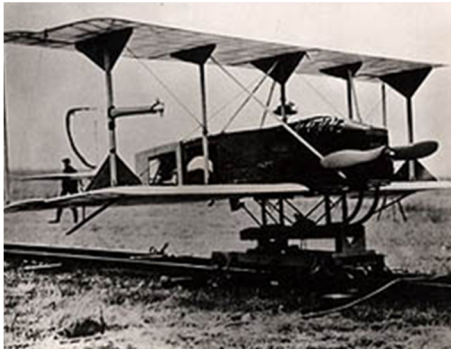


Fig. 1.1.1 Sperry Aerial Torpedo
(Copyright ©Cradle of Aviation Museum)



Fig. 1.1.2 Firebee
(Copyright ©Western Museum of Flight)



Fig. 1.1.3 RQ-2 Pioneer
(Copyright ©Smithsonian National Air
and Space Museum)



Fig. 1.1.4 RQ-4 Global Hawk
(Copyright ©U.S. Air Force)



Fig. 1.1.5 MQ-1 Predator
(Copyright ©U.S. Air Force)

このように従来は、軍需を背景に急速に発展してきた UAV であるが、3D 任務が想定される事例は非軍事用途でも多く、近年は、様々な分野での活用が期待されている。日本においてもこれまでに非軍事目的として研究開発が行われてきた。1987 年に農林水産省から委託を受けたヤマハ発動機が世界で初めて農薬散布を目的とする無人回転

翼 FAZER (Fig. 1.1.6) を開発し、現在では、国内の水稲耕作地の約 36%で同機が農薬散布を行っていると言われる^[8]。また、研究ベースの実験機としても UAV の活用が進んでおり、近年では、宇宙航空研究開発機構 (JAXA) が進める低ソニックブーム設計概念実証機 D-SEND (Drop test for Simplified Evaluation of Non-symmetrically Distributed sonic boom) (Fig. 1.1.7) が飛行実験を行って多くの研究成果を上げている^[9]。



Fig. 1.1.6 FAZER
(Copyright ©YAMAHA)



Fig. 1.1.7 D-SEND
(Copyright ©JAXA)

1.1.1.3 市場の拡大と課題

世界の防衛産業の発展状況に関する多くの情報源に基づいて、調査レポートを多数発信している Strategic Defence Intelligence が発行した報告書 **The Global UAV Payload Market 2012-2022** によると、UAV の 2012 年時点の市場規模（搭載機器を含む）はおよそ 39 億米ドル（2015 年 8 月現在で約 4850 億円）と算定され、2022 年までに 60 億米ドル（約 7460 億円）まで拡大すると予測されている^[10]。その中で、今後の成長分野と位置づけられているものは、持続的監視、敵防空網制圧と破壊、通信の中継、戦闘および搜索救難などの従来の軍需としての市場拡大であり、さらにもう一つの重要な促進要素として国土安全保障、災害管理、ならびに国境監視などの用途で UAV の導入が増加することなどが挙げられている。また、2013 年 11 月には、通販、Web サービス会社である Amazon.com, Inc. が Amazon Prime Air と呼ばれる小型 UAV（マルチローター型）を用いた配送サービスの構想を発表して大きな話題となった^[11]。この他にも映像の撮影（空撮）やプラントの保守点検等、今後 UAV の市場は急激に拡大することが期待されている^[12]。

市場拡大の一方で、UAV の健全な活用を目指した法整備が急務となっている。アメ

リカでは、2015 年 1 月にホワイトハウスの敷地内に一般人が遠隔操縦する機体が落下して問題となった^[13]。日本でも同年 4 月に、首相官邸の屋上で墜落した機体が発見された^[14]。小型の機体は、安価に入手でき誰でも容易に扱えるため、近年操縦を誤って落下させたことによる人との接触や、航空法への抵触、プライバシーの侵害などの事例が多発している。UAV 先進国であるアメリカは、一連の流れを受けて、米連邦航空局 (FAA: Federal Aviation Administration) が近代化及び改革法を 2012 年 2 月に成立させた。FAA は、UAV を公用無人航空機、民間無人航空機、趣味娯楽用模型機に大別しルール作りを進めている。商用利用の具体案として機体の重量、速度、高度などに制約を設け、オペレーションに関する規定を設けた^[15]。日本政府もまた、2015 年 6 月に小型無人機を対象とする航空法の改正案を国会に提出し法整備に乗り出した^[16]。改正案では、繁華街や住宅密集地などを飛行禁止区域と定め免許制度を導入するなど、安全な運行に向けた具体的な提案が多数なされた。

このように、今後 UAV の市場は拡大し法整備が急ピッチで進められ、活用範囲がますます増加していくものと考えられる。そのため、機体システムに関しても衝突や墜落をしない仕組みなど、安全な運用に向けたシステム開発に対する機運が急激に高まっている。

1.1.2 火星探査航空機開発プロジェクト

1.1.2.1 火星探査の歴史

1973 年, ソビエト連邦が打上げた Mars3 号^[17] (Fig. 1.1.8) が世界で初めて火星への着陸に成功したが, 着陸後約 20 秒で通信が途絶した. 本格的な火星探査は, アメリカの Viking 計画^[18]によって実施された. Viking は, 軌道周回機(オービタ) (Fig. 1.1.9) と着陸機(ランダー) (Fig. 1.1.10) で構成され, 軌道上と地上の両方からの画像撮影や, 標本採取などにより地質調査や生命の痕跡の調査などが行われたが, 生命体の存在は確認できなかった. その後は, アメリカ航空宇宙局 NASA (National Aeronautics and Space Administration), ロシア連邦宇宙機関 FSA (Russian Federal Space Agency), 欧州宇宙機関 ESA (European Space Agency) など世界中で火星探査機が打上げられてきた. 1996 年に NASA により火星に送られた Mars Global Surveyor^[19] (Fig. 1.1.11) は, 軌道上からの画像撮影, 赤外線レーザを用いた高精度な地表高度データの取得および地場観測などの膨大なデータを取得し, 火星の 3 次元マップや残留地場の強さを表すマップの作成など火星探査を大きく前進させた. 同時期にペアミッション機として打上げられた Mars Pathfinder^[20] (Fig. 1.1.12) は, ローバーと呼ばれる探査車であり, 火星地表面を走行することができた. 同探査により, 詳細な地表画像の撮影や大気, 岩石の成分分析が行われ, 太古の火星には, 水が存在したことが明らかとなった.

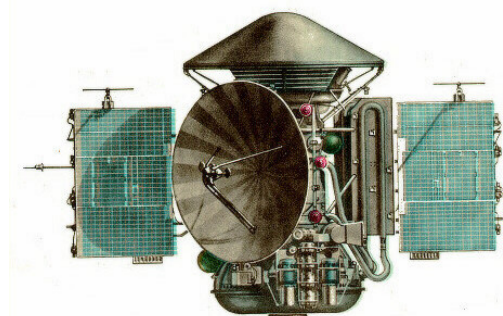


Fig. 1.1.8 Mars3
(Copyright ©NASA)

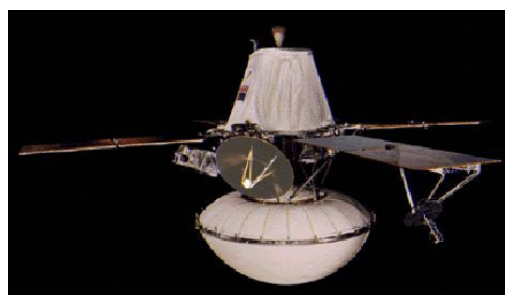


Fig. 1.1.9 Viking Orbiter
(Copyright ©NASA)

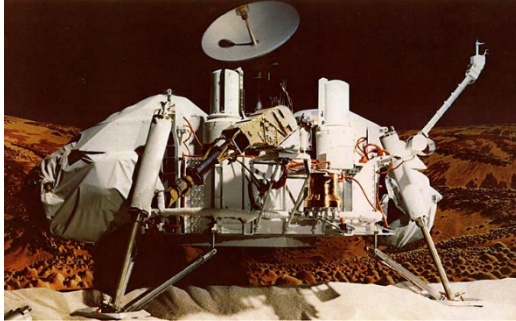


Fig. 1.1.10 Viking Lander
(Copyright ©NASA)



Fig. 1.1.11 Mars Global Surveyor
(Copyright ©NASA)

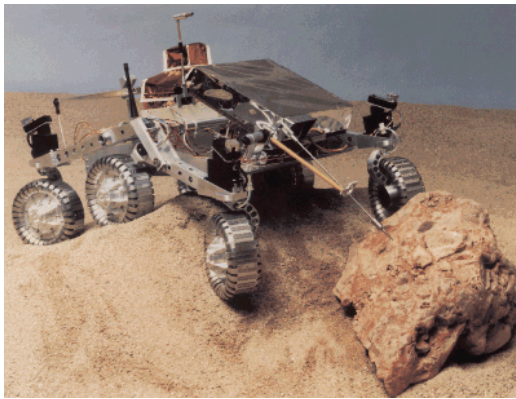


Fig. 1.1.12 Mars Pathfinder
(Copyright ©NASA)

日本では、1998 年に文部科学省宇宙科学研究所（JAXA: Japan Aerospace eXploration Agency の ISAS: Institute of Space and Astronautical Science）によって火星探査機「のぞみ」（計画名 PLANET-B）^[21]（Fig. 1.1.13）が打上げられた。同機では、火星の上層大気観測、地場観測などを主目的としていたが、火星の周回軌道の投入に失敗している。その後「のぞみ」が達成し得なかったミッションへの再挑戦や、火星地表面で科学的な研究成果を目指したい研究者らが集い、2008 年に ISAS 宇宙理学委員会、JSPEC: Japan SPace Exploration Center 宇宙探査委員会の承認を経て「次期火星探査ワーキンググループ（MELOS : Mars Exploration with Lander-Orbiter Synergy）」（Fig. 1.1.14）が発足した^[22]。この計画名と同名の島、Melos 島は「ミロのヴィーナス」が発見された地であり、Venus と Mars は神話の中で恋人同士とされる。MELOS はそのような火星への強い思いを込めたミッション名として付けられた^[23]。



Fig. 1.1.13 PLANET-B
(Copyright ©JAXA)

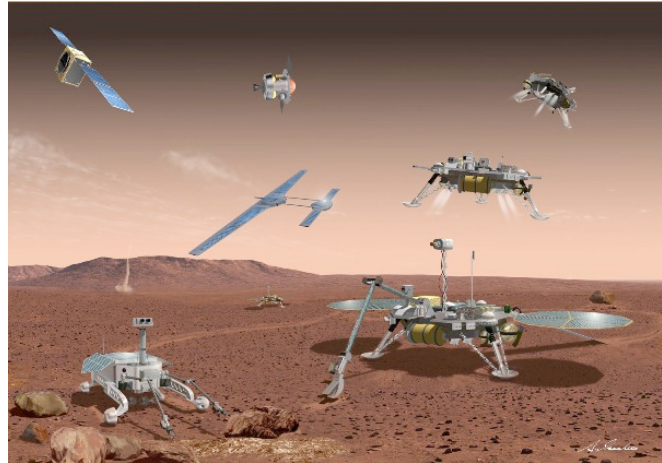


Fig. 1.1.14 MELOS Mission
(Copyright ©JAXA)

2020 年代の打上げを目指す MELOS1 では、オービタとランダーを用いた大気観測、気象観測、地表観測、内部構造観測、生命探査などが目標として掲げられている。従来の火星探査は、主に周回衛星とローバーを用いて行われていたが、この MELOS 計画では、UAV を探査機として用いるという革新的な探査手法が提案された。周回衛星を用いた高高度からの探査では、得られるデータの精度に限界があることや、地層断面などの観測が難しいなどの問題がある。また、ローバーを用いた探査では、移動範囲に大きな制約がある。例えば、先の Mars Pathfinder は、移動速度が秒速 1cm 程度と低速であり、着陸地点（着陸機）から半径 500m 以内のみ走行可能という制約があるため広範囲にわたる探査は難しい。また、火星には高さ約 25km にも及ぶ太陽系最大の火山とされるオリンポス山 (Fig. 1.1.15) や、深さ 5~7km 長さ約 4000km にも及ぶ太陽系最大の峡谷とされるマリネリス峡谷 (Fig. 1.1.16) が存在するが、起伏の激しい土地をローバーで探索することはリスクが高い。このような場合に大きなメリットをもたらすのが UAV による飛行探査であり、土地の起伏の影響を受けることなく、高速に広範囲の詳細なデータ取得が行えることが期待されている。NASA では、過去に高度約 1.6km、航続距離約 1000km の飛行が可能な火星探査航空機 ARES (Aerial Regional-scale Environmental Survey of Mars) ^[24] (Fig. 1.1.17) の検討が進められていたが、実現には至っていない。

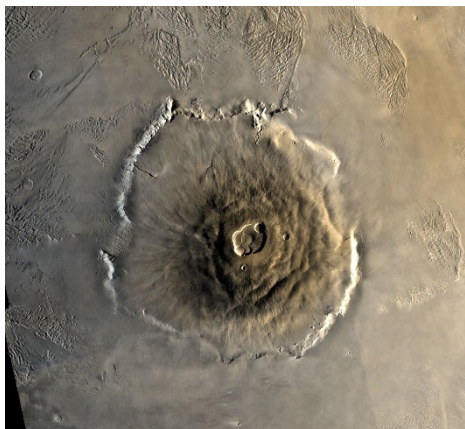


Fig. 1.1.15 Olympus Mons
(Copyright ©NASA)



Fig. 1.1.16 Valles Marineris
(Copyright ©DLA/ESA)

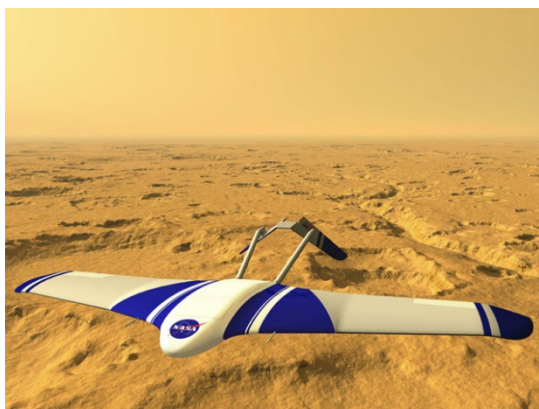


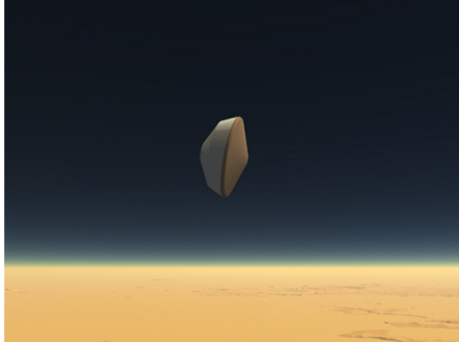
Fig. 1.1.17 ARES
(Copyright ©NASA)

1.1.2.2 プロジェクトの概要

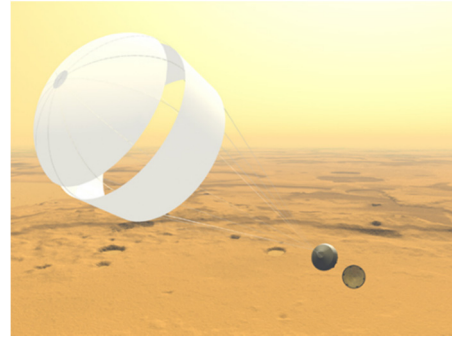
UAV を使った探査に関して、日本でも本格的な検討を行うため、2010 年に MELOS 計画の一環として火星探査航空機ワーキンググループ（以下、火星探査航空機 WG）が発足した。同 WG では、これまでに飛行機型の固定翼機、ヘリコプター型の回転翼機、トンボのような羽ばたき翼機、気球などの飛行移動体が検討されてきた。その中でも、固定翼機は、広範囲の探査が短時間で行え、制御性も高いことから優位であると考えられる。固定翼機の研究開発には、JAXA/ISAS をはじめ、九州工業大学、東北大学、金沢大学、金沢工業大学など複数の大学が、この世界初の偉業を実現するために活発な議論を交わしてきた。その中で九州工業大学は、システム設計、空力設計、構造設計を主に担当している。火星探査航空機は、火星大気エントリカプセルに格納されてロケット

で火星まで搬送された後、高度 2～3km 地点でエントリカプセルが開放されて機体が放出される。機体は、折り畳まれた主翼と胴体を展開した後、航続距離 100km 程度の飛行探査ミッションを実行する。この展開方法は、NASA の ARES により検討されていたもの (Fig. 1.1.18) とほぼ同じ方式である。ただし推進系には、2 発もしくは 4 発のプロペラを装備し着陸装置は備えていない。

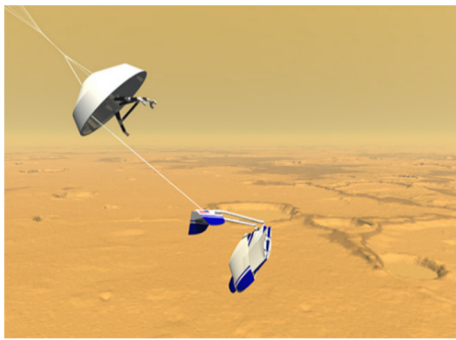
WG では、火星探査航空機の空力データを取得し、火星環境における飛行の実現性を評価するための高高度飛行試験の実施を計画している。同試験では、火星と同程度のレイノルズ数環境を再現することができる高度 36km からの滑空飛行を行い、ピトー管を用いたエアデータ、慣性センサによる姿勢データ、GPS を用いた位置データの取得により機体の空力データを解析する。2013 年 6 月に北海道帯広郡大樹町の大気航空宇宙実験場にて第 1 回目の高高度飛行試験に臨んだが、機体を上空へ運ぶための大気球システムに不具合が発生し中止となった (Fig. 1.1.19)。現在は、次の高高度飛行試験に向けて機体システムの改善などを行っている。



(a) Atmospheric Entry Phase



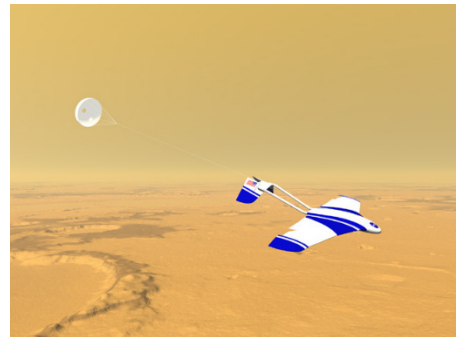
(b) Parachute Deployment Phase



(c) Aircraft Ejection Phase



(d) Aircraft Deployment Phase



(e) Pullout Phase



(f) Exploration Phase

Fig. 1.1.18 ARES Mission Sequence
(Copyright ©NASA)



(a) Balloon



(b) Flight Test Vehicle

Fig. 1.1.19 High-Altitude Flight Test Using Balloon
(Copyright ©JAXA)

1.1.2.3 機体システムの課題

一般的に火星の環境は、地球と似ていると言われ生命存在の可能性が話題にのぼることが多い。しかし、航空機を飛ばす場合には、両者の環境差異の影響は極めて大きい。ここでは、航空機の設計観点から飛行環境の違いについて述べる。Table 1.1.1 に火星と地球の環境を比較した表を示す。

Table 1.1.1 Planetary Environment of Mars and Earth

		Mars		Earth	
Gravity	[m/s ²]	3.66		9.81	
Gas Composition	[%]	CO ₂	: 95.3	N ₂	: 78.1
		N ₂	: 2.7	O ₂	: 20.9
		Ar	: 1.6	Ar	: 0.9
		O ₂	: 0.13	CO ₂	: 0.04
Atmospheric Pressure	[kPa]	0.6~1		101.3	
Density	[kg/m ³]	0.0118		1.225	
Sound Speed	[m/s]	258		340	

航空機の飛行性能に大きな影響を与える要因は、重力と大気密度であり、地球と比較して重力は地球の約 1/3、大気密度は約 1/100 である。翼に発生する揚力は、大気密度に比例する。航空機が火星を地球と同じ対気速度で飛ぶ場合、重力は 1/3 になるが揚力は 1/100 に低下するため、従来の航空機の設計では火星環境を飛ぶことが難しくなる。また機体は、ロケットのペイロードフェアリング内に格納し運搬することを想定しているため、寸法や重量にも大きな制約があり、従来よりも軽量かつ高い揚抗比を持つ高性能な航空機の設計が要求される。これらの課題を定量的に評価するため、大山らは、火星探査航空機の多目的設計探査を行った^[25]。同論文では、航続距離 100km 程度を飛行し、かつ機体の胴体と主翼を折りたたむことで直径 1m 程度の円柱に収めることを設計要求として、(1)必要カプセル直径の最小化、(2)航続距離の最大化、(3)巡航速度の最小化、(4)巡航速度と最小速度の差の最大化の 4 つを評価項目として最適化が行われてい

る。その結果、当初の性能推算式を用いた最適設計解では、要求を満たすことはできなかった。そこで、性能推算式に以下のような制約を付加すると設計要求を満たす解が得られることが分かった。

- i) モータ質量を 40%減少させる
- ii) 主翼質量を 30%減少させる
- iii) バッテリのエネルギー密度を 30%増加させる
- iv) 揚抗比を 20%増加させる

以下では、九州工業大学が主に研究を担当する ii)および iv)に関して研究状況の概要を記載する。

(1) 主翼質量を 30%減少させる

これまでに様々な火星探査航空機用の超軽量翼構造の研究開発が行われてきた^[26]。上記の設計要求を満たす数値目標は、面密度 389g/m^2 であり、マグネシウム削り出しの桁にポリイミドフィルムを融着して外皮とした翼構造や、炭素繊維強化プラスチック (CFRP : Carbon Fiber Reinforced Plastic) 製の桁に発泡剤外皮を用いた翼構造について試作評価が行われた。その結果、マグネシウムとポリイミドフィルムを用いた翼は、面密度 450g/m^2 となり、更なる軽量化を目指すためにクマダイ合金とマグネシウムリチウム合金を構造に用いる案も提案されている。また、CFRP と発泡材を用いた翼構造の面密度は、 486g/m^2 となった。重量増加の原因は、発泡剤の比重や接着剤の使用量の増加によるものであるとされているが、発泡材には圧力変化による発泡構造の崩壊などの問題もあることが分かった。そこで、用いる発泡材の発泡倍率の変更や主翼外皮も CFRP で製作するオールコンポジット翼構造も試作検討が進められている (Fig. 1.1.20)。今後 WG では、これらの構造案を比較評価しながら、火星探査航空機に適した主翼設計を模索していく。



Fig. 1.1.20 All Composite Wing Structure

(2) 揚抗比を 20%増加させる

地球上の一般的な航空機が飛行するレイノルズ数は、翼弦長を基準長として $10^6 \sim 10^7$ 程度である。それに対して火星探査航空機は、ペイロードとする寸法制約を考慮した際のレイノルズ数が 10^4 程度と見積もられ 2~3 桁ほど小さい。しかし、そのような低レイノルズ数領域における翼型の空力特性の知見は、極めて少なく翼型設計を試みる上で翼形状に対する流れ場と力の相関の解明が必須課題となる。そこで九州工業大学では、低レイノルズ数において「火星環境を模擬した変圧風洞試験」「流体画像解析（PIV：Particle Image Velocimetry）を用いた翼周りの流れ場の可視化」によって空力特性の解明を試みた。Fig. 1.1.21 は、変圧風洞試験によって得られた NACA0012 翼のレイノルズ数と迎え角の変化による二次元翼の揚力係数の一例を示す。また、Fig. 1.1.22 には、PIV 試験で取得した画像の一例を示す。それらの結果を評価したところ、低レイノルズ数領域では、翼型形状と流れ場、および力計測に相関が見られることが分かった^[27]。また、佐々木らにより火星環境における翼型の最適化計算が行われた（Fig. 1.1.23）。遺伝的アルゴリズムを用いた多目的最適化により個体を成長させ、更にクラスタ分析を用いたデータマイニングにより翼形状の特徴を把握して分類することに成功した（Fig. 1.1.24）^[28]。Fig. 1.1.24 左は、良解の個体集団について各クラスタ毎に色付けしている。また Fig. 1.1.24 右では、各色のクラスタ毎に平均化された翼型を表現している。この最適化計算で得られた新しい翼型は、石井翼と比較して約 30%もの揚抗比の増加が見られ、火星探査航空機的设计要求 20%を大幅に上回る結果を得た。以上の研究成果を活用して、今後は、火星探査航空機に実用的な翼型を構造強度や制御性能などの観点から勘案して総合的に決定していく予定である。

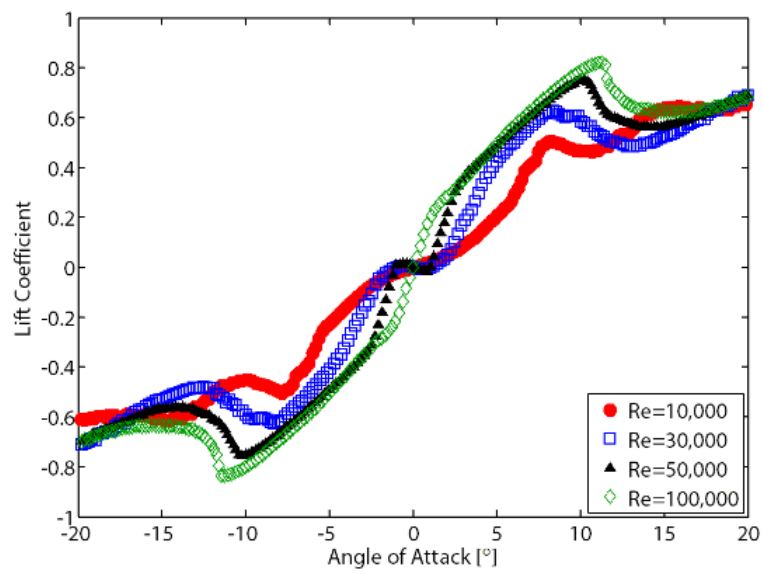


Fig. 1.1.21 Lift Coefficient of Two-Dimensional Airfoil

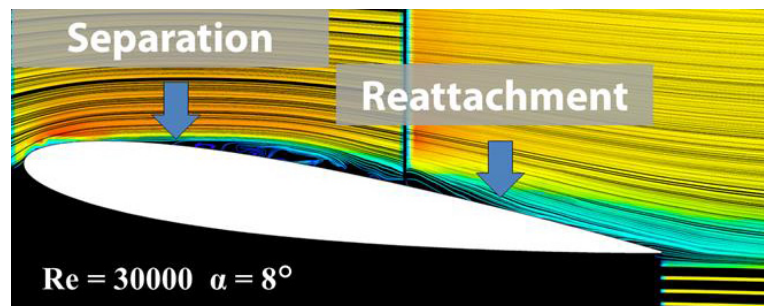


Fig. 1.1.22 PIV of NACA0012 Airfoil

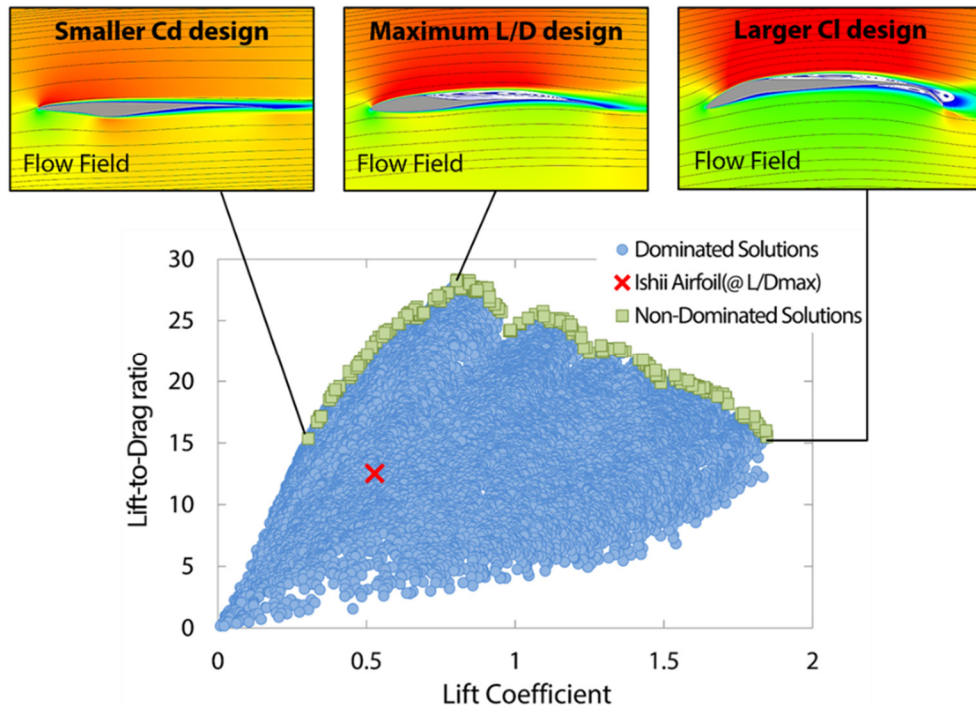


Fig. 1.1.23 Multi Objective Optimization of Mars Aircraft Airfoil

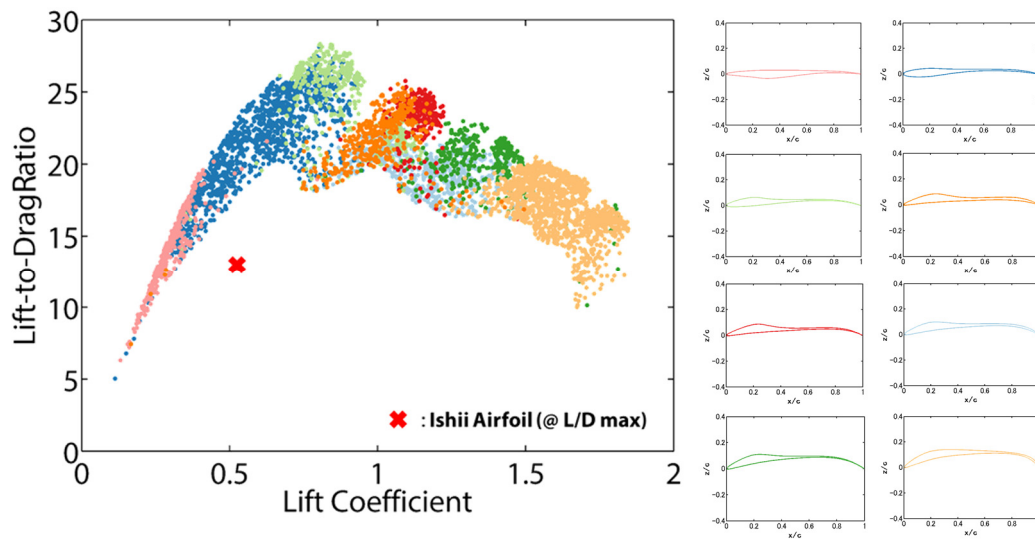


Fig. 1.1.24 Data Mining of Optimized Airfoil

1.1.2.4 自律航行システムの課題

火星で航空機を飛行させる場合、地球からパイロットが遠隔操縦することは、通信に時間がかかること（片道 5 分から 10 分程度）などから現実的ではない。従って、機体自身が判断して目的のミッションを行う自律航行システムが必須となる。しかし、火星では、地球のような GPS 衛星や磁気方位計が使えないこと、大気密度が低くエアデータの取得精度が低下することなどから航法誘導制御の観点からも極めて自律航行が困難な環境である。また、気象面でも悪条件が予想される。NASA の Mars Reconnaissance Orbiter の撮影画像からは、直径約 70m、高度約 20km に達する巨大な旋風が発生していることが確認されている^[29] (Fig. 1.1.25)。この旋風は、ダストデビル（砂塵嵐）と呼ばれ、地表のダスト粒子を巻き上げて移動する。火星大気は、熱容量が小さく温度変化が激しいため、地球に比べて活発な対流が生じる。これにより巨大なダストデビルが発生しやすい。探査車の観測では、ダストデビルの通過に伴う気圧の低下が 1 日に約 1 回は観測されており、惑星全体で大小様々なダストデビルが頻繁に発生していることが明らかとなっている^[30]。



Fig. 1.1.25 Dust Devil
(Copyright ©NASA/JPL/University of Arizona)

このように火星で航空機を飛行させる場合には、航法誘導制御に関しても様々な技術的課題が存在し、それらを解決しなければならない。火星探査航空機の自己位置推定アルゴリズムについて、カルマンフィルタによる画像処理と測位信号を用いた手法が研究されている^[31]。同論文では、良好な推定結果が得られているが、推定精度と処理時間はトレードオフの関係にあり、搭載できる CPU のスペックにも重量やサイズなどの制

約があるため今後精査が必要とされている．制御の面からも突風に強いロバストな制御器の開発が必要であり，航空機の乱気流事故の防止技術として JAXA で研究されているドップラーライダーを用いた突風応答軽減制御^[32]などの応用が考えられる．

1.1.2.5 本研究の目的

本研究では，火星探査航空機の自律誘導アルゴリズムについて検討する．火星探査航空機のミッションでは，航行距離約 100km の飛行探査を想定しているが，より詳細なデータ取得と画像撮影を行うためには，できるだけ低空で飛ぶ必要がある．そして，ミッションの一案として，マリネリス峡谷付近の探査が想定されている．マリネリス峡谷における飛行探査では，従来手法では困難であった地層などが観測できる可能性があり，極めて興味深い探査場所である．ただしマリネリス峡谷は，非常に複雑な地形をしているため，目的の観測地点を飛行するには，山々や岸壁を避けて飛ぶなければならない．この場合，事前に飛行経路を計画してオンボードコンピュータに組み込んでおくなどの対応が考えられるが，実際には，確実に計画通りの位置から飛行探査が開始されるという保障が無い．そこで，機体自身が現在地から観測地点までの飛行軌道を自身で判断する自律誘導アルゴリズムが必要となる．また，ダストデビルなどの影響で計画軌道から逸脱する可能性も考慮した場合，随時軌道を更新して修正するリアルタイム誘導を実現することが望ましく，これはミッションの成立性を大きく左右すると考えられる．そこで本研究では，多数の山々を避けて飛行するリアルタイム障害物回避軌道生成法を目的とした自律誘導アルゴリズムの構築を目指す．

1.1.3 有翼ロケット実験機

1.1.3.1 ロケット開発の歴史

30 年間で延べ 135 回の宇宙飛行を行った NASA のスペースシャトルが 2011 年についてその役目を終えた。スペースシャトルは、再使用をコンセプトとして実用化された世界初の有人宇宙輸送システムであり、これまでに多数の人工衛星の打上げ、宇宙環境を利用した科学実験や国際宇宙ステーション（ISS: International Space Station）の建設などを行ってきた。同機は、主にオービタ、外部燃料タンク、固体補助ロケットの三つの部分で構成される。そのうち有翼型のオービタは、滑空して帰還し、固体補助ロケットはパラシュートで回収され再整備して利用される部分再使用型の宇宙輸送システムである。計画を立ち上げた NASA の長官は、当初スペースシャトル計画は、機体を再利用することで打上げコストが 1 回あたり 1050 万ドル（約 12 億円）程度と発表していた。しかし、世界で最も複雑な機械と言われたこのシステムは、実際に再利用するためには膨大なメンテナンス費用と人員コストがかかり、さらにチャレンジャー号とコロンビア号の事故による設計変更などの影響を受けて、2007 年時点で 1 回の打上げコストは 10 億ドル（約 1200 億円）に達していた^[33]。そして、今後機体の老朽化により失敗のリスクが高まることや、膨大な運用費の確保が必要となることからスペースシャトル計画の終了が決定した。NASA は、現在スペースシャトルの後継としてカプセル型の有人宇宙船 Orion^[34]を Lockheed Martin と共同で開発し、打上げに成功した（Fig. 1.1.26）。それと同時に COTS（Commercial Orbital Transportation Services）と呼ばれる民間による国際宇宙ステーションへの商業軌道輸送サービス計画を立ち上げるなど、これまで国家戦略であった宇宙開発を民間企業にも委託する試みを進めている。

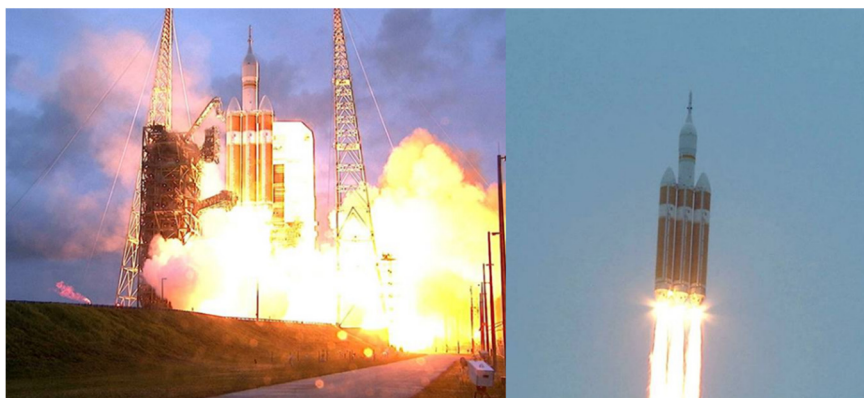


Fig. 1.1.26 Orion Launch Experiment
(Copyright ©NASA)

またアメリカでは、NASA による民間委託業務以外にも、民間企業独自の宇宙開発の試みが盛んに行われている。その代表例として、X Prize Foundation により有人弾道宇宙飛行を競うコンテスト Ansari X Prize が開催され、Scaled Composites 社が開発する SpaceShipOne^[35] が 2004 年に世界で初めて民間による高度 100km の宇宙空間到達と弾道飛行という有人宇宙飛行を実現させた (Fig. 1.1.27)。SpaceShipOne は、運搬用航空機 (White Knight) により高度 15km まで運ばれた後、切り離されてロケットエンジンを点火して高度 100km まで一気に上昇する。弾道飛行後に機体は、滑空飛行を行って目的の飛行場に着陸する。現在は、宇宙ビジネスを計画する Virgin Galactic が Scaled Composites からの技術提供を受けて SpaceShipTwo を開発し、各種試験を行っている。同社の計画では、年間 500 人を 1 人 25 万ドル (約 3000 万円) で宇宙へ送るとしている。現在は、弾道飛行の段階ではあるが、夢の完全再使用型宇宙輸送システムを実現させ、飛行機に乗るように手軽に宇宙へ行ける新しい時代の幕開けが目前まで迫っている。



Fig. 1.1.27 SpaceShipOne
(Copyright ©Scaled Composites)

日本でも、これまでに無人の再使用型宇宙往還機の研究開発が行われてきた。1980 年代後半から、宇宙往還技術試験機 HOPE-X (H-II Orbiting Plane Experimental) の開発が JAXA の前身である宇宙開発事業団 (NASDA : National Space Development Agency of Japan) と航空宇宙技術研究所 (NAL : National Aerospace Laboratory of Japan) によって行われた^[36]。並行して開発を進める H-II ロケットのペイロードフェアリング部分を HOPE-X に置き換えて軌道上に打上げ、軌道を周回した後滑空飛行により地上で回収する構想を目指した。宇宙開発委員会で 1996 年 1 月 24 日に改訂された「宇宙開発政策大綱、第 2 章、宇宙開発の重要活動」^[37]より引用すると、HOPE-X の目的は、「従来のロケット技術による輸送コストと比べ、大幅なコスト低減が可能な再

使用型輸送系の技術基盤育成の一環として、HOPE-X の開発を進め、飛行実験を実施する。これにより、無人有翼往還機の主要技術の確立を図るとともに、将来の再使用型輸送機に必要な技術蓄積を図る。」と定められている。同プロジェクトは、軌道再突入実験機（OREX: Orbital Reentry EXperiment）^[38]（Fig. 1.1.28）、極超音速飛行実験機（HYFLEX: HYpersonic FLight EXperiment）^[39]（Fig. 1.1.29）、小型自動着陸実験（ALFLEX: Automatic Landing FLight EXperiment）^[40]（Fig. 1.1.30）により様々な実験を実施して空力、誘導制御などで多くの成果を上げたが、2000 年 8 月に計画が凍結された。想定していた開発費を大幅に超過したことや、開発最盛期における度重なる H-II ロケットの打上げ失敗、宇宙予算の削減などが凍結の主な理由と言われている^[41]。



Fig. 1.1.28 OREX
(Copyright ©JAXA)



Fig. 1.1.29 HYFLEX
(Copyright ©JAXA)



Fig. 1.1.30 ALFLEX
(Copyright ©JAXA)

時をほぼ同じくして、文部省宇宙科学研究所 ISAS（現 JAXA 宇宙科学研究所）では、1982 年に有翼飛翔体ワーキンググループが発足し、1984 年に有翼式宇宙往還機 HIMES（Highly Maneuverable Experimental Space vehicle）の構想が提案された

[42]. ISAS は、有翼飛翔体のみで上昇を行い、宇宙空間に到達し弾道飛行をした後、地上に滑空帰還する単段式の完全再使用型宇宙輸送システムの実現、つまり「航空機のようにエンジンに燃料を補給するだけで宇宙と地球を往復できるロケット」を目指した。HOPE-X では、地球周回軌道への打上げを想定していたのに対し HIMES では、弾道飛行としている点で違いがある。同プロジェクトでは、完全再使用型のロケット飛翔体に関連する技術習得を目的として、大気圏再突入飛行における空力、耐熱、飛行制御に関連する技術や、無人飛翔体の誘導、自動着陸に関連する知見を獲得するため数多くの実験が行われた。1986 年、1987 年には、宇宙研能代実験場にて高度 1000m, 2000m でヘリコプタから飛翔体を切り離し飛行させる低速滑空飛行試験が行われた (Fig. 1.1.31)。計 5 回の飛行実験が実施され、空力特性、誘導制御特性などに関して重要な知見が得られた。低速滑空飛行試験に成功したため、次のステップとして大気圏への再突入時の飛行特性の検証を目的とする再突入実験が行われた。高度 18km まで上げた気球から、機体長 2m の実験機を打ち上げるロックーン方式が採用され、1 回目の実験が 1988 年 9 月に実施された (Fig. 1.1.32)。しかし、高度 18km に到達した時点で気球が突如破裂し、機体は墜落した^[43]。約 3 年後となる 1992 年 2 月に、気球に破裂防止策を施し 2 回目の飛行実験が行われた。午前 8 時に機体は、気球から分離されブースターへ点火、その後高度 67km まで上昇し、ガススラスタを用いた頂点付近の姿勢制御、および空力舵面を用いた姿勢制御という当初の目的を計画通りに果たした^[44]。さらに、その後の開発計画として着陸誘導実験が計画されていたが、様々な要因により再突入実験の完遂をもって実機の開発計画は中止となった。

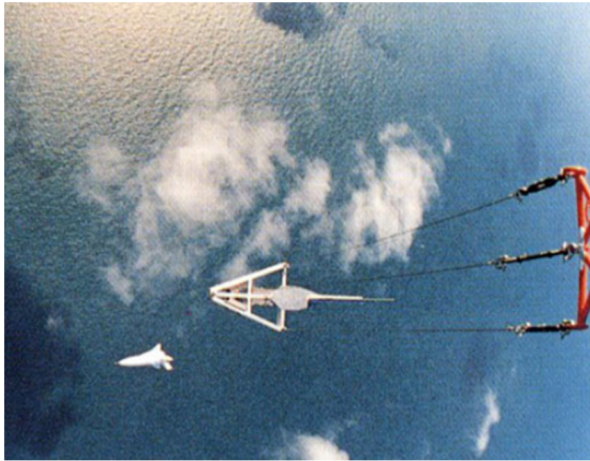


Fig. 1.1.31 Gliding Test
(Copyright ©JAXA)



Fig. 1.1.32 Reentry Flight Test
(Copyright ©JAXA)

このように日本でも、様々なチャレンジが行われ多くの成果を挙げてきたが、未だに再使用型宇宙輸送システムの実現には至っていない。これは技術的な課題以外にも、アメリカと比較して10分の1程度しかない国の宇宙開発関連予算などの影響が大きいと考えられる。そこで、日本でも国策に依らずに宇宙開発を継続的に進め、宇宙開発コストの大幅な削減を図るため、民間企業が主導するロケット開発の機運が高まっている。一例として、堀江貴文氏が創始者となり安価な超小型衛星打上げ用のロケットの実現を目指すSNS株式会社は、NPO 法人北海道宇宙科学技術創成センター（HASTIC: Hokkaido Aerospace Science and Technology Incubation Center）と協力して機体を開発し、高度約6kmに到達する打上げ実験に成功している^[45]。

1.1.3.2 プロジェクトの概要

九州工業大学では、完全再使用型宇宙往還システムの実現を目指して、2005年に国際航空連盟（FAI: Federation Aeronautique Internationale）によって定められた宇宙との境界とされる高度100kmに到達し、弾道飛行後に地上へ滑空帰還する有翼ロケット実験機WIRES（Winged REusable Sounding rocket）の研究開発を開始した。本プロジェクトは、先に述べた有翼式宇宙往還機HIMESのコンセプトを引き継いでおり（滑走路への着陸を除く）、現在の機体外形はHIMESと同じである。WIRESのサブオービタルミッションフェーズとしては、空力舵面およびエンジンジンバル機構を用いた上昇中の姿勢制御、高層希薄大気中のガスジェットスラスタを用いた再突入姿勢制御、大気圏再突入後の滑空誘導制御飛行およびパラシュートとエアバッグを用いた機体の

軟着陸回収を想定している (Fig. 1.1.33). HIMES では, 高高度滑空試験など帰還フェーズの試験を優先的に実施してきたが, 有翼型のように大きな主翼をもつ非対称形状の場合, 上昇中にも大きな揚力が発生することや風の影響を受けやすいことなど, 計画軌道通りに上昇させることが極めて困難となる. そこで, 本プロジェクトではロケット本来のミッションフェーズを踏まえて, 上昇フェーズから技術的な課題に挑戦し, 実験を行っている.

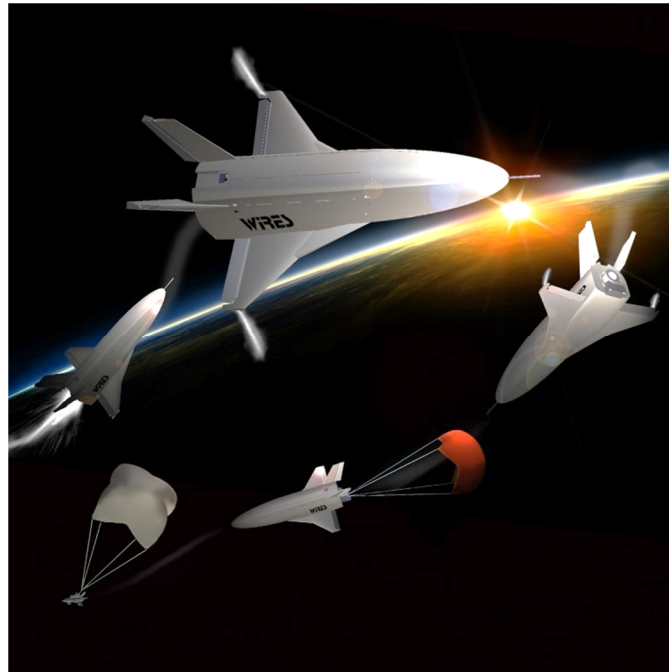


Fig. 1.1.33 Artist Concept of WIREs

有翼ロケットプロジェクトは, 九州工業大学機械知能工学研究系の宇宙システム研究室および宇宙クラブが中心となり, JAXA やメーカを交えた産官学連携プロジェクトである. Table 1.1.2 には, 2015 年 12 月現在プロジェクトに参加しているメンバーを示す. 多数の機関と連携しながら設計を進め, 週に一度のプロジェクト会議では, 九州工業大学のメンバーが進捗把握と課題に対する議論を行っている. また, 月に一度の合同設計会議では, プロジェクトに参加する関係機関を招へいし, 設計に関する詳細な打合せを行っている (Fig. 1.1.34). このように意思統一を行い, チーム一丸となって夢の実現を目指している.

Table 1.1.2 Project Team

九州工業大学	全機システム設計, 試験運用
JAXA	推進システム設計, 安全審査
(株)IHI	推進システム設計
(株)IHIEアロスペース	推進システム設計
川崎重工業(株)	全機システム設計, 構造設計
東レカーボンマジック(株)	構造設計
中国工業(株)	タンク設計
(株)ゼノクロス	電装設計
PDエアロスペース(株)	技術, 運営コンサルティング



Fig. 1.1.34 Airframe and Engine Collaborative Design Meeting

1.1.3.3 開発状況

開発計画と達成目標を Fig. 1.1.35 に示す。宇宙空間への到達は、現在のところ 2020 年を想定している。2005 年の研究開始当初は、市販の小型モデルロケットで実験を繰り返し、ロケット開発の基礎技術を蓄えた (Fig. 1.1.36 (a))。そして、機体の軽量化を目的とする複合材を用いた構造製作や、地上で模擬飛行を行えるハードウェアインザループシミュレータの開発などに着手し、2009 年までに全長 1.1m の小型有翼ロケット実験機 WIRES#011 による飛行実験が 5 回行われ、上昇中の姿勢制御技術の実証に成功した (Fig. 1.1.36 (b))。その後、全長 1.7m となる円筒型ロケット実験機 WIRES#012 を開発した。WIRES#012 は、同規模の有翼型ロケット実験機 WIRES#014 の開発に向けて、二段式パラシュートとエアバッグを用いた回収システムおよび非常系システムの技術実証を目的とした機体であり、3 回の実験の末に、2011 年に機体の軟着陸を成功させた (Fig. 1.1.36 (c))。そして 2013 年には、上昇および滑空時の空力舵面を用いた姿勢制御による航法誘導制御システムの技術実証を目的とする WIRES#014-1 の飛行実験を実施した。エンジン点火後、高度約 600m まで上昇したものの、エアデータシステムの動作不良や制御システムの不良などにより滑空への姿勢遷移に失敗し、機体は墜落した (Fig. 1.1.36 (d))。しかし、テレメトリ情報と機体に搭載したメモリによりほぼ全てのデータ取得に成功した。原因究明の後、打上げ性能の向上を目的として機体構造 (WIRES#014-2) を東レカーボンマジック㈱と共同製作して更なる軽量化を図り地上燃焼試験に臨んだ (2014 年) が、エンジンノズルの脱落という不具合により機体を焼損した (Fig. 1.1.36 (e))。そして、エンジンの不具合について CT スキャンによる画像解析やノズルの補強などの十分な対策を講じて WIRES#014-3 を再製作し、2015 年 7 月に地上燃焼試験に成功した。その後、ハードウェアインザループシミュレーションにより制御性能の向上を行い、2015 年 11 月 20 日に WIRES#014-3 の飛行実験を実施した。飛行高度約 1000m まで上昇し、その後滑空飛行への遷移および回収機構による軟着陸を成功させた (Fig. 1.1.36 (f))。今後、更に航法誘導制御システムに改良を重ねて WIRES#014 の飛行実験を複数回行う予定である。

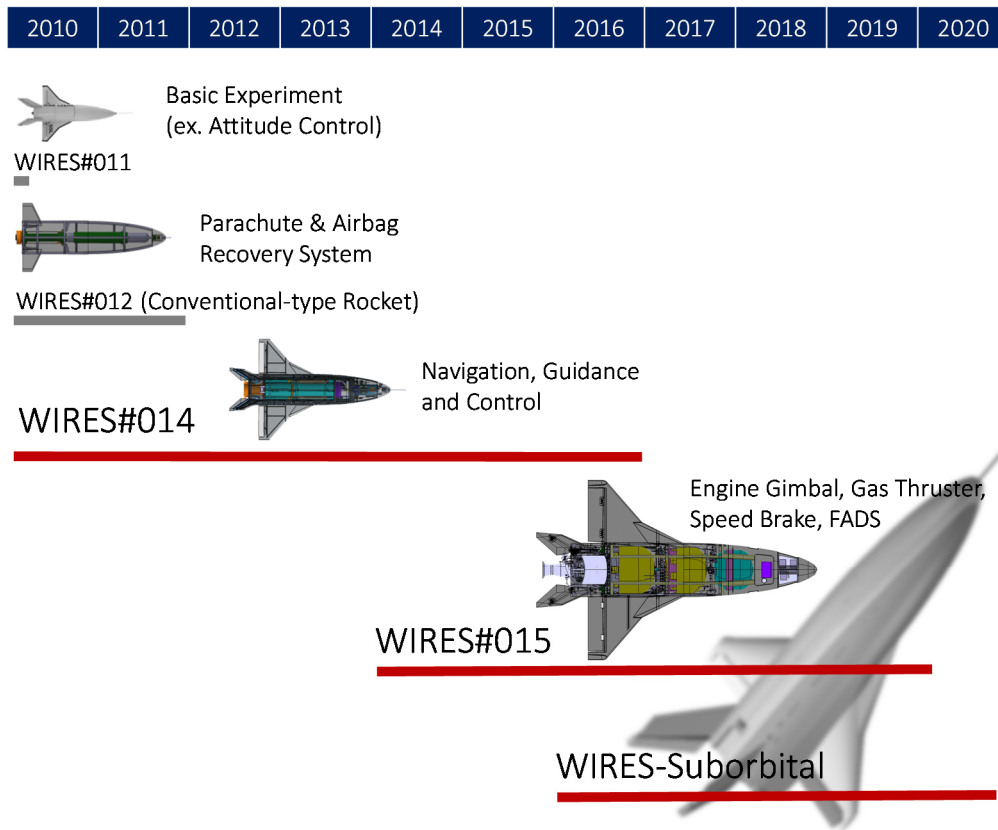


Fig. 1.1.35 WIRES Development Plan



(a) Flight Test of Commercial Rocket



(b) Flight Test of WIRES#011

Fig. 1.1.36 Development History of WIRES Project
(to be continued)



(c) Flight Test of WIREs#012



(d) Flight Test of WIREs#014-1



(e) Combustion Test of WIREs#014-2

Fig. 1.1.36 Development History of WIREs Project
(to be continued)



(f) Flight Test of WIREs#014-3

Fig. 1.1.36 Development History of WIREs Project
(to be concluded)

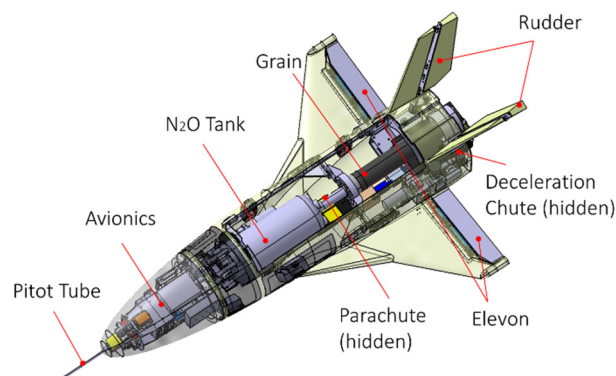
有翼ロケットプロジェクトでは、WIREs#014の開発と並行して、次期有翼ロケット実験機 WIREs#015の開発も進めている。WIREs#015では、宇宙空間到達に必要な技術基盤の実証を目的としており、エンジンジンバル機構を用いた上昇中の姿勢制御、軌道頂点付近のガスジェットスラストを用いた滑空姿勢遷移や、スピードブレーキを用いた高迎角飛行などの技術実証が新たな試みとなる。WIREs#015の開発は、JAXAやメーカーと連携して進めており、アメリカの大学とロケットエンジンや地上設備の開発、発射場の提案などで協力を得る予定である。さらに、宇宙空間に到達するサブオービタル機（WIREs-Suborbital）の概念設計も進めている。

1.1.3.4 実験システム

同プロジェクトでは、宇宙開発コストの劇的な削減と安価な宇宙旅行の実現を目指して、機体だけでなく発射台（ランチャー）、地上支援設備（GSE：Ground Support Equipment）、管制センタなど、宇宙往還に必要なシステム全体の開発を行っている。ここでは、WIREs#014に用いている実験システムの概要を述べる。

Fig. 1.1.37 (a)に WIREs#014の機器配置図を示す。機体は、エレボンとラダーの空

力舵面により姿勢が制御される。エレポンは、航空機のエレベータとエルロン両方の機能を果たす。ランチャーは、立ち上げると高さ約 7m となり (Fig. 1.1.37 (b)), 運搬時には、木箱に収納されてトレーラと連結して移動できるように設計されているため、ランチャーの設置による打上げ場所の制約を受けない。GSE は、射点で機体への燃料充填や点火などを行うシステムであり、ほぼ全ての機器が管制センタからの遠隔操作により作動する (Fig. 1.1.37 (c))。管制センタは、車両内に設置されているため移動が容易であることが特徴であり、GSE の充填状況や機体のデータをモニタリングし、さらに保安体制の状況など全ての情報が集まり、実験実施の判断とオペレーションが行われる (Fig. 1.1.37 (d))。このように、打上げ場所に制約を持たない移動式の実験システムと遠隔操作システムを構築して、高い安全性と利便性を兼ね備えた実験システムの実現を目指している。



(a) System Arrangement of WIREs#014



(b) Launcher



(c) Ground Support Equipment



(d) Control Center

Fig. 1.1.37 Flight Experiment Setup

1.1.3.5 本研究の目的

有翼ロケットは、主翼を用いて飛行するためミッション遂行中に不測の事態が起こった際には緊急着陸を試みることが可能となる。これは、安全性や経済性の観点から極めて優位な特徴である。しかし、従来の宇宙輸送システムは、事前の軌道計算に基づいて打上げられ、計画軌道からの微少誤差に対して修正を施す運用が行われてきた。このような誘導システムでは、計画軌道から大きく逸脱する飛行中断などの急な措置に対処することは困難であり、実際に緊急時には機体を爆破することで被害を最小限に食い止める方法が用いられてきた。

本研究では、有翼ロケットのメリットを最大限に活かすため、無人航空宇宙機に適用する自律誘導アルゴリズムを取り扱う。自律誘導アルゴリズムの実現には大きく二つの課題がある。一つは、ノミナル飛行の帰還時だけでなく緊急時にも対応可能とする高速な誘導計算を行うことである。特に緊急時には、飛行状態や環境が急激に変化することが想定されるため、実時間での誘導解の算出が重要となる。もう一つの重要な課題は、飛行禁止区域の回避である。目的の帰還地点に到達するまでに、都市部や国の重要施設などの上空を避けて飛行する誘導計画が求められる。アメリカ合衆国ニューメキシコ州には、世界初の宇宙商業利用を目的とする宇宙港 **Spaceport America** が開港されているが、周辺には軍事施設が点在しており、その地域は将来的に有翼ロケットのサブオービタル実験機を打ち上げる有力な候補地の一つとして検討されている。本研究では、これらの要求を満たす飛行禁止区域を避けて目的地まで帰還する自律誘導アルゴリズムの構築を目指す。

1.2 自律誘導アルゴリズムの先行研究

航空宇宙機の自律制御を扱う研究は、世界的に多く行われているが、自律誘導を含めた自律航行の研究例は、比較的少ない。従来の人間が操縦する航空機において制御の発展は、パイロットの操縦負担を大幅に軽減するなど直接的な実利が大きい。一方で誘導は、高高度飛行の際に障害となる物は少なく、事前に計画した軌道を飛行すれば十分であり、誘導の自律化の必要性は低い。このような理由から従来の自律誘導の研究は、制御と比較して少ないと考えられる。しかし、今後パイロットが搭乗しない UAV が発展し、低空を飛ぶ機体が建物の間を飛び交う場合や、互いの衝突を回避するなど、事前の誘導計画だけでは不十分となる事象への対応能力を向上させるためには、自律誘導が必須となる。

本節では、航空宇宙機用の自律誘導に関する先行研究を取り上げて、本研究の方針と意義を明確にする。従来研究のほとんどは、現在地から目標地点までの最短距離（最短時間）となる軌道を求める問題を扱っており、そのなかでも主に二つのグループに分けられる。一方は、機体の運動モデルを考慮して追従性の高い軌道生成を目指した研究であり、他方は、モデルの運動をあまり考慮せず、経路探索アルゴリズムを用いて経路を高速に探索することを目指した研究である。本研究が目指している「複雑な障害物を避けて飛ぶことが可能となるリアルタイム自律誘導」を実現する上で、従来研究から得られる知見や想定される課題について考察し、研究方針を示す。

1.2.1 運動モデルを考慮した最適軌道生成の研究例

Richards らは、2002 年に混合整数線形計画法（MILP：Mixed Integer Linear Programming）を航空機の衝突回避軌道生成問題に適用した^[46]。この研究では、機体の運動を最適化の一種である線形計画法に適用させるため、質点系における 2 次元平面運動（高度方向の運動を考慮しない）モデルとして近似的に取り扱った。飛行速度を一定とした上で発生する旋回率に制約条件を設けており、目的地までの移動時間を評価関数として最小化を試みている。その結果、1 つの機体が障害物を避けて目的地まで到達する最短軌道（Fig. 1.2.1 (a)）や複数機が衝突する事象における回避軌道（Fig. 1.2.1 (b)）の生成にそれぞれ成功しており、更に複数の経由地点を通過する軌道の生成にも成功している。

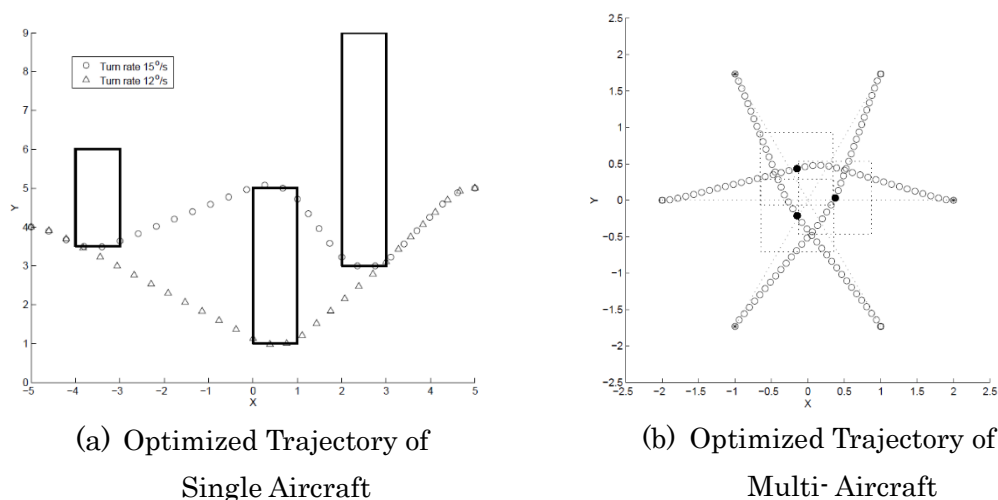


Fig. 1.2.1 Trajectory Optimization using MILP

一方で Miyamoto らは、より厳密な運動モデルを立てて最適軌道生成を行った。宇宙往還機の帰還滑空飛行において、3 次元空間を運動する質点系の連立常微分方程式を解き、目的地までの障害物を回避する最適軌道生成法を提案した^[47]。最適化手法には、生物が進化する過程をモデル化した遺伝的アルゴリズム（GA：Genetic Algorithm）を用いており、飛行時間、目標位置と終端位置の誤差、目標方位角と終端方位角の誤差などをひとつの評価関数として扱い単目的最適化を行っている。GA による最適化では、全ての個体について軌道と適応度を計算する必要があり膨大な計算コスト（計算時間）が掛かるため、この課題の克服がリアルタイム誘導を実現する上で困難な課題とされた。同研究では、軌道をフーリエ級数で表現することで最適化変数を削減し、さらに並列計算が可能な集積回路である FPGA（Field Programmable Gate Array）に実装すること

により計算速度の問題を解決し、1 秒程度での軌道生成に成功している (Fig. 1.2.2).

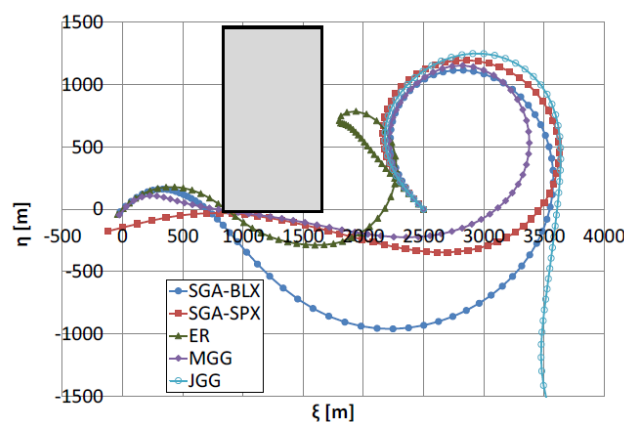


Fig. 1.2.2 Trajectory Optimization using GA

これらの先行研究は、いずれも運動モデルを考慮して最適化計算を行うため、機体の追従性が高い障害物回避軌道が得られると思われる。しかし、いずれの研究でも障害物は1箇所または数機の衝突を想定しているため、障害物の著しい増加に伴うリアルタイム性の評価が必要と考えられる。また、Richards らの手法では、2次元空間のみに対応しており、有翼ロケットのような滑空帰還飛行による高度変化を伴う場合には3次元空間への拡張が必須課題となる。さらに線形計画問題に適用させるため、非線形モデルを扱った厳密な障害物回避軌道の生成に適用するのは困難である。一方、Miyamoto らの研究では、非線形モデルによる3次元空間のリアルタイム軌道生成に成功している。同研究ではさらに、GAにより一つの解軌道に収束してしまうことを避けて様々な解を選択可能とするために、動的分散遺伝的アルゴリズム (DynDGA: Dynamically Distributed Genetic Algorithm) を提案している^[48]。しかし DynDGA には、各軌道の特徴量の算出や非類似度の計算などが必要とされ、計算量が大幅に増大するため、オンボードによるリアルタイム性の確保には、より高性能な FPGA を複数機並列で計算させるなどの工夫が必要であることが記されている。そのため、火星探査航空機のような小型化、軽量化を厳しく要求されるシステム構成に適用することは容易ではない。また、高性能な FPGA は、一般的なマイクロコンピュータと比較して非常に高価なため、開発費の制約が厳しい安価な小型 UAV 等への適用は困難と思われる。

そこで本研究では、障害物が著しく増加してもリアルタイムの軌道生成を可能とし、小型、軽量の機体にも実装可能とする低コストかつ高速なアルゴリズムの構築を目指す。

1.2.2 経路探索アルゴリズムを用いた最短経路生成の研究例

これまでに紹介した最適化アルゴリズムによる軌道生成法は、計算負荷が主な課題として考えられるが、複雑な障害物環境にも対応可能で、高速に飛行経路を探索する手法として経路探索法を応用した研究例がある。経路探索法は、これまで主にロボティクス分野の経路計画問題（またはパスプランニングと呼ばれる）に対するアプローチとして研究が進められてきた（経路探索法について、詳しくは第2章で述べる.）。その経路探索法を UAV に適用した研究例として、2005 年に Doeblner らによるリアルタイム障害物回避経路計画法が提案された^[49]。障害物となる地形が存在する 3 次元空間をグリッド空間で表現し、各グリッドの交点（ノードと呼ばれる）情報を用いて A*（A-Star）アルゴリズムで最短経路問題を解いた。A*アルゴリズムは、パスプランニングにおいて最短経路問題を収束計算を必要とせず高速に解くことができるアルゴリズムとして知られ、最短経路を保障する経路探索問題に適した最適化手法の一種である。この手法を用いた結果では、探索するグリッドの分解能によって得られる経路の精度が異なり（Fig. 1.2.3）、良解を得るためにグリッドの分解能を向上させると、反対に計算時間が増加することが示されている（論文では 5 秒から 38 秒まで増加している）。

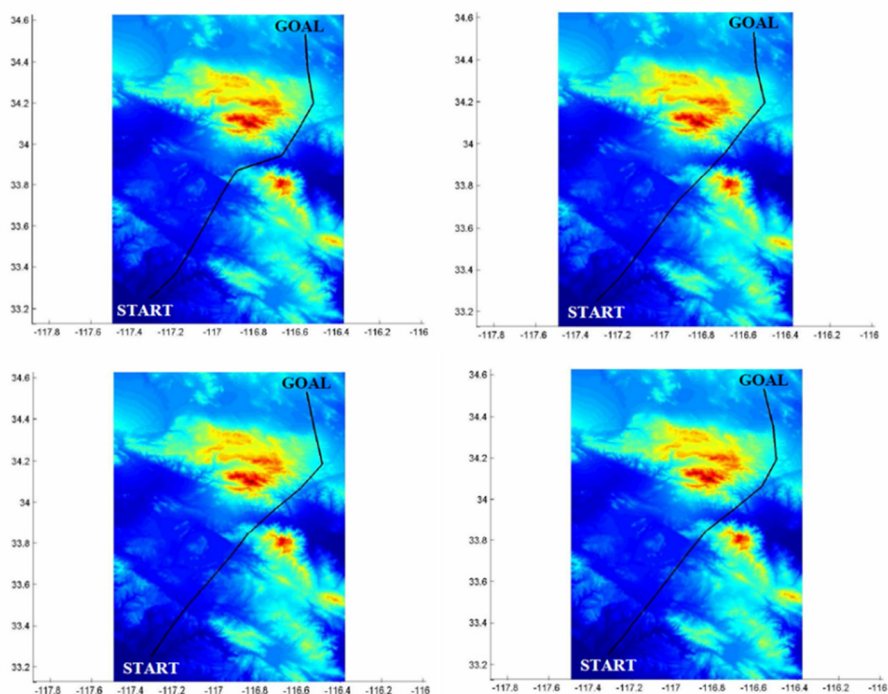


Fig. 1.2.3 Real-Time Path Planning Using A* Algorithm

2008 年には、Hrabar らによって障害物のリアルタイム検知を考慮した回転翼機の衝突回避経路計画法が研究された^[50]。ステレオカメラを用いて障害物の位置を検知し、構造化手法の一種である PRM (Probabilistic Roadmap Method) により障害物に衝突しないノードの構造体を作成し、D* (D-Star) と呼ばれる A* を改良した手法を用いて飛行経路の探索を行った。シミュレーションの結果では、障害物を避けて飛ぶ結果が多く示されたが、衝突する失敗例も複数見られた (Fig. 1.2.4)。これは、経路計画そのものは衝突を回避していても、機体のダイナミクスが影響して軌道を逸脱することが原因であると記されている。一方で、PRM と D* の組み合わせにより、計算コストを抑えた高速探索を可能にしている。これは A* や D* といったアルゴリズムは通常、Doebbler らの研究のように全探索領域をグリッドで分割した膨大な数のノードを用いて探索するが、PRM を用いることにより、少ないノード数での探索を可能にしているためであると考えられる。

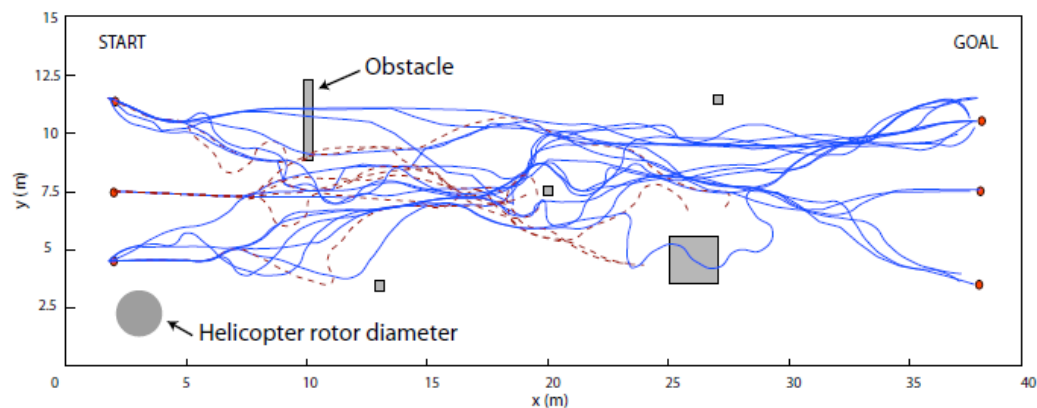


Fig. 1.2.4 Real-Time Path Planning using D* Algorithm

経路探索アルゴリズムと最適化アルゴリズムを組み合わせ、さらに複雑な環境を想定した課題に取り組んだ研究例がある。2012 年に Mitsutake らは、地形や雨域などの複雑な飛行禁止区域を回避しながら多数のウェイポイントを巡回する 3 次元空間の経路生成問題を研究した^[51]。従来の研究では、衝突回避問題と巡回順最適化問題を同時に考慮していたのに対し、満武らが提案する A*-EC 法では、それらを分離し、はじめに生成したグリッド空間について A* を用いて衝突回避問題のみを解決し、衝突回避ノードの集合体についてのみ、GA による組み合わせ最適化を行うことで計算の効率化、高速化を実現した。ソフトウェアシミュレーションの結果より A*-EC 法を用いた場合の

探索時間は、従来法に比べて 1/100 程度まで高速化されたと述べられている。また、UAV を用いたリアルタイムシミュレーションによる検討では、集中豪雨を回避する課題を想定した動的障害物に対しても飛行経路の更新を行うことで良好な軌道を獲得している (Fig. 1.2.5)。

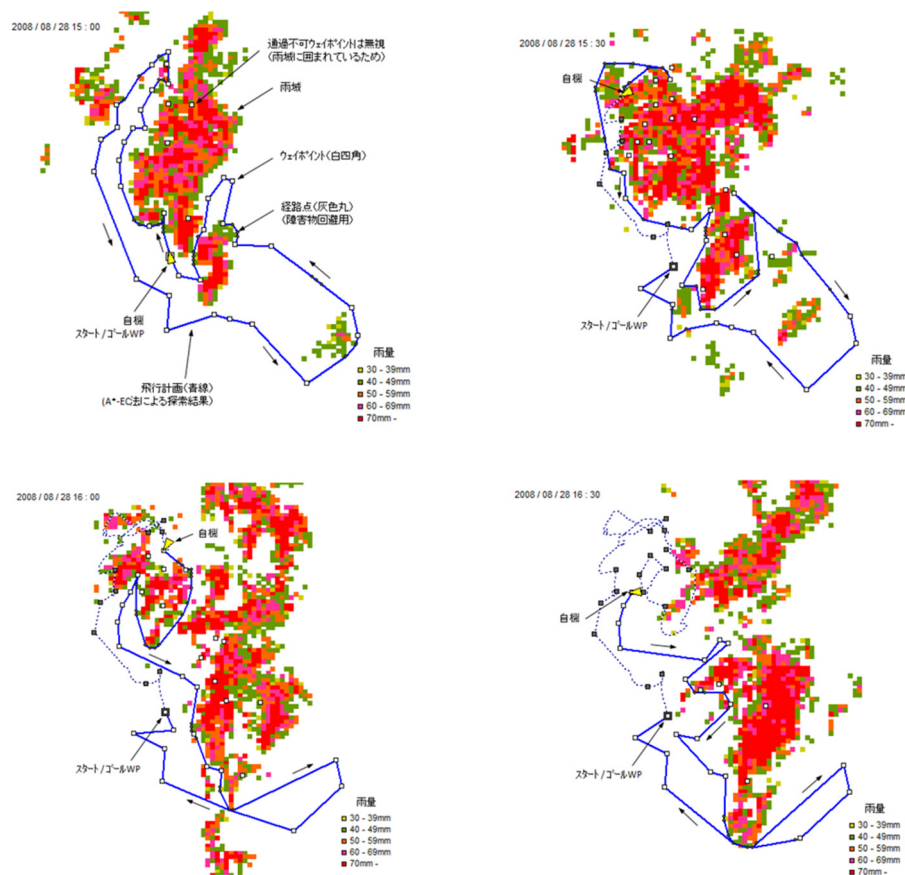


Fig. 1.2.5 Real-Time Path Planning using A*-EC

このように、経路探索法を用いた障害物回避問題は、様々な計算の高速化手法が研究されている。しかし、経路探索法を航空宇宙機の誘導に用いるためには大きな課題が考えられる。それは、経路探索法を用いた従来研究の多くはモデルの運動がほとんど考慮されておらず、障害物との衝突回避性を十分に満足できない、つまり障害物に衝突する可能性が大きいことを意味する。経路探索法では、ノードと呼ばれる点と点を結合することで経路を生成することが一般的であるため、基本的に探索される解経路は直線、ま

たは曲率が予め決められた曲線等で表現された幾何的な経路となる．この場合、実際のモデルの運動との乖離が問題となる場合が考えられる．ホバリングにより静止が可能な回転翼機などの機動力が極めて高い UAV に対して、幾何経路上を飛行する際にはモデルとの誤差は小さく抑えられるため問題とはなりにくい．一方で、火星探査航空機や有翼ロケットのような静止不可能かつ運動方向に制約のある固定翼機においては、幾何経路の通りに飛行することは難しく、その誤差が経路の逸脱や障害物との衝突の大きな要因となる．誤差を無視するためには、極めて広範囲の経路生成問題とすることも考えられるが、アルゴリズムがミッションの設定に制約を課すことは好ましくない．また、火星探査航空機は、100km 程度の探査ミッションを想定しており、有翼ロケットは滑空帰還飛行であることから、モデルの運動を無視できるほど広範囲の飛行を想定していない．このような場合には、従来手法では障害物回避の保障が十分ではなく、衝突のリスクが高まることが懸念される．また、小型 UAV が狭く複雑な環境を飛行する際にも、機体の運動特性を十分に考慮した誘導アルゴリズムが望ましい．

そこで本研究では、モデルの運動を考慮した軌道を生成することで、高い追従性と障害物回避性を実現するためのリアルタイム誘導アルゴリズムの構築を目指す．

1.2.3 先行研究のまとめと本研究の意義

本節の最後に、これまでに紹介してきた先行研究の特徴をまとめ、本研究の意義と新規性について明確にする．各先行研究および本研究の特徴を一覧にまとめたものを Table 1.2.1 に示す．

Table 1.2.1 Feature of Research

	Methods			Characteristics		
	Dynamics Model	Optimization	Path Planning	Optimality	Dynamical Correctness	Calculation Speed
Richard, A (2002)	Point Mass	MILP	—	○	△	×
Miyamoto, S (2014)	Point Mass	GA	—	○	○	×
Doebbler, J (2005)	—	(A*)	A*	○	×	△
Hrabar, S (2008)	—	(D*)	PRM & D*	○	×	△
Mitsutake, K (2012)	—	GA	A*-EC	○	×	×
Present Research	Point Mass	—	RRT	×	○	○

図に示す特徴の最適性（Optimality）から読み取れるように、ここに挙げた先行研究は、全て最適解が得られるアルゴリズムである．フライトコストが最も少なくなる最短軌道を生成することは、誘導問題において極めて重要な課題であることが分かり、その満足のためにいずれも最適解が得られるアルゴリズムとなっている．論文の記載内容から計算コスト（Calculation Cost）について各種法を一律に評価することは難しいが、MILP や GA などを用いた最適化手法は、一般的に収束計算に膨大な計算コストが掛かるため、低い評価（コストがかかる）と判断した．A*や D*といった手法も最適解を与えるアルゴリズムではあるが、収束計算を必要とせず高速に解を得られる．しかし、その解法には、グリッド空間を形成する必要があるなど、A*や D*を実行する前処理の計算負荷が無視できないため評価を中としている．機体の運動特性の考慮（Airframe Dynamics Correctness）については、質点系運動方程式の数値解法により軌道を算出した Miyamoto らの研究成果が最も詳細にモデルを考慮していると判断して高い評価

とし、Richard らの研究では、定常飛行を仮定したモデルであるため評価を中とした。一方で、Doebbler や Hrabar, Mitsutake らの研究では、ノード（すなわち次の経路）を選択する際に、経路の角度などに制約条件を設けることで運動の考慮を簡易的に行っているが、結果として導かれる経路はノード同士を結合した直線の不連続な経路となるため、低い評価（追従しにくい軌道）とした。

以上の考察を踏まえて本研究では、これまでに研究がほとんど進められていない領域である、運動特性 (Dynamical Correctness) を考慮し、かつ計算コスト (Calculation Cost) を抑えたアルゴリズムを目指す。機体の運動モデルを考慮することで、高い追従性と障害物回避性を保持し、最適性を追わない着想により計算コストを抑えた、高速なリアルタイム障害物回避軌道生成手法を構築する。本研究の実現は、有翼ロケットや火星探査航空機のみならず、小型の UAV など計算資源に限りがある機体に対しても、瞬時に障害物を避けて目的地まで飛ぶことを可能にすることが期待できる。

1.3 研究方針

本研究では、機体の運動モデルを考慮して、多数の障害物が存在する複雑な飛行環境下でも実時間内に衝突回避できる高速な軌道生成アルゴリズムの構築を目指す。本節では、研究手法の方針を記す。ここで、本稿において“経路”とは、機体の運動モデルを考慮していない幾何的な直線または曲線の連続を指し、“軌道”は、運動モデルが考慮された飛行に適した直線または曲線のことを示す。

1.3.1 障害物回避誘導に適したランダム探索の構築

経路探索法は、障害物を回避する経路を探索するために用いられる。経路探索法には様々なアルゴリズムが存在するが、その各種経路探索法の特徴などの詳細については第2章で述べる。それらの中でも特に探索環境の影響を受けにくい特徴を持ち、二点間の高速経路探索が可能なランダム探索アルゴリズムである RRT (Rapidly-Exploring Random Tree) を本研究の経路探索手法として適用する。RRT は、探索域内でランダムに木の枝を成長させて探索を進めるアルゴリズムである。このランダムな特性から、RRT が生成するスタートとゴールを結ぶ経路は無駄な屈折を多数含んでおり、必ずしも航空宇宙機が飛び易い経路ではない。そこで本研究では、RRT に無駄な屈折の省略化手法を加え、以降に説明する機体のモデルを考慮した軌道生成に適した工夫を施す。本論文では、この経路探索の過程を“ランダム探索”と呼び、各種経路探索法の特徴を比較検討し、さらにランダム探索アルゴリズムのシミュレーションを行うことで、障害物数の著しい増加に対しても問題なく衝突を回避する経路探索が行えることを検証する。

1.3.2 航空宇宙機に適した力学フィルタの構築

ランダム探索では、障害物を回避する幾何経路が得られる。しかし、幾何経路をそのまま目標経路として航空機が飛行すると、必ずしもその経路に沿って飛行することが保証されているわけではなく、その誤差から障害物に衝突する恐れがある。前節で述べた先行研究の中には、機体の機動力に比べて極めて広範囲な探索域を設定することで、幾何経路のままでも誤差が無視できるように扱う研究も存在する。しかし、都市部のような複雑な環境における小型 UAV の近距離飛行などを想定する場合には、そのような制約は不適切となる。そこで、得られた幾何経路を厳密に飛べる軌道へと修正する必要がある。本研究では、従来ロボットアームなどの動作計画法等で一般的な手法として知られる逆運動学理論の考え方を応用してこの問題を解決する。逆運動学計算では、先に目

標値が与えられ、その目標値に合うモデルの入力量を逆算することで、要求する変化量を加える事ができる。ただし、本研究で扱う質点系の運動方程式の場合、入力項と出力項が互いに複雑に連成しているため制御器を適切に用いることが困難である。そこで、常微分方程式に対してヤコビアンを導入し、微小運動においては線形とみなせる仮定を用いることで入力項と出力項を分離する。そして逆運動学の考えに基づいて解き目的の値に近づける制御を可能にする。こうして運動モデルに基づき描かれる軌跡を目標軌道とし、更に逐次計算の過程で得られる推力、迎角、バンク角の時系列データを誘導が制御に与えるコマンドとする。このように力学モデルに基づいて幾何経路を修正する過程を本研究では“力学フィルタ”と呼ぶ。

1.3.3 ランダム探索と力学フィルタを用いた軌道プランナ

ランダム探索と力学フィルタを包括した本研究の誘導アルゴリズム全体を“軌道プランナ”と呼ぶ。本論文では、はじめにランダム探索と力学フィルタを組み合わせた基礎的な軌道プランナを構成し、軌道生成シミュレーションによりアルゴリズムの評価を行う。基礎的な軌道プランナでは、ランダム探索で障害物回避経路を探索し、その後に力学フィルタによって飛行軌道へ変換する。このとき、力学フィルタ後の軌道が障害物に衝突してしまうなど様々な問題が生じることが判明した（詳細は第3章で述べる）。そこで本論文では、ランダム探索に力学フィルタを融合させた新たな手法（融合型軌道プランナ）を提案する。本手法では、ランダムに与えられる軌道角度の目標値に対して力学フィルタを行うことにより、飛行可能な軌道を描きながら障害物を回避する探索が可能となる。この着想により、厳密な障害物回避性能を保持することが期待できる。そこで本論文では、融合型軌道プランナの軌道生成シミュレーションを行い、その有効性を検証すると共に、6自由度飛行シミュレーションにより、本アルゴリズムの誘導性能を評価する。

1.3.4 飛行試験によるオンボードリアルタイム誘導アルゴリズムの評価

研究アルゴリズムのリアルタイム誘導の性能評価を目的として、小型 UAV を開発する。UAV のオンボードコンピュータへアルゴリズムを実装し、飛行中に各種センサから得られる情報を用いて融合型軌道プランナの誘導計算を行うことによりリアルタイム性能を評価する。

1.4 論文構成

本論文は以下のように構成される.

第1章では, 研究背景と研究目的について先行研究や文献を取り上げて説明する.

第2章では, 各種経路探索手法を取り上げて, その特徴をシミュレーションにより確認し, 軌道プランナに適したアルゴリズムを比較検討する.

第3章では, 経路探索で得られた結果に対して, 機体モデルを考慮した飛行可能な軌道を獲得する力学フィルタの手法について述べる. また, ランダム探索と力学フィルタを組み合わせた基礎的な軌道プランナを構築する. さらに, 同アルゴリズムによる軌道生成シミュレーションを行い, 評価および考察をする.

第4章では, 第3章で明らかとなった課題からアルゴリズムを改良し再構築する. また, 軌道生成シミュレーションによって提案手法の有効性を検証する.

第5章では, 第4章で構築した研究手法を用いて, リアルタイムの6自由度誘導飛行シミュレーションを行う. シミュレーションは火星探査航空機, 有翼ロケット実験機の両モデルで行う. また, それらの結果を示して考察する.

第6章では, 提案手法によるUAVを用いたリアルタイム誘導計算実験の結果について述べる.

第7章では, 本論文の結論を述べる.

第 2 章 航空宇宙機の誘導に適した経路探索手法の構築

2.1 はじめに

障害物を避けたり，現在地から目的地に到達するための経路を得る手法を経路探索法（またはパスプランニング）と呼び，主に地上の自走ロボットやロボットアームの動作計画などを対象として研究が行われている．本研究で取り扱う衝突回避問題を解くためには，一般的に以下の手順が必要となる．

- I. 探索領域を設定し，探索域内に障害物と対象物を配置する．
- II. 探索領域を構造化する．
- III. 構造化された探索空間に対して経路探索を行う．
- IV. 必要に応じて経路を修正する．

一般的に経路探索法が上記手順のどこまで含むのかという線引は明確に定義されていないが，本章では，これらの課題に対する手法全てを包括して経路探索法と定義する．また，各種経路探索アルゴリズムの特徴を比較検討し，本研究の飛行誘導アルゴリズムに適した手法に関して詳細について説明する．さらに，実際に同手法を用いた多数の障害物環境下における経路探索シミュレーションの結果を示す．

2.2 経路探索アルゴリズムの比較検討

経路探索アルゴリズムは多数存在するが，本節では一般的によく使われる下記 3 種類のアルゴリズムについて特徴を比較検討し，その中から本研究に適する手法を選定する．

- I. A* (A-Star)
- II. PRM (Probabilistic Roadmap Method)
- III. RRT (Rapidly-Exploring Random Trees)

2.2.1 A* (A-Star)

A*アルゴリズムは、それまで最短経路問題を解くグラフ探索手法として主流であった Dijkstra's Algorithm^[52] (ダイクストラ法) を効率化した手法として 1968 年に Hart, P.E.らによって提唱された^[53]. A*は、スタート地点からゴール地点までの最小コスト探索解を保障するアルゴリズムである. 同手法では、ヒューリスティック関数 h^* を用いたコスト計算により、コストの低い地点から優先的に探索を進めることで最短経路を導くことができる. ヒューリスティックとは、ある程度の妥当性を持った見積もりを意味する. A*に用いられる評価関数を Eq. (2.1)に示す. $f(n), g(n), h(n)$ は、それぞれ全コスト、スタートから地点 n までのコスト、地点 n からゴールまでのコストを表す. ここで、地点 n からゴールまでのコストは、未来の不明確な情報が必要となるので、推定コスト $h^*(n)$ を導入する. $h^*(n)$ が Eq. (2.2)の制約を満たすとき、推定全コスト $f^*(n)$ の最も小さい経路から優先的に探索することで、最短経路を保障することができる. Eq. (2.2)の制約を満たす $h^*(n)$ の代表例は、任意の地点 n からゴールまでの直線距離である. 距離をコストとする場合、必ず直線距離が最短となり Eq. (2.2)の制約を満たす. このとき最短経路問題を解くことができる.

$$f^*(n) = g(n) + h^*(n) \quad (2.1)$$

$$0 \leq h^*(n) \leq h(n) \quad (2.2)$$

本手法では、探索を行うために前節で述べた探索領域の構造化が必要となる. A*では、構造化手法として一般にセル分割法^[54]が用いられる. これは、探索領域を互いに重ならない細かいセル領域に分割し、隣接するセル間の接続の可否により探索経路として使えるか否かが判断される. その結果得られたセル構造集合において A*により経路探索が実行される.

Fig. 2.2.1 にアルゴリズムの概念図を示す. はじめに探索領域内をセルに分割する (図左). そして、全コストが低くなるセルから順に探索を行っていく (図右). いま、地点 n を探索したとき、この経路は障害物に衝突するので、次にコストが小さいと思われる地点 n 横の四角について探索が行われる.

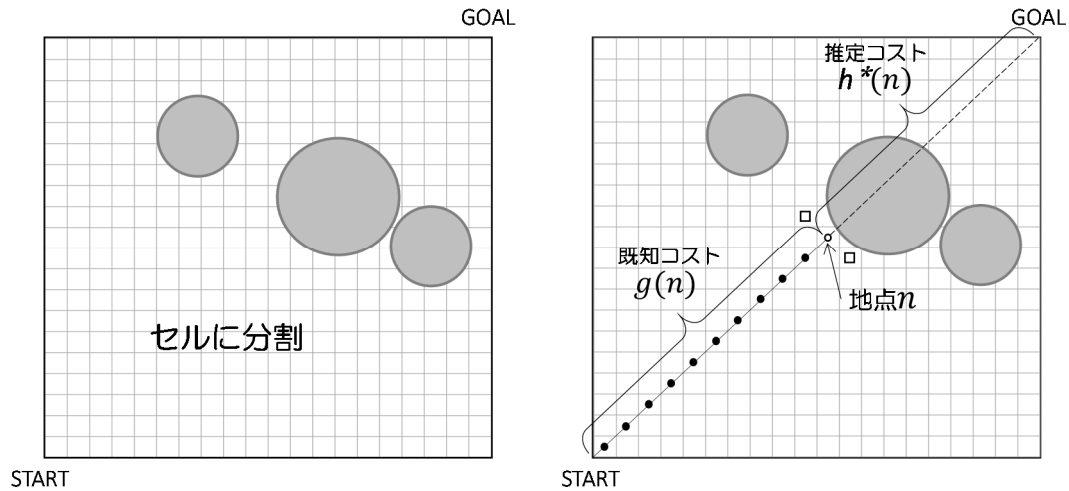
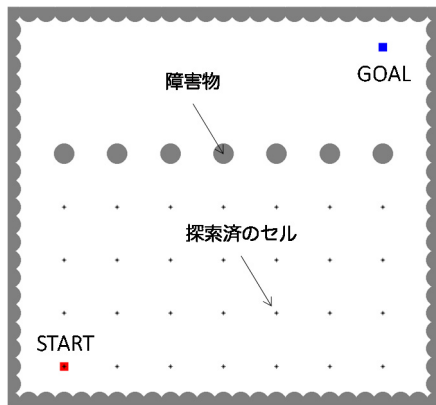
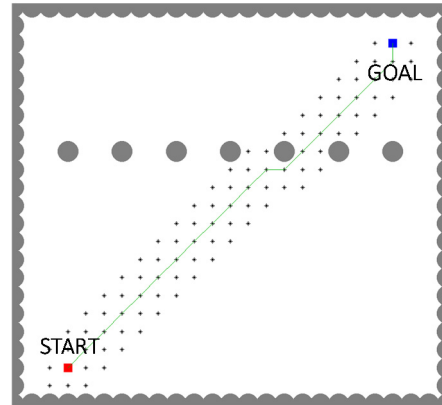


Fig. 2.2.1 A* Algorithm

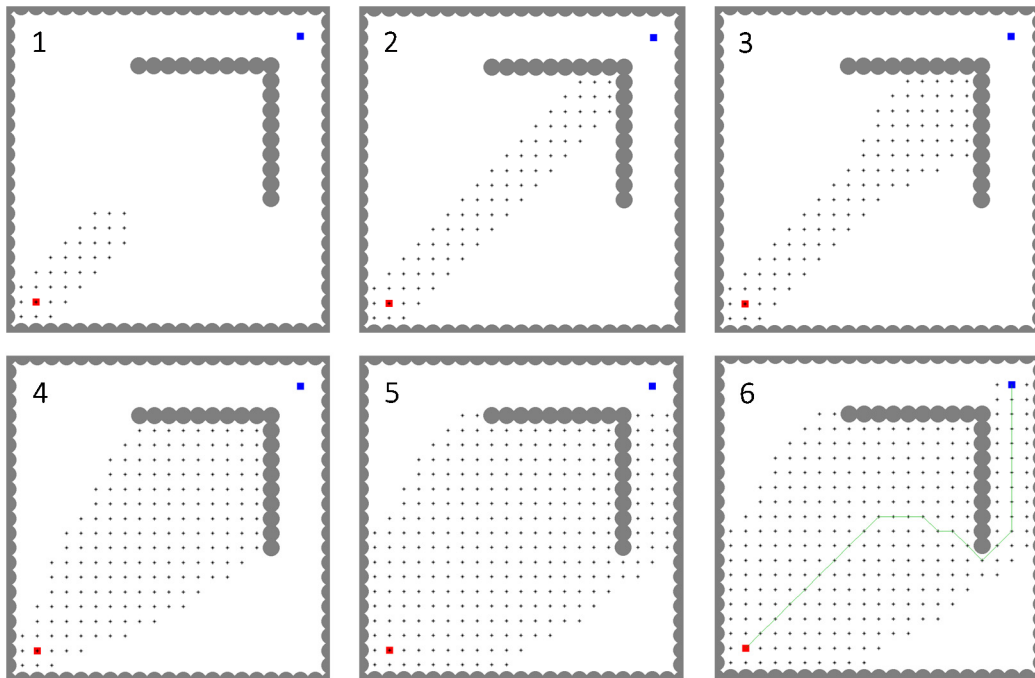
A*は最短経路を保障するので、エネルギーに限りのある航空宇宙機などでは有効な手法と思われる。しかし、一方で探索結果が必ず得られるとは限らないことや、高速性などに課題があるとされる。前者の課題は、構造化の際のセル分解能に依存する。セルの分解能が探索環境に対して低い場合、Fig. 2.2.2 (a)に示すシミュレーション結果のように探索が止まってしまうケースがある。図では、赤色四角をスタート地点、青色四角をゴール地点、灰色円を障害物として表現し、各ドットは探索が行われたセル領域を表す。同結果では、次に探索されるべきセルが、障害物と全て重なっているため探索が停止している。同じ障害物環境でも探索領域に対して十分にセル分解能が高い場合、Fig. 2.2.2 (b)のように解経路を得ることができる。しかし、分解能を高くすると計算量が増大するなどの問題もあるため、セル分解能の決め方に困難な課題がある。また、後者の高速性の課題は、セル分解能の問題だけでなく探索環境（障害物の配置）にも大きく依存する。A*は、最短コスト（ここでは最短経路）から優先的に探索を進めるため、Fig. 2.2.2 (c)に示すような大きく迂回する必要がある障害物環境などに対して、解を得るのに時間がかかる。



(a) Low Resolution Architecture



(b) High Resolution Architecture



(c) Weak Case

Fig. 2.2.2 Simulation Results of A* Algorithm

2.2.2 PRM (Probabilistic Roadmap Method)

PRM^[55]は、探索領域を構造化する手法として古くからロボティクス分野で広く用いられる。PRMでは、探索領域内に任意の数のランダム点を配置して (Fig. 2.2.3 左) それら全ての接続判定を行うことで構造化する (Fig. 2.2.3 右)。そこで得られた構造体は、その後ダイクストラ法や A*などを用いてグラフ探索され、解経路が導かれる。本手法は、先述のセル分割法と異なり構造体をランダムに構成するので、無駄な構造化を省略できる可能性がある。しかし、本手法も A*同様にランダム点の最適数を検討しなければならない課題がある。ランダム点が少なすぎるとスタートとゴールを接続できないケースが発生し (Fig. 2.2.4 (a)), ランダム点が多すぎると障害物と各線分の接触判定などに膨大な時間がかかり高速性に問題が発生するなどの課題がある。

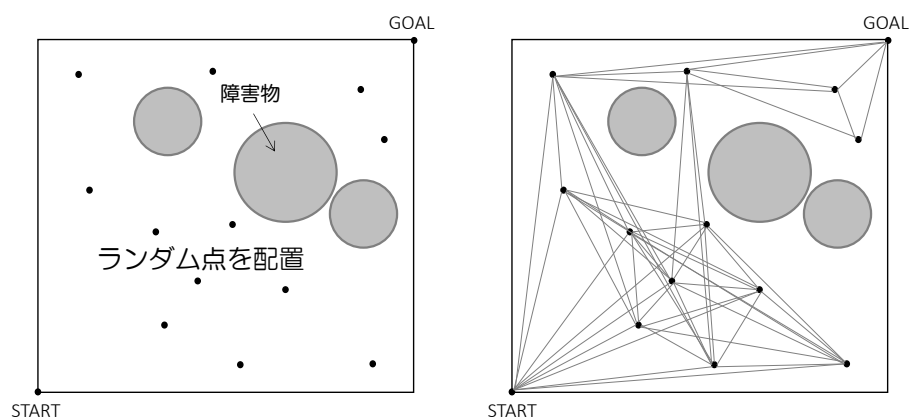


Fig. 2.2.3 PRM

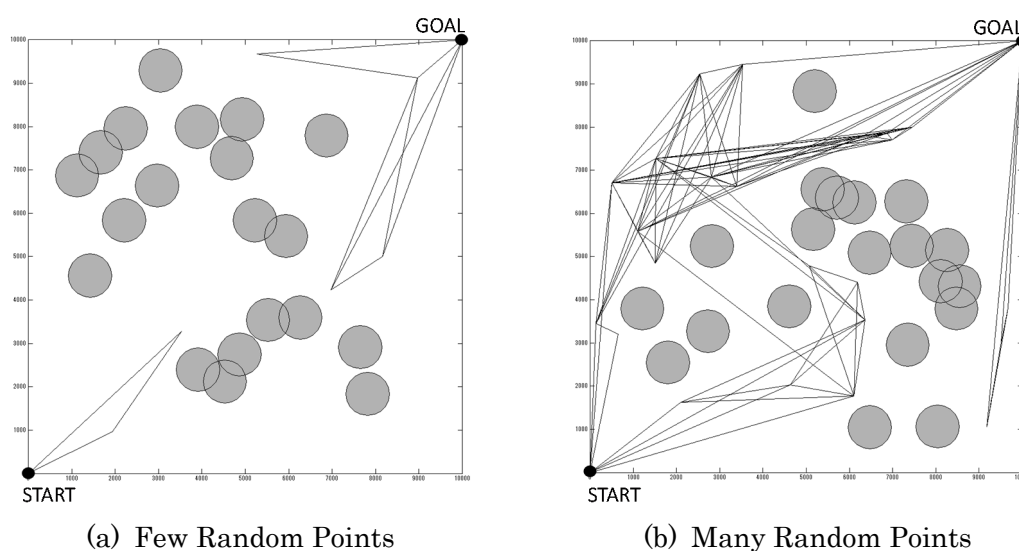


Fig. 2.2.4 Simulation Results of PRM

2.2.3 RRT (Rapidly-Exploring Random Trees)

RRT^[56]は 1998 年に LaValle によって発案されたランダム探索手法である。本手法は、探索領域の構造化と解経路の探索を同時に行う点が特徴的である。1 回の試行でランダム点を領域内に 1 点だけ定め、その点に向けてツリーの枝を伸ばす。そして、ゴール地点と結合できるまでツリーの成長を繰り返し解経路を得る。本手法は、ゴールと結合できた時点で探索を終了するため無駄な構造化（ランダム点の配置）が無く、コスト計算を必要とせず計算負荷が低い。また、ランダムな方向にツリーが成長するため領域内を均一に探索し、A*のように障害物配置に依存した得意、不得意が発生しにくい。

Fig. 2.2.5 には、2.2.1 項で行った A*と同様な障害物配置でシミュレーションした結果を示す。RRT の二点間結合アルゴリズムでは、ゴールに向かって接続を試みる手順がある（詳細は 2.3 節で述べる）ため、はじめにゴール方向にツリーが成長しているが、その後はゴール付近を優先することなく、領域内を均等に探索していることが分かる。そのため、迂回路などの発見に有利である。さらに RRT の特徴として、ゴールと結合するまで探索を継続することで、セル分割法や PRM のようにゴールまで到達しないという事象を回避できる。ただし、最短経路などの最適解を獲得することはできず、乱数に依存するため同じ環境でも毎回結果が異なり、探索時間も変動するなどの問題がある。

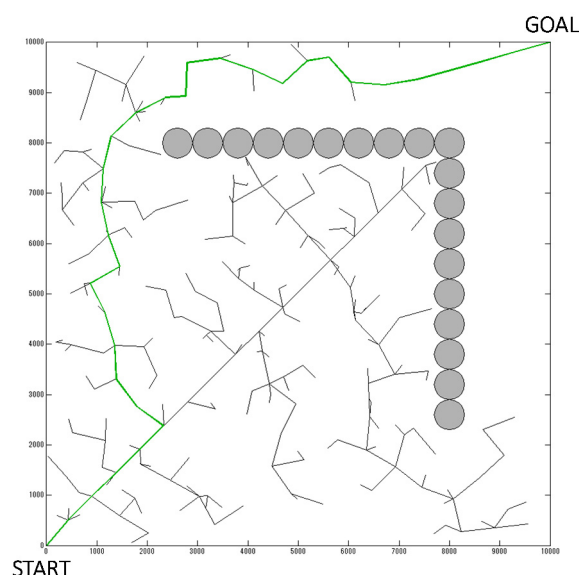


Fig. 2.2.5 Simulation Result of RRT

2.2.4 軌道プランナに適用する経路探索法の選定

これまでに、各種経路探索アルゴリズムの概念と手法を説明し、シミュレーションによりその特徴を確認した。これらの結果から軌道プランナに適用する経路探索法を検討する。

本研究の軌道プランナでは、コンセプトとして高速な軌道生成を挙げている。また、確実に目的地まで誘導する必要があるため、経路探索に失敗するケースの少ないアルゴリズムが望ましい。よって、無駄な構造化やコスト計算を行わずに高速に演算が可能で、障害物環境に依らず探索領域を均一に探索でき、かつ確実に経路を得ることができる **RRT** が本研究の経路探索手法に最も適していると考えた。**RRT** は先述の通り、最短経路を得ることはできないが、本研究では、膨大な障害物環境下でも有用なアルゴリズムを目指しており、最短でなくとも高速に解軌道を得ることに主眼を置くため、最適性の欠落は問題とはならない。よって **RRT** を本研究の経路探索法として適用する。

2.3 RRT の軌道プランナへの適用

本節では、**RRT** の手法について詳細を述べる．また、航空宇宙機の誘導として用いるための工夫も加える．**RRT** を経路探索法として用いるうえで、後述の衝突回避、二点間結合、経路簡略化アルゴリズムは一般的なアイデアとして存在するが、その手法に関して明確に記載されたものはほとんどない．そこで、本論文では、2.3.3 項～2.3.5 項でそれらのアルゴリズムを独自に定義する．

2.3.1 疑似乱数モデル

RRT の探索アルゴリズムには乱数モデルが必要となる．しかし、コンピュータで分布が一樣かつ周期性の無い完全な乱数を発生させることは難しく、実際には計算により一意に決まる一定周期を持つ擬似乱数列を用いて対応する．本論文のシミュレーションでは、疑似乱数モデルとして 1996 年に松本眞らにより考案された、長周期（1 周期が $2^{19937} - 1$ 回）、高次元均等分布（623 次元超立方体中に均等分布）を持つメルセンヌツイスタ^[57]（Mersenne Twister）を用いる．

ここで、実際に 3 次元空間においてメルセンヌツイスタが均等分布することをシミュレーションで確認する．Fig. 2.3.1 では、1000 点のランダム点（青色プロット）を空間内に配置した．結果より、ランダム点が空間内に均等に分布することが明らかとなった．

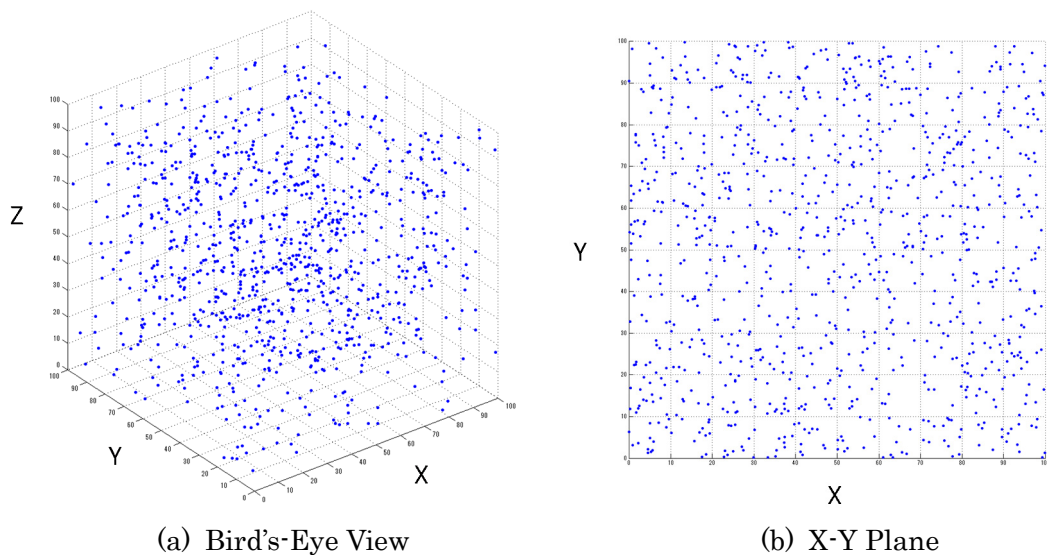


Fig. 2.3.1 Simulation Result of Mersenne Twister
(to be continued)

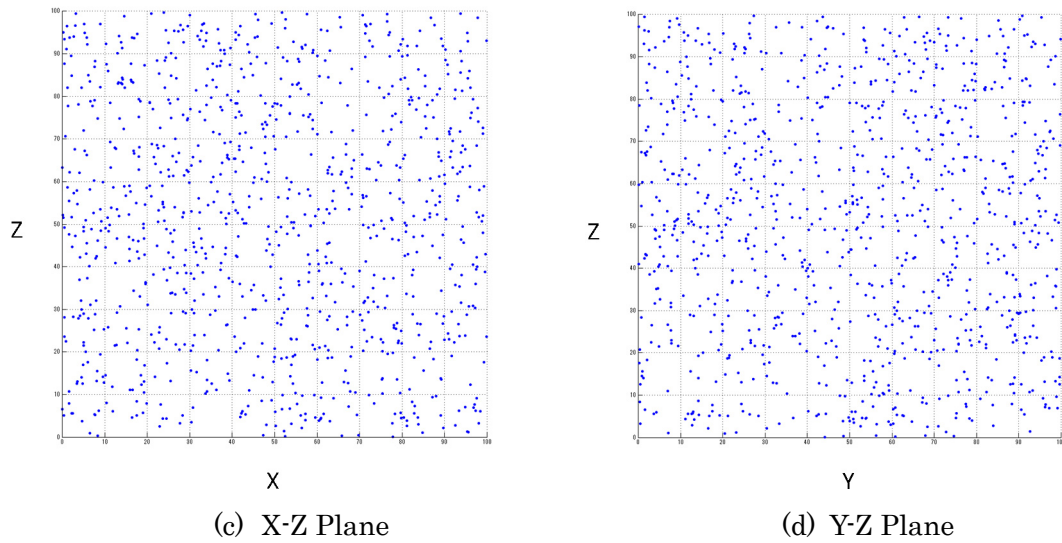


Fig. 2.3.1 Simulation Result of Mersenne Twister
(to be concluded)

2.3.2 基本アルゴリズム

ここでは、RRT の最も基本的なアルゴリズムを説明する。Fig. 2.3.2 にアルゴリズムのフローチャートを示し、探索過程の模式図を Fig. 2.3.3 に示す。RRT では、まずツリーの成長する基準点を事前に設定する (Fig. 2.3.3 (a))。一般にツリーの枝をエッジ (Edge) と呼び、枝と枝を結ぶ節 (屈折部分) をノード (Node) と呼ぶ。ここで、基準点は初期ノード (Initial Node) となる。次にランダム点 (Randomized Point) を 1 点のみ領域内に打つ (Fig. 2.3.3 (b))。そして、ツリー的全ノードのうちランダム点に最も近い点 (Nearest Node) を探索する (Fig. 2.3.3 (c))。一度目の最近傍点は必ず Initial Node となる。次に、最近傍点からランダム点に向けて予め設定した最大エッジ長さ ϵ だけエッジを成長させる (Fig. 2.3.3 (d))。そして成長後の座標が新たなノード (New Node) となる。このとき、最近傍点とランダム点の距離が ϵ 以下のときは、最近傍点とランダム点が直接結ばれる。最大エッジ長さ ϵ の制約は、エッジどうしの交差を防ぎ、木のよう成長させるために必要となる。これらの過程を反復してツリーは枝分かれを繰り返して成長していく (Fig. 2.3.3 (e))。以上が RRT の基本的な探索アルゴリズムである。

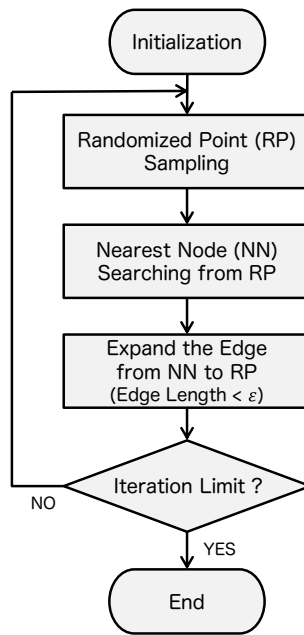


Fig. 2.3.2 Flow Chart of Basic RRT

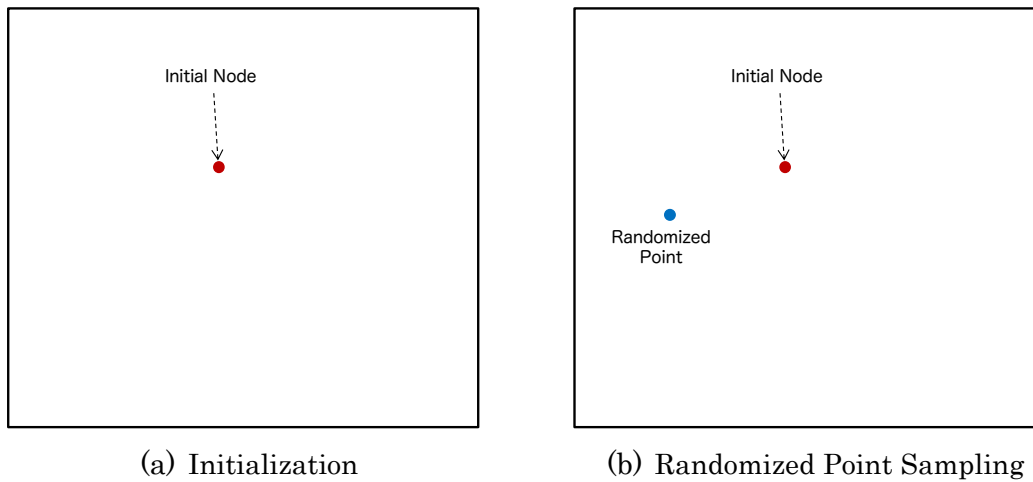
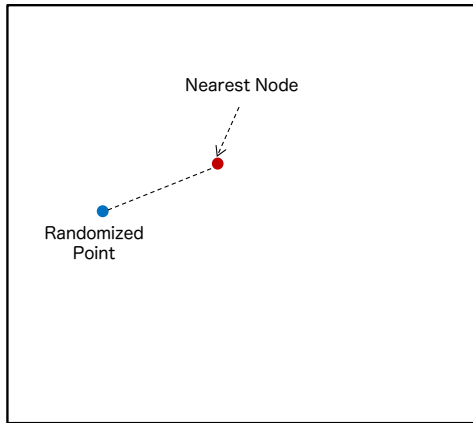
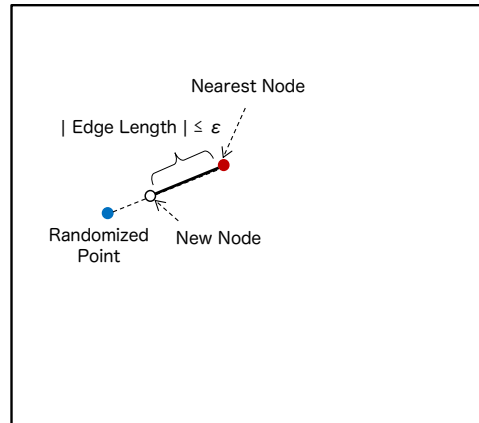


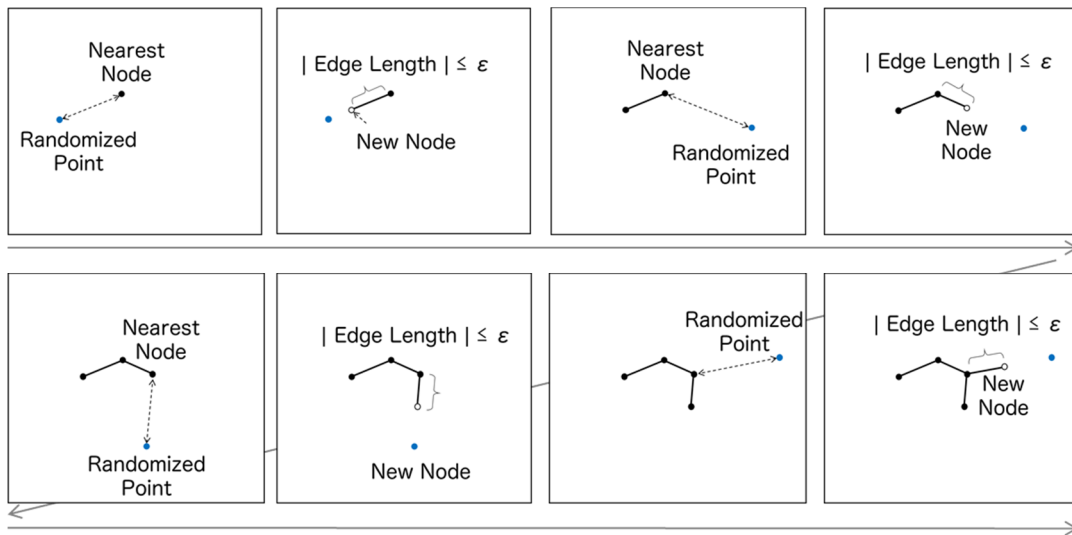
Fig. 2.3.3 Basic RRT Algorithm
(to be continued)



(c) Nearest Node Searching



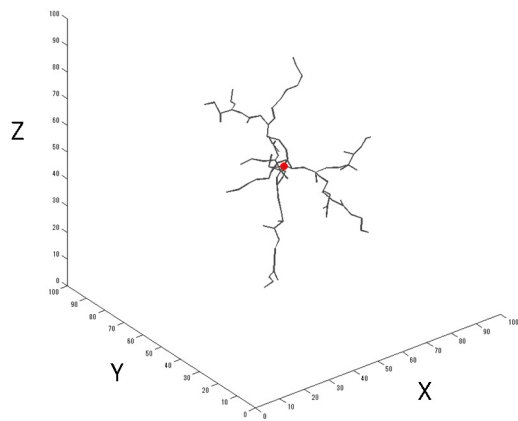
(d) Expand the Edge



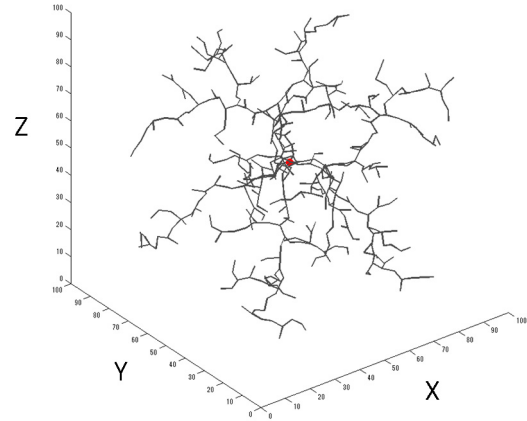
(e) Iteration

Fig. 2.3.3 Basic RRT Algorithm
(to be concluded)

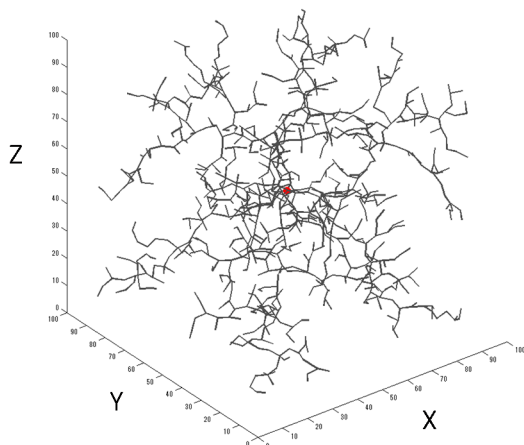
上記の基本アルゴリズムを用いて 3 次元空間でツリーを成長させたシミュレーション結果を Fig. 2.3.4 に示す. 中心の赤色丸プロットは初期ノードを表し, 探索回数 100, 500, 1000, 2000 回におけるツリーの成長過程を示している. シミュレーション結果より, 探索領域内でツリーが均一に成長 (探索) している様子が分かる.



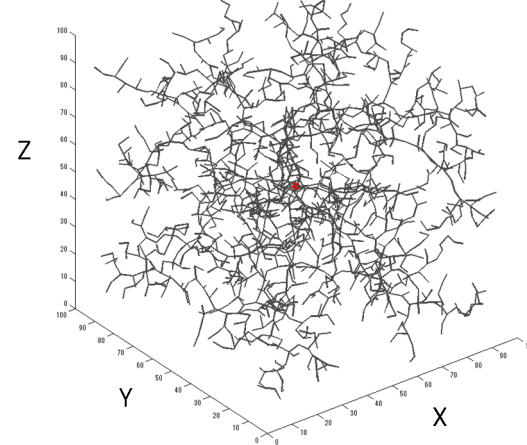
(a) Iteration 100



(b) Iteration 500



(c) Iteration 1000



(d) Iteration 2000

Fig. 2.3.4 Simulation of Basic RRT

2.3.3 衝突回避アルゴリズム

RRT の基本アルゴリズムにエッジと障害物の衝突判定機能を付与する．衝突判定はエッジを成長させる際に行われる (Fig. 2.3.5)．衝突判定の模式図を Fig. 2.3.6 に示す．本手法では，障害物を円形として模擬し（図内灰色丸），障害物の中心点からエッジに下ろした垂線の長さ D を障害物の半径 r と比較することで衝突を判定する．ただし，エッジと障害物に一定の距離を保つため，衝突回避マージン τ を設ける．垂線の長さ D が障害物の半径 r と衝突回避マージン τ の和以上であれば衝突しないと判断する．なお，衝突する際には，エッジを成長させることなく次の新たなランダム点が 1 点打たれる．

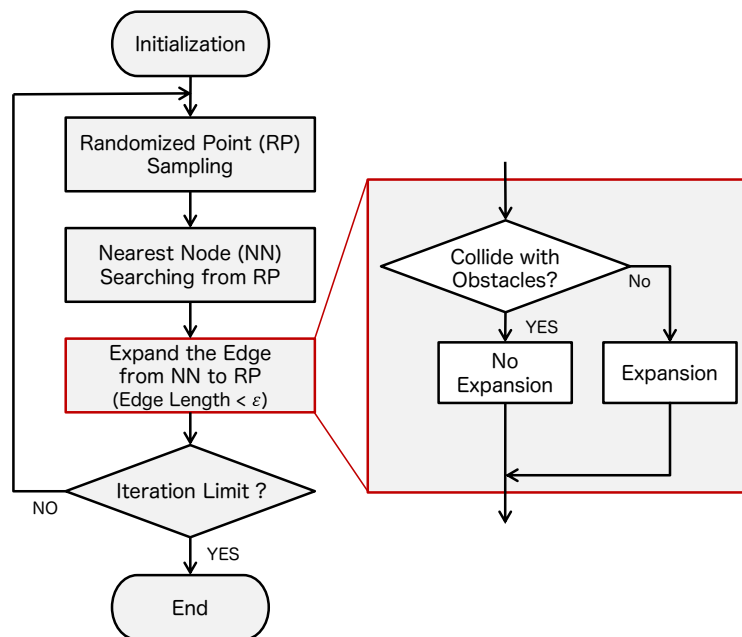


Fig. 2.3.5 Flow Chart of RRT with Collision Detection Algorithm

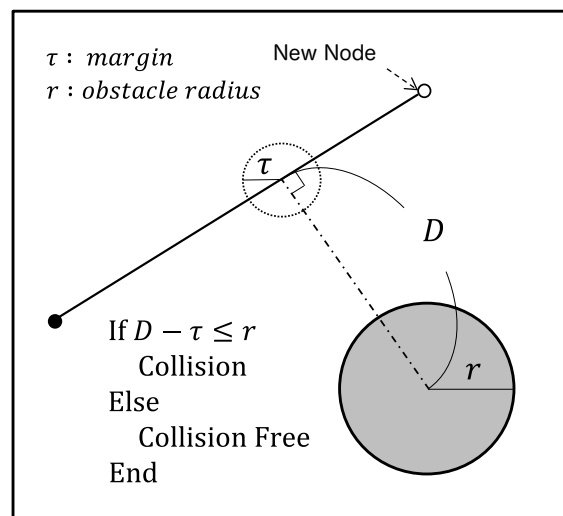


Fig. 2.3.6 Collision Detection Algorithm

ここで、先ほどの RRT のシミュレーションに障害物回避アルゴリズムを付加したシミュレーションを行う．シミュレーション結果を Fig. 2.3.7 に示す．結果より、障害物を回避して探索が行われることを確認した．

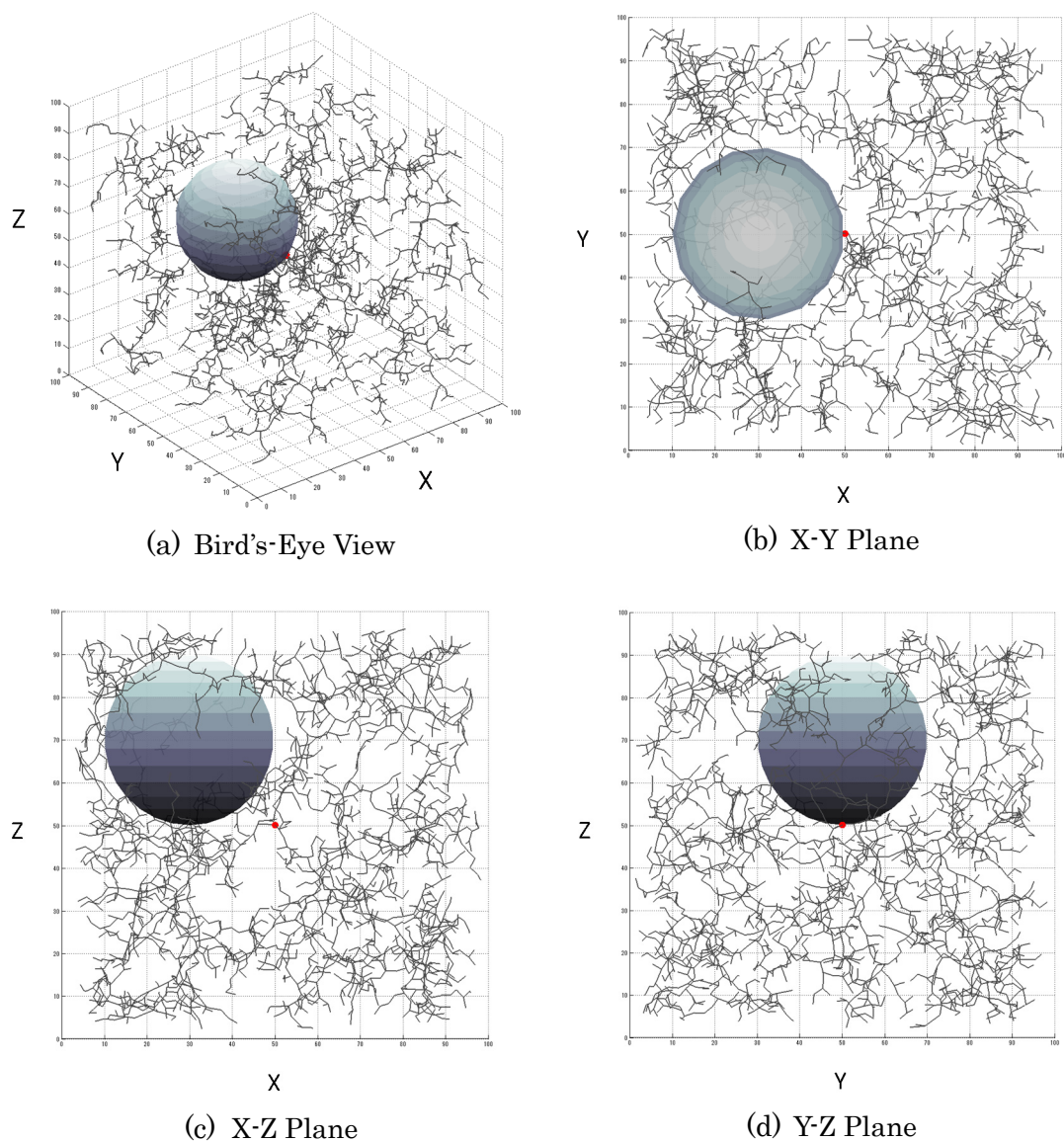


Fig. 2.3.7 Simulation Result of RRT with Collision Detection Algorithm

2.3.4 二点間結合アルゴリズム

ここまでに扱ったアルゴリズムは、始点からツリーを成長させるだけであったが軌道プランナにおいては、現在値から目的地までの二点間経路を生成する必要がある。そこで、二点間結合を可能にするアルゴリズムを付加する。フローチャートと概念図をそれぞれ Fig. 2.3.8 と Fig. 2.3.9 に示す。同手法では、エッジを成長させた後に、全ノードの中からゴールに対する最近傍点 (Nearest Node) を選び出す。そして、最近傍点からゴールに向かって最大エッジ長さ ε のエッジを障害物と衝突するまで複数回成長させる。

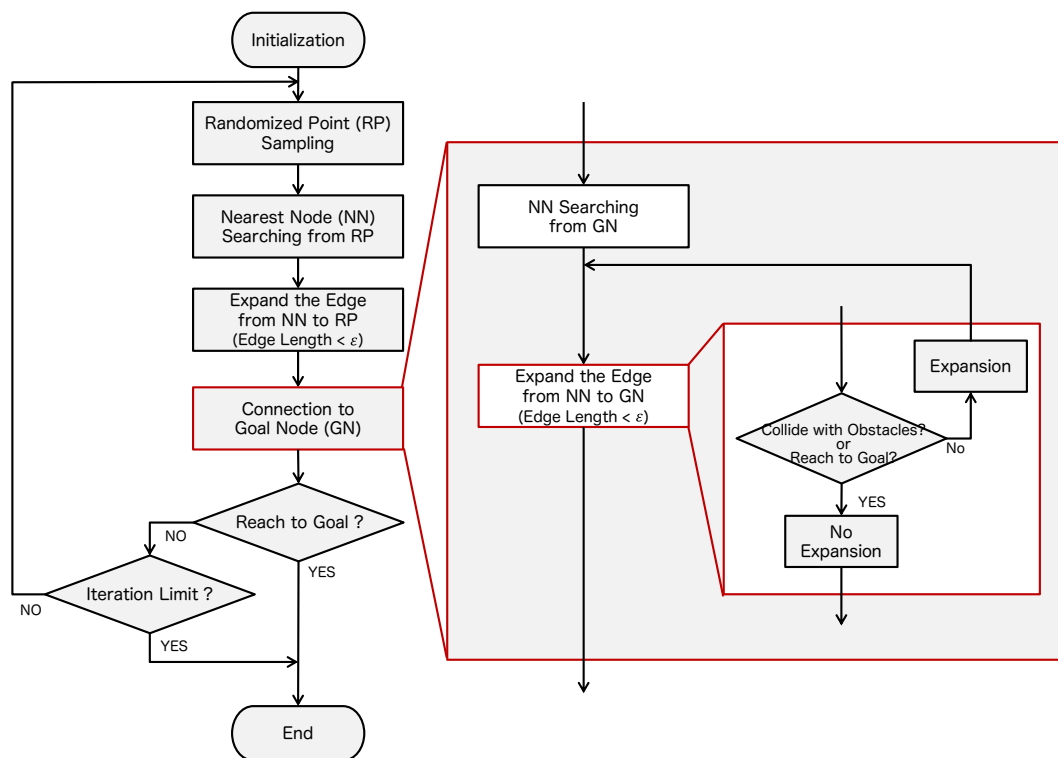


Fig. 2.3.8 Flow Chart of RRT with Connection Algorithm

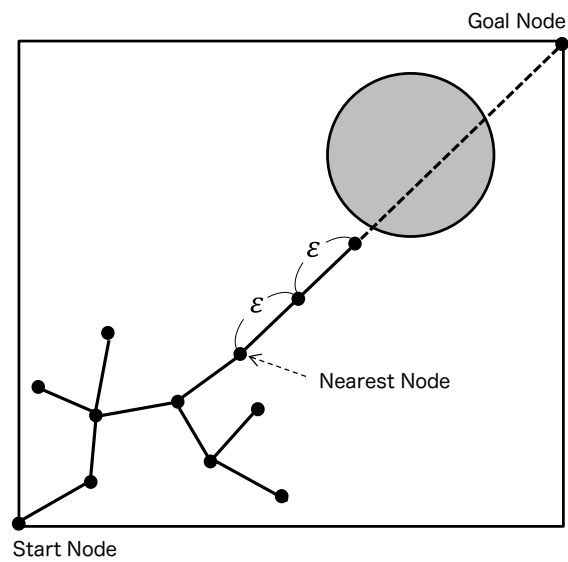


Fig. 2.3.9 Connection Algorithm

二点間結合アルゴリズムを付加した探索シミュレーションを行う． Fig. 2.3.10 にシミュレーション結果を示す．赤色実線で示す経路がスタートとゴール間で結合された経路を表す．結果より，障害物を回避しながら二点間の結合が可能となったことが確認できる．

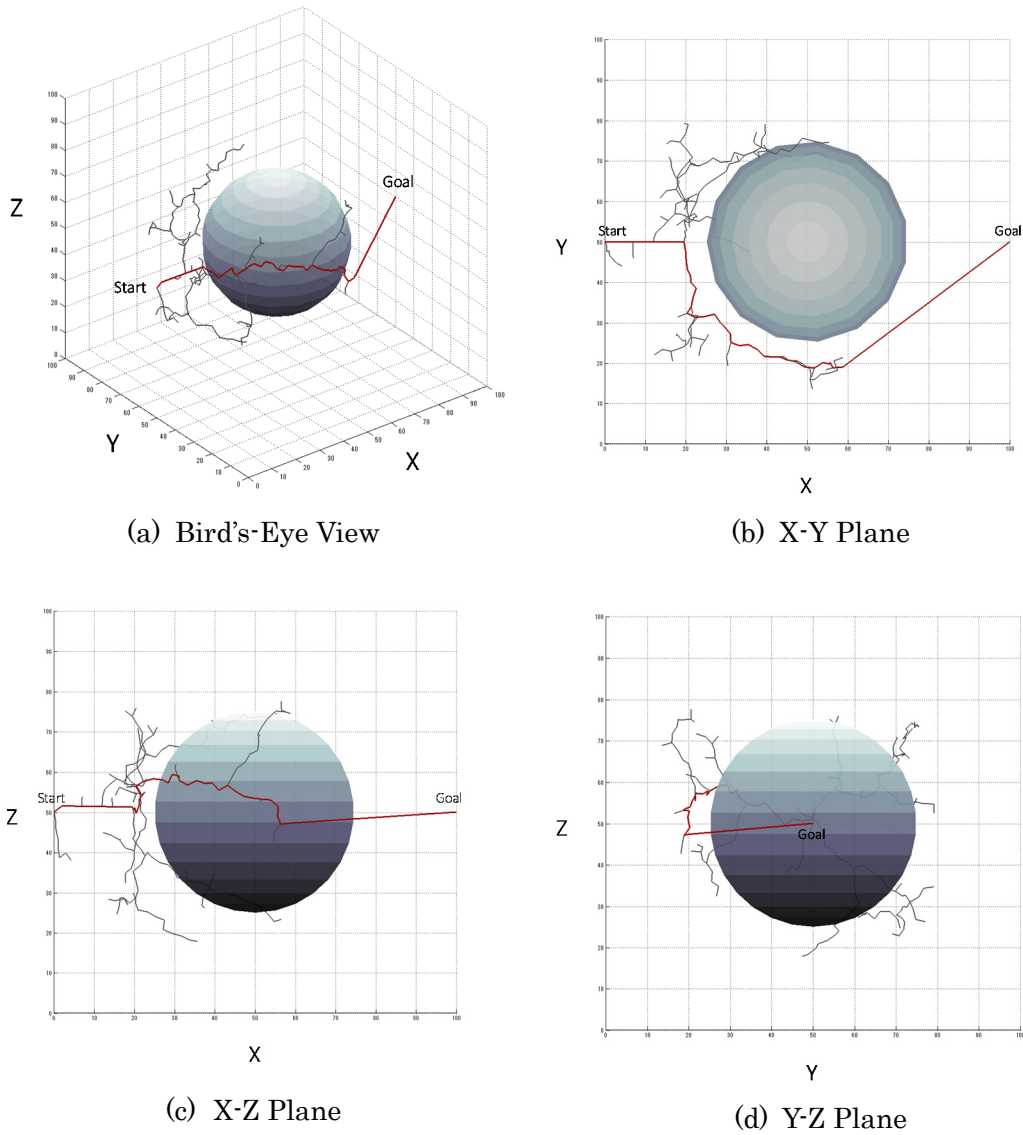


Fig. 2.3.10 Simulation Result of RRT with Connection Algorithm

2.3.5 経路簡略化アルゴリズム

2.3.4 項のシミュレーション結果から分かるように、RRT で得られる経路は、そのランダム特性により多くの屈折を持つ。航空宇宙機の誘導を考える上で、できるだけ無駄な屈折を減らし、平滑な経路にすることが望ましい。そこで本研究のランダム探索では、得られた RRT の解経路に対して無駄な屈折の省略を行う。同アルゴリズムのフローチャートと概念図を Fig. 2.3.11 と Fig. 2.3.12 に示す。スタートとゴールを結ぶ経路上のノードはそれぞれ ID を持っている（図内の数字）。このとき、ゴール地点の ID9 から順に遡ってスタート地点の ID1 と成す線分（赤色破線）の衝突判定を行う。この衝突判定は、2.3.3 項で述べた衝突判定アルゴリズムに基づいて行われる。図では、はじめに線分 ID1-9 の衝突判定を行うが、障害物に衝突している。次に線分 ID1-8 で同様に衝突判定を行う。このように順次 ID を遡っていくと線分 ID1-5 のとき、はじめて衝突することなく接続できる。この衝突の無い ID1-5 を新たなエッジ（緑色実線）とする。次に再び ID5-9 から順に遡って衝突判定を行う。これをゴールに到達するまで繰り返し、経路を省略していく。最終的には、衝突が無く無駄な屈折が省かれた経路（ID 1-5-6-8-9）が軌道プランナに用いる解経路として得られる。

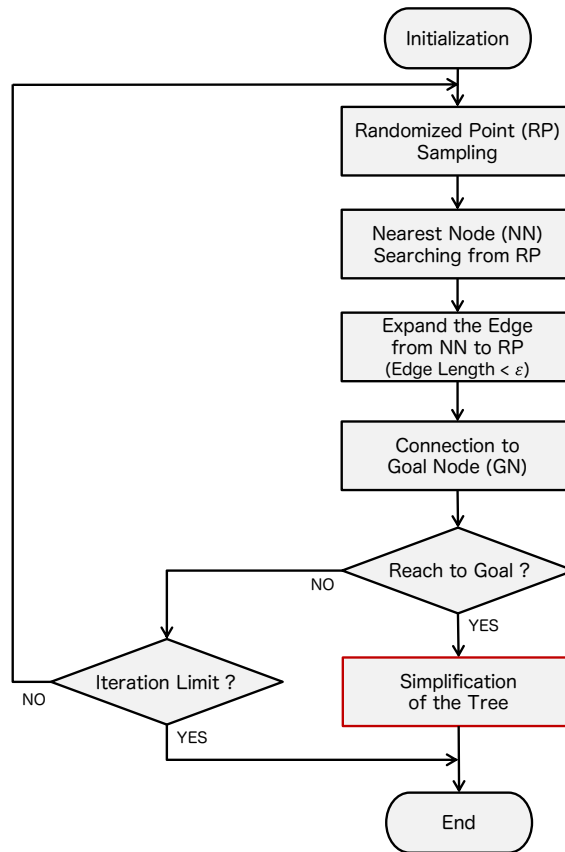


Fig. 2.3.11 Flow Chart of RRT with Shortcut Algorithm

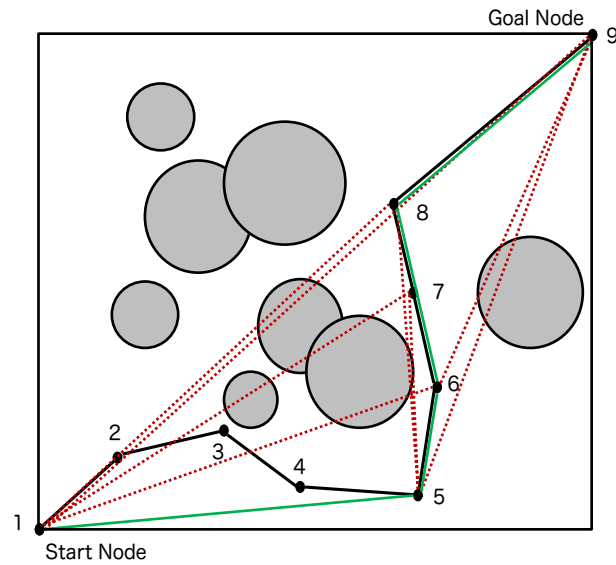


Fig. 2.3.12 Shortcut Algorithm

本アルゴリズムをシミュレーションにより検証する．シミュレーション結果を Fig. 2.3.13 に示す．図の緑色実線で表されるように，無駄な屈折が省かれた経路が生成できていることが確認できる．

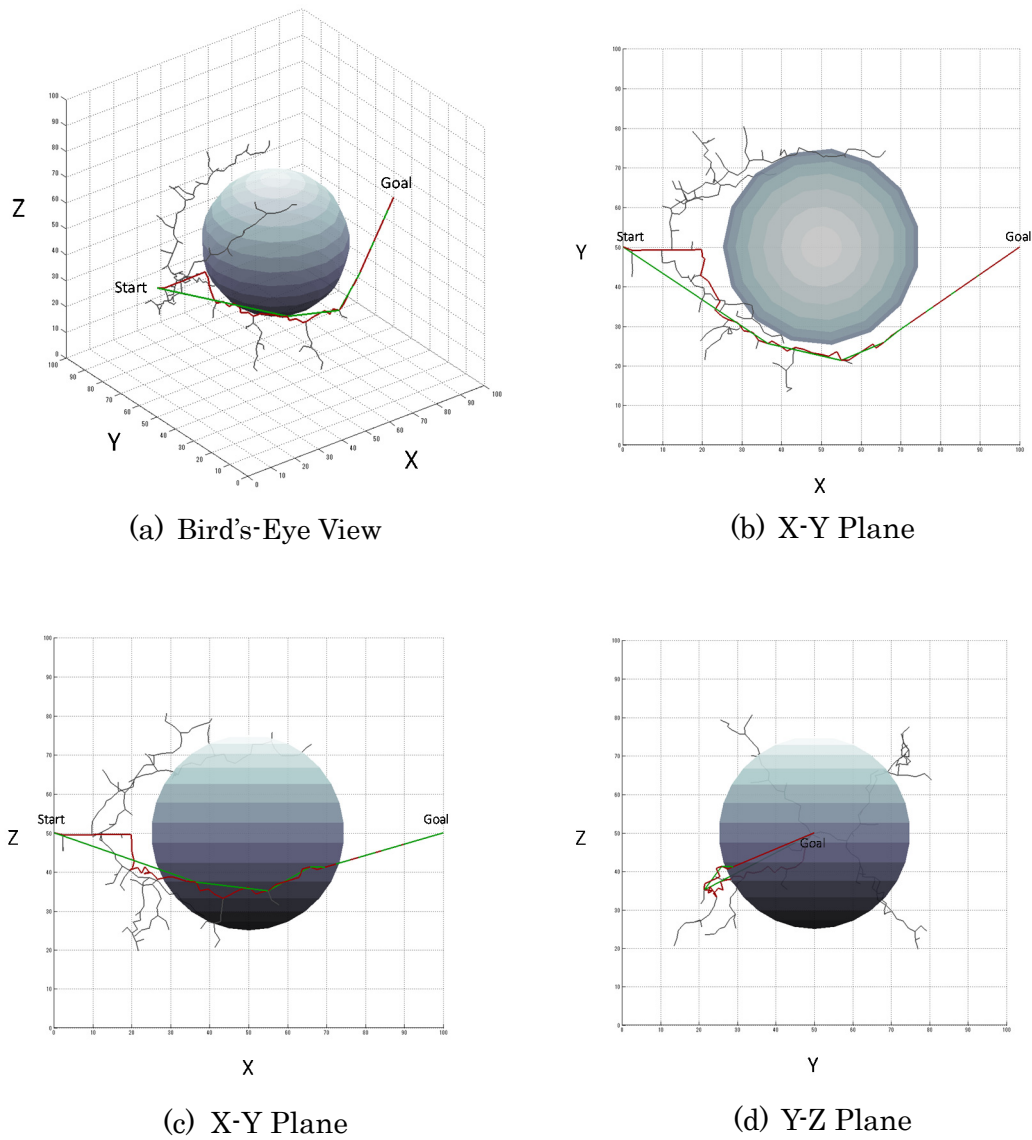
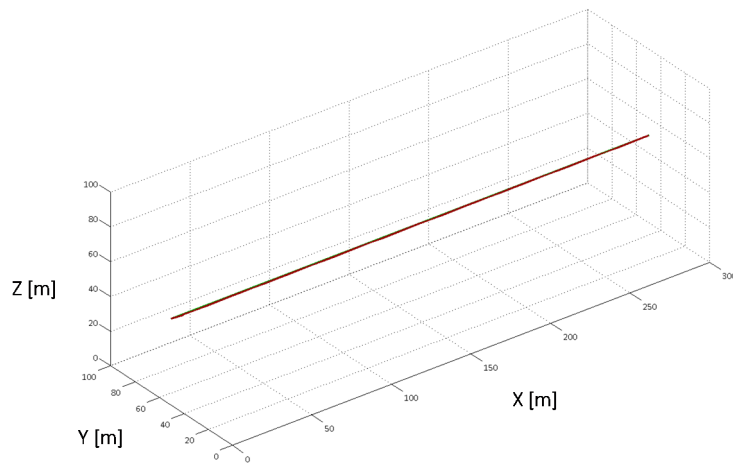


Fig. 2.3.13 Simulation Result of RRT with Shortcut Algorithm

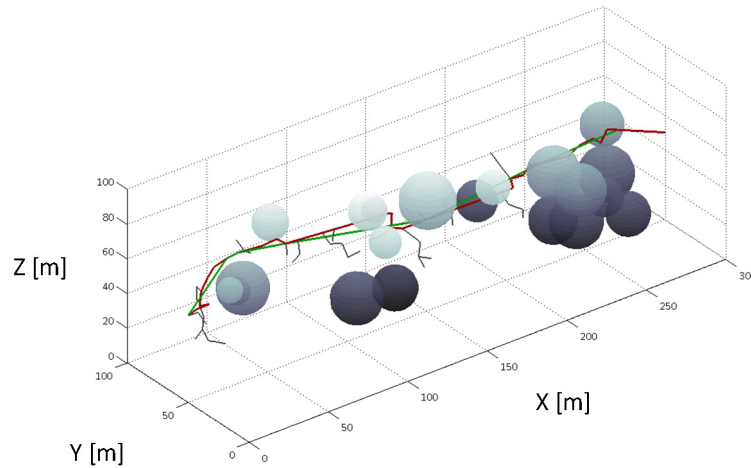
以上のアルゴリズム構成を軌道プランナに適用する手法とし，本論文では，この経路探索フェーズを包括して“ランダム探索”と呼ぶ．

2.4 ランダム探索のシミュレーション

前節で構成したランダム探索アルゴリズムを用いて、障害物の数や位置を変化させた経路探索シミュレーションを行う。障害物の数は、0 個から 100 個まで 20 個刻みで増加させる。また、障害物の寸法や位置はランダムに配置している。各障害物数に対してランダム探索を実行した結果を Fig. 2.4.1 に示す。これらの結果より、障害物が多数存在する環境においても問題なく障害物を回避する経路探索が行われることが分かった。

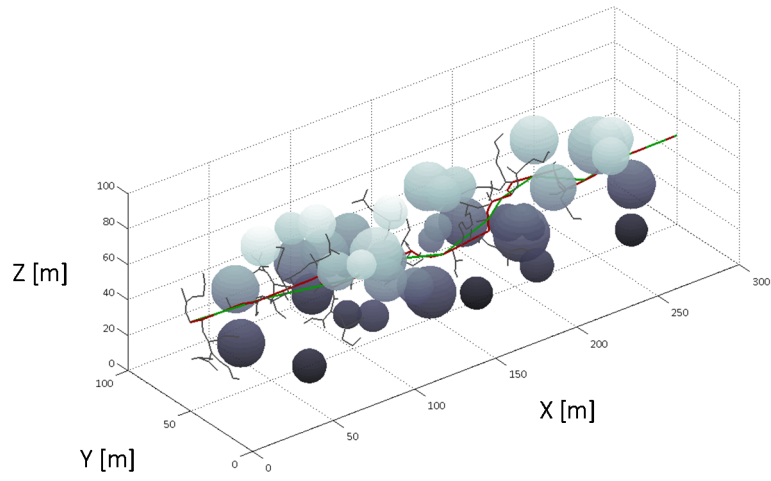


(a) Obstacle Number 0

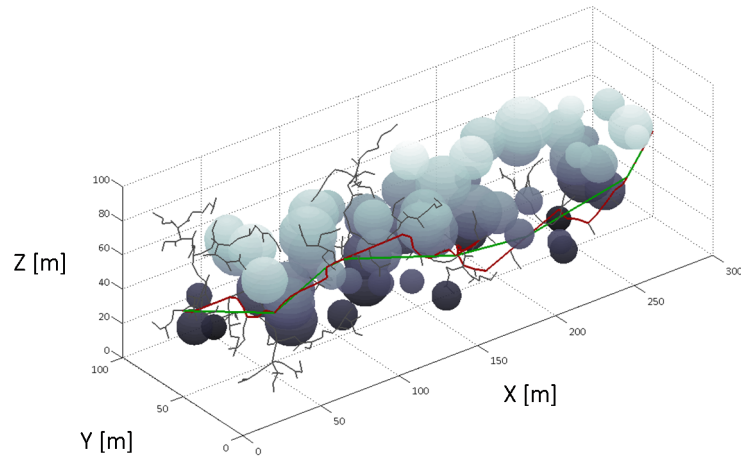


(b) Obstacle Number 20

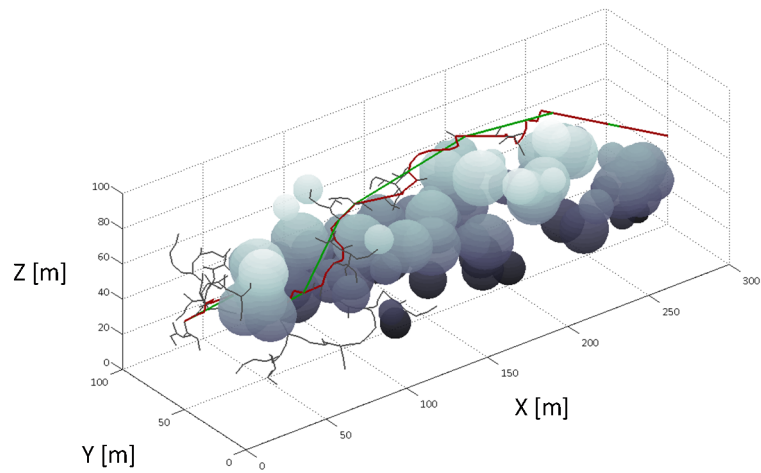
Fig. 2.4.1 Simulation Results of Random Search
(to be continued)



(c) Obstacle Number 40

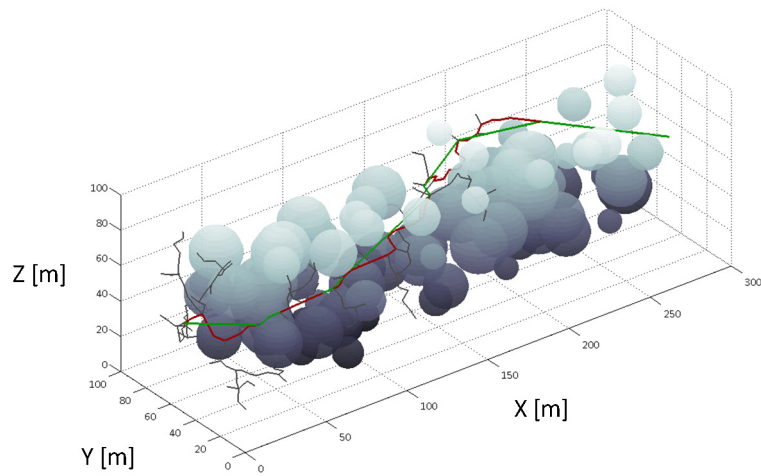


(d) Obstacle Number 60



(e) Obstacle Number 80

Fig. 2.4.1 Simulation Results of Random Search
(to be continued)



(f) Obstacle Number 100

Fig. 2.4.1 Simulation Results of Random Search
(to be concluded)

さらに、各障害物数におけるランダム探索の反復回数（ランダム点の数）を比較する（Fig. 2.4.2）。ここでは、各環境においてランダム探索を 10 回実行し、その平均値を示している。図の結果より、障害物数が増加するにつれて、ほぼ比例して反復回数が増加することが分かった。

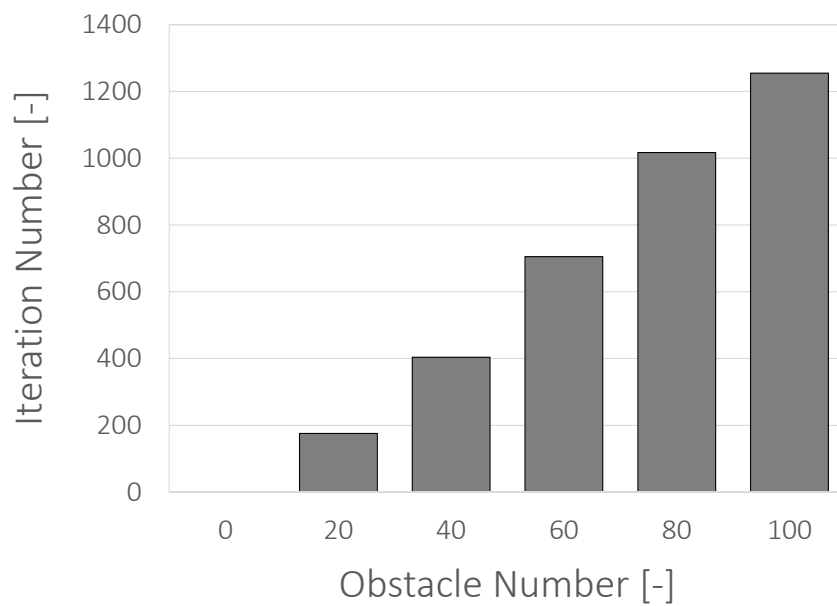


Fig. 2.4.2 Iteration Number of Each Obstacle Environment

2.5 本章のまとめ

本章では、軌道プランナに適用するランダム探索の手法を構築するため、各経路探索アルゴリズムのシミュレーションを行い、比較検討した。また、RRT を航空宇宙機の誘導に適したアルゴリズムにするため、障害物回避、二点間結合、経路簡略化アルゴリズムを付与した。次に、障害物を段階的に増加させた場合のランダム探索のシミュレーションを行い、障害物の増加に対しても対応できるアルゴリズムであることが検証できた。さらに、障害物数と反復回数の相関を取得し、障害物の増加が探索に与える影響を評価した。

第3章 力学フィルタと基礎的な軌道プランナの構築

3.1 はじめに

力学フィルタの概念は、山根らが研究したヒューマノイドロボットの運動生成法として1999年に提唱された^[58]。二足歩行を行うロボットは、力学的に不安定な事象が多いため、従来は予め設計された所定の運動を行うだけのもや、学習法を用いてトライ＆エラーを繰り返し、所望の動作を獲得する研究がほとんどであった。しかし、それらはロボットの運動の多様性や即応性に課題があるとされる。そこで、汎用かつ実時間で実行可能な運動生成法として提案された手法が力学フィルタである。力学フィルタは、ロボットに要求する運動が力学的に実現不可能な場合、その要求に対して適切な修正をリアルタイムで加えることで、ロボットにとって力学的な整合性を満足する運動に変換する機能を果たした。例えば、モーションキャプチャーにより得られた人間の動きをロボットが即座に真似をする場合、人間とロボットのダイナミクスの相違によってうまく動作ができなかったり、転倒する場合が考えられるが、力学フィルタによりロボットにとって無理のない動作にリアルタイムで修正され、安全に動作させることが可能となる。

本研究では、この力学フィルタの概念に着目した。前章のランダム探索で得られる経路は折れ曲がった経路を成し、それに沿って機体が飛ぶことは不可能であるため、障害物との衝突や墜落などの危険がある。本研究の力学フィルタは、この経路に対して航空宇宙機の力学的整合性を与え、飛べる軌道へと変換する機能を果たす (Fig. 3.1.1)。

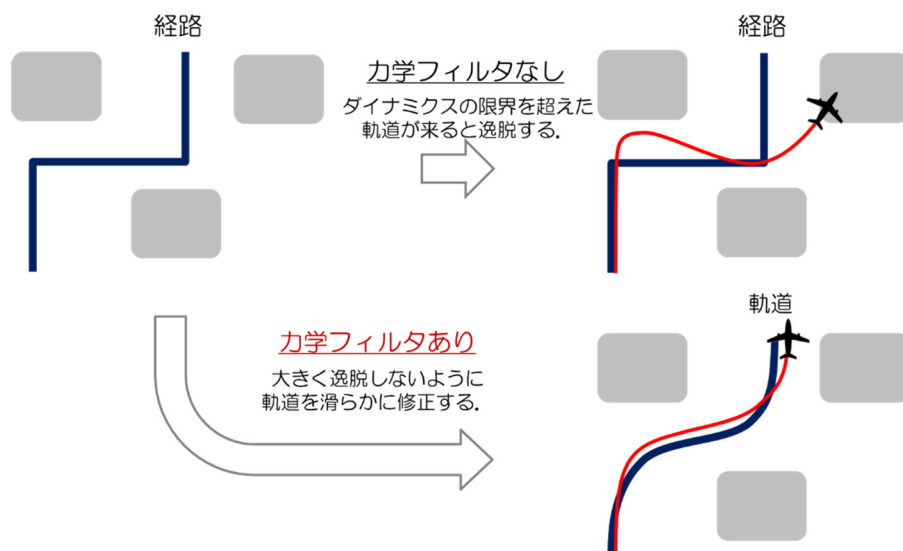


Fig. 3.1.1 Conceptual Diagram of Dynamics Filter

3.2 力学フィルタの設計概念

本研究の力学フィルタの主な目的は、ランダム探索によって与えられる飛行困難な障害物回避経路を、航空宇宙機の力学モデルに基づいてフィルタリングすることで飛行可能な軌道に変換することである。しかし本研究では、軌道の生成のみを行うのではなく、制御に必要なコマンドの生成までを行う。従来の誘導アルゴリズムでは、緯度、経度、高度（または経路角、方位角）などの軌道情報をコマンドとして制御器に与えるが、それらの情報を制御器でどのように利用するかは、重要な検討課題となる（Fig. 3.2.1）。

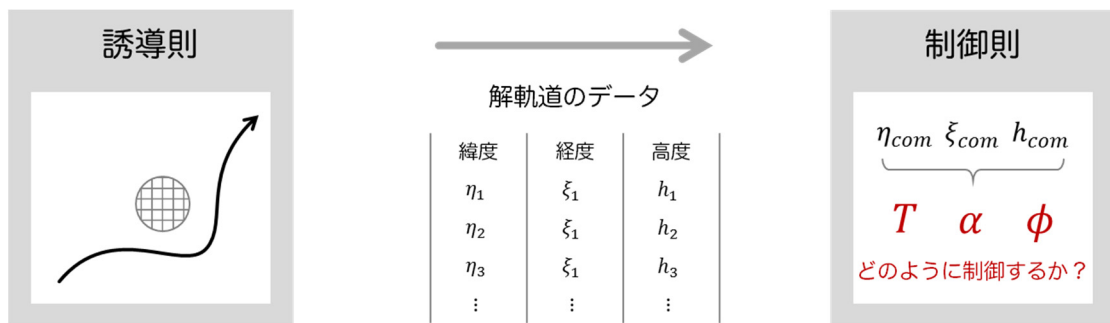


Fig. 3.2.1 Problem of Trajectory Generation and Guidance Command

一方で提案する力学フィルタの手法では、機体モデルの微分方程式を数値解析的に計算することで、解軌道の座標データだけでなく、解軌道を実現するための時系列コマンドも同時に導く（Fig. 3.2.2）。そのため制御器には、誘導から出力されるコマンドをそのまま制御入力に用いることができ、制御設計の負担が大幅に軽減されることが期待できる。このように、誘導アルゴリズムで制御入力に直接対応したコマンドを生成できる手法は、本研究の大きな利点の一つである。



Fig. 3.2.2 Dynamics Filter and Guidance Command

3.3 対象モデル

3.3.1 運動方程式の導出

本節では，航空宇宙機の運動のモデル化を行う．火星探査航空機は，プロペラを推進器とする固定翼機であり，有翼ロケットも滑空飛行を考慮するため，それらの運動モデルは一般的な航空機と同じとして扱える．また，将来的に多くの UAV への汎用性を考慮しても一般的な運動モデルを採用することが望ましい．

航空機の運動モデルは，通常 3 軸方向の並進運動および 3 軸周りの回転運動の 6 自由度非線形常微分方程式として表現される．そして特に制御器の設計などにおいては，平衡点周りの線形近似式を用いて，縦系の運動と横方向系の運動に切り分けて扱われるのが一般的である．しかし，姿勢の制御と比較して時定数が十分遅いとみなせる軌道の変化において，それほど厳密な運動を計算する必要性は低く，リアルタイム性を満足するためにも計算負荷を削減する工夫が必要であることから，機体の運動は質点系として扱い並進運動のみを考える．機体の運動は地球局所固定座標系として表し，方位角は磁北を零度として時計回りに成す角度を正に取る．また，経路角 γ は，機体の速度ベクトル V が系の水平面（ $\xi - \eta$ 平面）との成す角度とし，上昇を正とする．Fig. 3.3.1 に座標系の定義を示す．

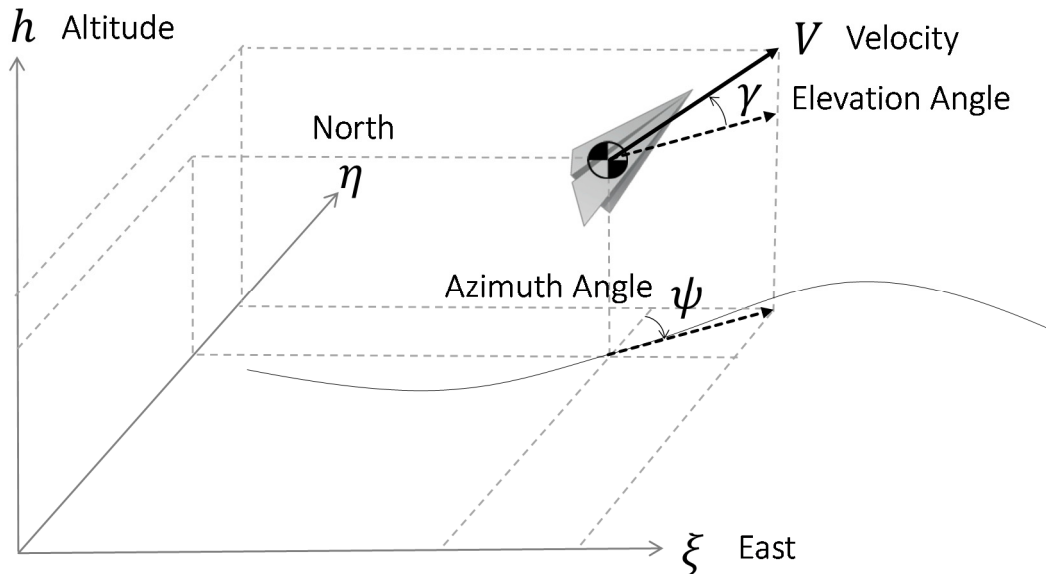


Fig. 3.3.1 Coordinate System Definition

次に，速度変化，経路角変化，方位角変化に関する連立常微分方程式の導出を以下で行う．ここで，有翼ロケットの場合は滑空飛行を行うので，推力の項は以降ゼロとして取り扱う．まず，速度ベクトル V を通る $\xi - \eta$ 平面に垂直な面で縦の質点系運動モデルを Fig. 3.3.2 のように考える．

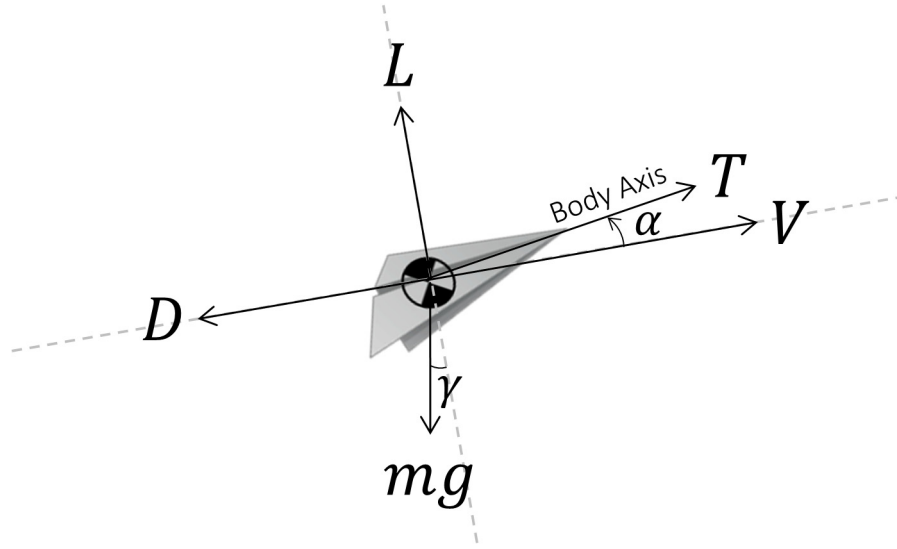


Fig. 3.3.2 Longitudinal Dyanmics

ここで，時間 t ，質量 m ，重力 g ，速度 V ，推力 T ，揚力 L ，抗力 D ，迎角 α ，経路角 γ とする．この時，機体の速度方向の運動方程式は釣り合いより Eq. (3.1) となる．

$$m \frac{dV}{dt} = T \cos \alpha - D - mg \sin \gamma \quad (3.1)$$

速度方向に垂直なベクトル方向の運動は，経路変化を円弧の運動と近似した場合，曲率半径を r_γ ，経路角速度を ω_γ ，向心加速度を a_γ と定義すると， a_γ は Eq. (3.2) のように表現でき，式変形すると Eq. (3.3) となる．このとき，軌道の経路角変化は向心加速度で生じるので，経路角の時間変化に対する方程式は Eq. (3.4) と書ける．

$$a_\gamma = r_\gamma \omega_\gamma^2 \quad (3.2)$$

$$a_\gamma = V \frac{d\gamma}{dt} \quad (\because V = r_\gamma \omega_\gamma, \omega_\gamma = \frac{d\gamma}{dt}) \quad (3.3)$$

$$mV \frac{d\gamma}{dt} = T \sin \alpha + L - mg \cos \gamma \quad (3.4)$$

方位角変化に対する向心加速度 a_ψ は同様に Eq. (3.5)と書けるが，ここではバンク角を考慮していないため，方位角の時間変化はゼロとなる．

$$a_\psi = V \cos \gamma \frac{d\psi}{dt} \quad (3.5)$$

上記の各運動方程式をバンク角を考慮して拡張すると下記のように書ける．

$$m \frac{dV}{dt} = T \cos \alpha - D - mg \sin \gamma \quad (3.6)$$

$$mV \frac{d\gamma}{dt} = T \sin \alpha + L \cos \phi - mg \cos \gamma \quad (3.7)$$

$$mV \cos \gamma \frac{d\psi}{dt} = L \sin \phi \quad (3.8)$$

ただし，揚力および抗力は大気密度 ρ ，主翼面積 S ，揚力係数 C_L ，抗力係数 C_D とすると

$$L = \frac{1}{2} \rho V^2 S C_L \quad (3.9)$$

$$D = \frac{1}{2} \rho V^2 S C_D \quad (3.10)$$

であり，上式を Eq. (3.6)~ Eq. (3.8)に代入すると速度，経路角，方位角変化に関する方程式が得られる．

$$m \frac{dV}{dt} = T \cos \alpha - \frac{1}{2} \rho V^2 S C_D - mg \sin \gamma \quad (3.11)$$

$$mV \frac{d\gamma}{dt} = T \sin \alpha + \frac{1}{2} \rho V^2 S C_L \cos \phi - mg \cos \gamma \quad (3.12)$$

$$mV \cos \gamma \frac{d\psi}{dt} = \frac{1}{2} \rho V^2 S C_L \sin \phi \quad (3.13)$$

ここで、揚力係数および抗力係数はそれぞれ以下の迎角の関数として表せる。ただし、 c_{L1} , c_{L2} , c_{D1} , c_{D2} , c_{D3} は各機体の空力特性を表す定数である。

$$C_L = c_{L1} + c_{L2} \alpha \quad (3.14)$$

$$C_D = c_{D1} + c_{D2} \alpha + c_{D3} \alpha^2 \quad (3.15)$$

また、高度、ダウンレンジ、クロスレンジの方程式は以下の通りである。

$$\frac{dh}{dt} = V \sin \gamma \quad (3.16)$$

$$\frac{d\xi}{dt} = V \cos \gamma \sin \psi \quad (3.17)$$

$$\frac{d\eta}{dt} = V \cos \gamma \cos \psi \quad (3.18)$$

以上を機体の運動モデルとして力学フィルタを構成する。

3.3.2 機体諸元

ここでは，力学フィルタで用いる機体モデルの主要な諸元および空力特性を述べる．

3.3.2.1 火星探査航空機

火星探査航空機 WG で検討が進められている機体の外観と主要諸元を Fig. 3.3.3 および Table 3.3.1 に記す．

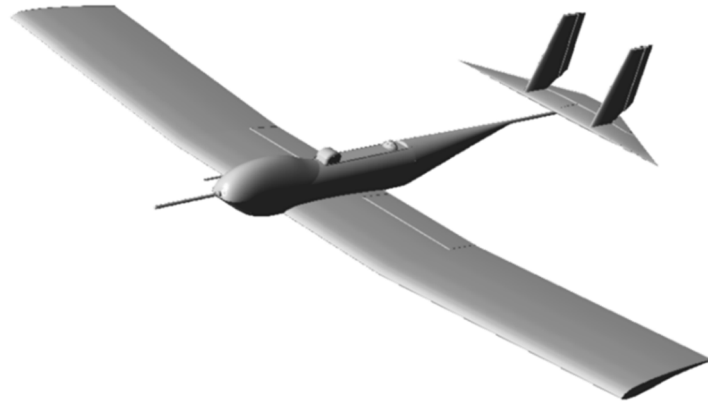


Fig. 3.3.3 Mars Exploration Aircraft Model

Table 3.3.1 Mars Exploration Aircraft Specifications

Wing Span	\bar{b}	[m]	2.42
Wing Area	S	[m ²]	1.15
Mass	m	[kg]	4.24
Inertia Moment of x-Axis	I_{xx}	[kg · m ²]	0.455
Inertia Moment of y-Axis	I_{yy}	[kg · m ²]	1.200
Inertia Moment of z-Axis	I_{zz}	[kg · m ²]	1.609

また，ISAS により同機体の全機モデルを用いた風洞試験が行なわれ，各空力係数が取得された．揚力係数，抗力係数の測定結果を Fig. 3.3.4 に示す．なお，同風洞試験は火星環境を模擬したレイノルズ数 33,000 の結果である．

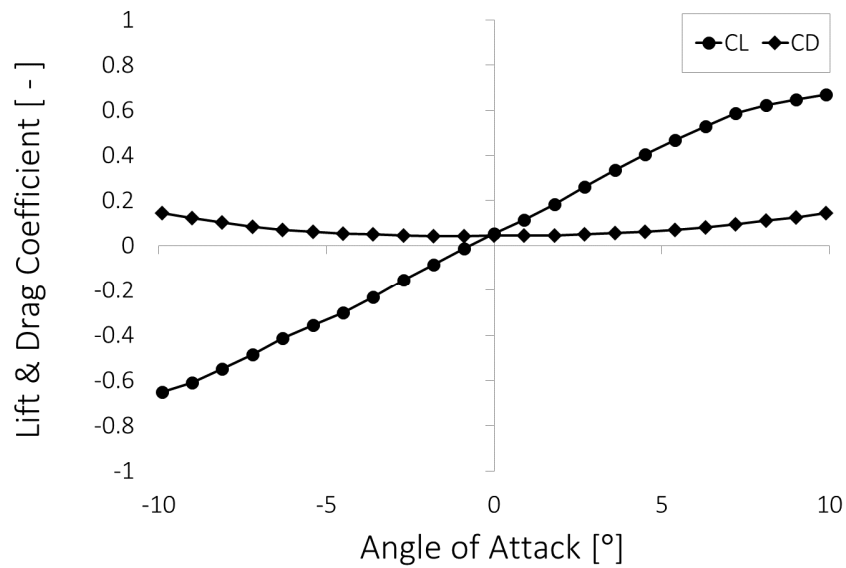


Fig. 3.3.4 Lift and Drag Coefficients of Mars Exploration Aircraft

3.3.2.2 有翼ロケット実験機

九州工業大学が開発する小型有翼ロケット実験機 WIRE#014 の機体外観および諸元を Fig. 3.3.5 と Table 3.3.2 に示す. 同機体の空力モデルは, ISAS により開発された HIMES と同じであり, 当時, 小型模型により計測された風洞試験結果を流用する. マッハ数 0.1 における揚力係数, 抗力係数の測定結果を Fig. 3.3.6 に示す.



Fig. 3.3.5 WIRE#014 Model

Table 3.3.2 WIRES Specifications

Wing Span	\bar{b}	1.08	[m]
Wing Area	S	0.47	[m ²]
Mass	m	40.0	[kg]
Inertia Moment of x-Axis	I_{xx}	0.548	[kg · m ²]
Inertia Moment of y-Axis	I_{yy}	5.274	[kg · m ²]
Inertia Moment of z-Axis	I_{zz}	5.323	[kg · m ²]

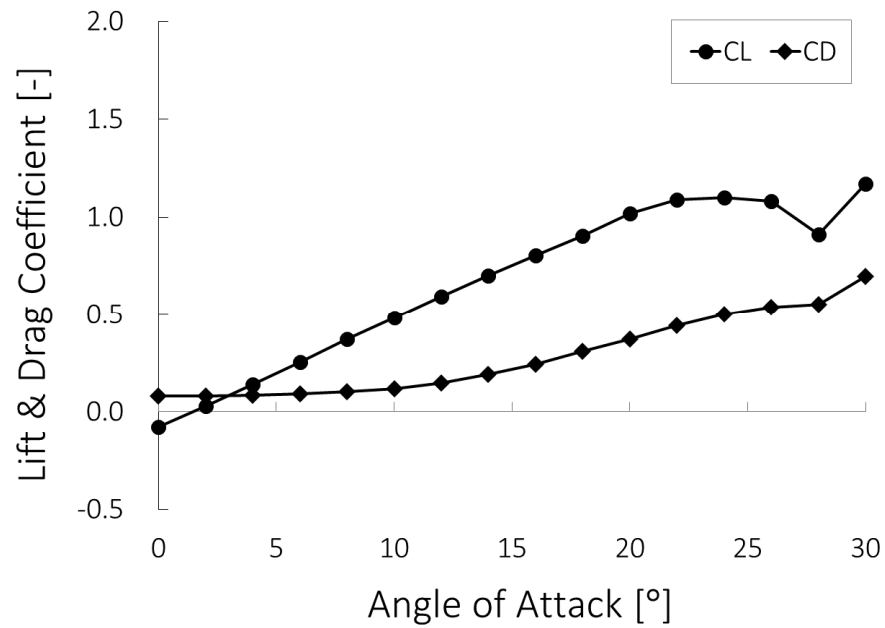


Fig. 3.3.6 Lift and Drag Coefficients of WIRES

3.4 力学フィルタの構築

3.4.1 運動方程式の線形化

航空機のダイナミクスは極めて強い非線形性を持つため、運動の評価にはしばしば釣り合い状態（定常飛行）からの微小擾乱運動を仮定して線形近似された運動方程式が用いられる。

ここで、参考として前節の運動モデルに対して微小擾乱を仮定した線形化式を記載する。まず、推力 T 、バンク角 ϕ 、迎角 α 、速度 V 、経路角 γ 、方位角 ψ 、揚力係数 C_L 、抗力係数 C_D についてそれぞれ平衡点を添え字 0 とし、微小変化量を δ とすると定常飛行時からの微小擾乱は式 Eq. (3.19)のように書ける。Eq. (3.20)において $C_{L\alpha}$ 、 $C_{D\alpha}$ は平衡点周りの迎角変化における揚力係数および抗力係数変化の傾きを表す。

$$\begin{bmatrix} T \\ \phi \\ \alpha \\ V \\ \gamma \\ \psi \end{bmatrix} = \begin{bmatrix} T_0 + \delta T \\ \phi_0 + \delta\phi \\ \alpha_0 + \delta\alpha \\ V_0 + \delta V \\ \gamma_0 + \delta\gamma \\ \psi_0 + \delta\psi \end{bmatrix} \quad (3.19)$$

$$\begin{bmatrix} C_L \\ C_D \end{bmatrix} = \begin{bmatrix} C_{L0} + C_{L\alpha}\delta\alpha \\ C_{D0} + C_{D\alpha}\delta\alpha \end{bmatrix} \quad (3.20)$$

このとき、定常飛行の条件により $\phi_0 = 0, \gamma_0 = 0$ を与えて式を整理し、行列表示を用いると次式のような状態方程式で書ける。

$$\begin{bmatrix} \delta\dot{V} \\ \delta\dot{\psi} \\ \delta\dot{\gamma} \end{bmatrix} = \mathbf{M}_1 \begin{bmatrix} \delta V \\ \delta\psi \\ \delta\gamma \end{bmatrix} + \mathbf{M}_2 \begin{bmatrix} \delta T \\ \delta\phi \\ \delta\alpha \end{bmatrix} + \mathbf{M}_3 \quad (3.21)$$

ここで、 \mathbf{M}_1 、 \mathbf{M}_2 はそれぞれ状態量と入力にかかる係数行列であり、 \mathbf{M}_3 は偏差として残る係数行列である。

$$\mathbf{M}_1 = \begin{bmatrix} -\frac{\rho S V_0 C_{D0}}{m} & 0 & -g \\ 0 & 0 & 0 \\ \frac{\rho S C_{L0}}{m} & 0 & 0 \end{bmatrix} \quad (3.22)$$

$$\mathbf{M}_2 = \begin{bmatrix} \frac{\cos \alpha_0}{m} & 0 & -\frac{1}{m} \left(T_0 \sin \alpha_0 + \frac{\rho S V_0^2 C_{D\alpha}}{2} \right) \\ 0 & \frac{1}{m V_0} \left(T_0 \sin \alpha_0 + \frac{\rho S V_0^2 C_{L0}}{2} \right) & 0 \\ \frac{\sin \alpha_0}{m V_0} & 0 & \frac{1}{m V_0} \left(T_0 \cos \alpha_0 + \frac{\rho S V_0^2 C_{L\alpha}}{2} \right) \end{bmatrix} \quad (3.23)$$

$$\mathbf{M}_3 = \begin{bmatrix} \frac{1}{m} \left(T_0 \cos \alpha_0 - \frac{\rho S V_0^2 C_{D0}}{2} \right) \\ 0 \\ \frac{1}{m V_0} \left(T_0 \sin \alpha_0 + \frac{\rho S V_0^2 C_{L0}}{2} - mg \right) \end{bmatrix} \quad (3.24)$$

また、釣り合い飛行を考えると Eq. (3.11)および Eq. (3.12)は

$$0 = T_0 \cos \alpha_0 - \frac{1}{2} \rho V_0^2 S C_{D0} \quad (3.24)$$

$$0 = T_0 \sin \alpha_0 - \frac{1}{2} \rho V_0^2 S C_{L0} - mg \quad (3.25)$$

となり、式 Eq. (3.21)に代入すると、 $\mathbf{M}_1, \mathbf{M}_2$ は簡潔に表わせて、偏差行列 \mathbf{M}_3 は消去される。以上より、機体のダイナミクスが以下の簡単な線形多入力多出力システムの状態方程式として書けるようになる。

$$\begin{bmatrix} \delta \dot{V} \\ \delta \dot{\psi} \\ \delta \dot{\gamma} \end{bmatrix} = \mathbf{M}_1' \begin{bmatrix} \delta V \\ \delta \psi \\ \delta \gamma \end{bmatrix} + \mathbf{M}_2' \begin{bmatrix} \delta T \\ \delta \phi \\ \delta \alpha \end{bmatrix} \quad (3.26)$$

$$\mathbf{M}_1' = \begin{bmatrix} -\frac{\rho S V_0 C_{D0}}{m} & 0 & -g \\ 0 & 0 & 0 \\ \frac{\rho S C_{L0}}{m} & 0 & 0 \end{bmatrix} \quad (3.27)$$

$$\mathbf{M}_2' = \begin{bmatrix} \frac{\cos \alpha_0}{m} & 0 & \frac{\rho S V_0^2}{2m} (C_{L0} - C_{D\alpha}) - g \\ 0 & \frac{g}{V_0} & 0 \\ \frac{\sin \alpha_0}{m V_0} & 0 & \frac{\rho S V_0}{2m} (C_{L\alpha} + C_{D0}) \end{bmatrix} \quad (3.28)$$

本来与えられていた Eq. (3.12)~ Eq. (3.14)の非線形運動方程式では，入力項 (T, α, ϕ) が各微分方程式の状態変数と共に複雑に連成してるため，入力に用いる制御設計が極めて煩雑になる．しかし，このように線形化することで入力を他の状態量から切り分けることが可能になり，出力に対する入力の影響が明確になるため取り扱いが容易となる．

以上が線形化式の特徴である．しかし，線形化された方程式は本研究が目指している複雑な障害物回避軌道の生成に適用させることは難しい．なぜなら，航空宇宙機が3次元的に障害物を回避するためには速度や姿勢を大きく変化させる必要があり，定常飛行からの微少変動という仮定を満たさないためである．そこで，本研究では前節で与えられた非線形運動モデルを平衡点周りでの線形化を行うことなくそのまま用いる．そして所望の運動を獲得するために制御を行えるよう工夫する．そこで，ロボティクスの分野で特にマニピュレータの動作計画法として研究される逆運動学理論^[59] (Inverse Kinematics) を用いて力学フィルタを構築する．

3.4.2 逆運動学

マニピュレータの動作計画法では，アームの関節角度を決めて，各リンクのモデルに基づき手先の位置と姿勢を算出する手法を一般的に順運動学と呼ぶ．それに対して，手先の目標位置が先に与えられる場合，目標位置を満たすような関節角度を後から逆算する手法を逆運動学と呼び，あらかじめ通るべき場所が指定されるような障害物回避問題などに有効な手段として知られる．順運動学では，関節角度が決まれば手先の位置と姿勢は一意に決まる．一方逆運動学では，目標の値を満たす解が複数存在する場合が多い．例えば，3自由度を持つ2次元平面運動のロボットアームでは，目標の手先位置と姿勢が同じ場合に2つの解を持つ．Fig. 3.4.1に順運動学と逆運動学の模式図を示す．一般に逆運動学とは動力学を含まないが，本研究の力学フィルタでは，この逆運動学の考え方を適用して微分方程式を数値解析することにより所望の解を求める手法に着目した．

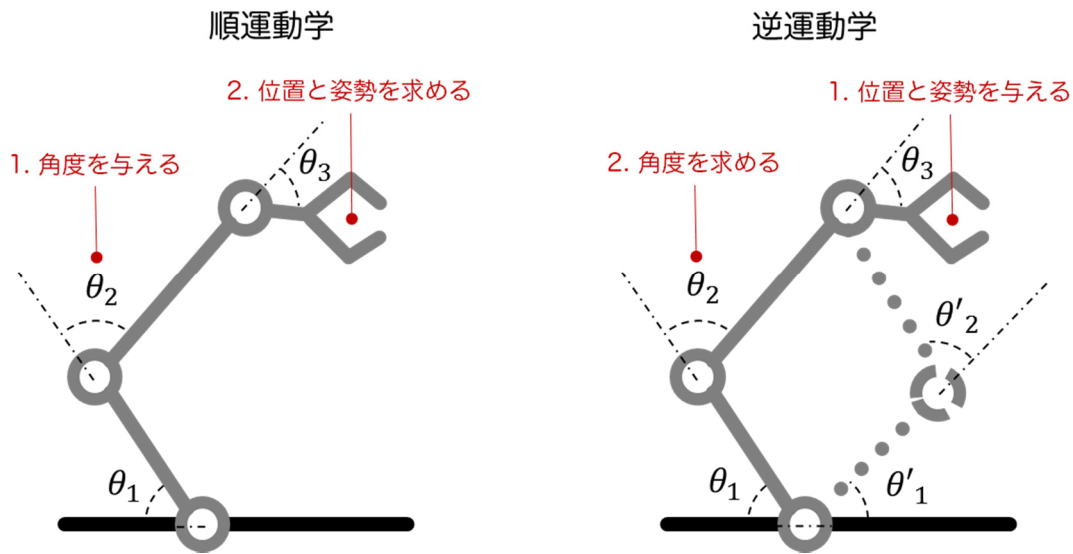


Fig. 3.4.1 Forward Kinematics and Inverse Kinematics

3.4.3 逆運動学を用いた力学フィルタ

ここまで、逆運動学は予め目標となる位置や姿勢が与えられる場合に、その目標へ収束させる方法として有効であることを説明した。ここでは、逆運動学の考え方を航空宇宙機の運動方程式に適用させた力学フィルタのアルゴリズムについて述べる。

はじめに、力学フィルタを簡潔に表すブロック線図を Fig. 3.4.2 に示す。力学フィルタは、図に示すように極めて単純な閉ループ制御系を構成している。Plant には、3.3 節で述べた質点系の運動方程式が用いられる。力学フィルタでは、制御器を持つ運動方程式を逐次計算することにより軌道を生成し、さらに目標値に近づくために必要な入力の時系列データを獲得する。

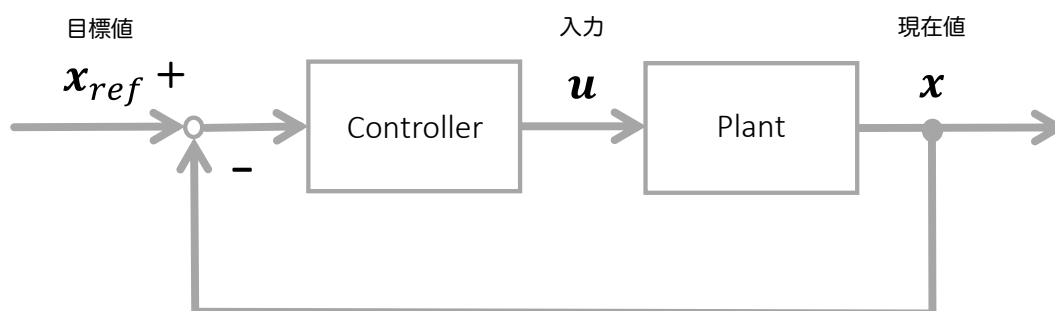


Fig. 3.4.2 Block Diagram of Dynamics Filter

次に、力学フィルタの計算式を示す。いま、力学フィルタで用いる状態量ベクトル \mathbf{x} 、入力ベクトル \mathbf{u} 、関数ベクトル \mathbf{f} をそれぞれ下記のように定める。

$$\mathbf{x} = \begin{Bmatrix} V \\ \gamma \\ \psi \end{Bmatrix} \quad (3.29)$$

$$\mathbf{u} = \begin{Bmatrix} T \\ \alpha \\ \phi \end{Bmatrix} \quad (3.30)$$

$$\mathbf{f} = \begin{Bmatrix} \frac{1}{m} \left(T \cos \alpha - \frac{1}{2} \rho V^2 S C_D - mg \sin \gamma \right) \\ \frac{1}{mV} \left(T \sin \alpha + \frac{1}{2} \rho V^2 S C_L \cos \phi - mg \cos \gamma \right) \\ \frac{1}{mV \cos \gamma} \left(\frac{1}{2} \rho V^2 S C_L \sin \phi \right) \end{Bmatrix} \quad (3.31)$$

このとき、機体の運動方程式は

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (3.32)$$

と表せる。Eq. (3.31)、Eq. (3.32)から分かるように入力 \mathbf{u} は、状態量 \mathbf{x} に対して複数の方程式に連成しているため、制御器を一意に決定することが困難である。そこで、入力項を分離するため、両辺微分すると Eq. (3.32)は

$$\ddot{\mathbf{x}} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} + \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} \frac{d\mathbf{u}}{dt} \quad (3.33)$$

$$\ddot{\mathbf{x}} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \dot{\mathbf{x}} + \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} \dot{\mathbf{u}} \quad (3.34)$$

となる。ここで、本来ならば Eq. (3.34)の右辺第一項は、無視すべきではないが本論文の力学フィルタでは、省略して Eq. (3.35)とする。これで、入力 \mathbf{u} を一回微分 $\dot{\mathbf{u}}$ として分離可能となった。

$$\ddot{\mathbf{x}} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} \dot{\mathbf{u}} \quad (3.35)$$

ただし，ヤコビアン \mathbf{J} は逐次計算の過程における現在値を用いて計算される．

$$\begin{aligned}\mathbf{J}(\mathbf{x}, \mathbf{u}) &= \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} = \begin{bmatrix} \frac{\partial \dot{V}}{\partial T} & \frac{\partial \dot{V}}{\partial \alpha} & \frac{\partial \dot{V}}{\partial \phi} \\ \frac{\partial \dot{\gamma}}{\partial T} & \frac{\partial \dot{\gamma}}{\partial \alpha} & \frac{\partial \dot{\gamma}}{\partial \phi} \\ \frac{\partial \dot{\psi}}{\partial T} & \frac{\partial \dot{\psi}}{\partial \alpha} & \frac{\partial \dot{\psi}}{\partial \phi} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{m} \cos \alpha & -\frac{1}{m} \left\{ T \sin \alpha + \frac{1}{2} \rho V^2 S_w (c_{D2} + 2c_{D3} \alpha) \right\} \\ \frac{\sin \alpha}{mV} & \frac{1}{mV} \left\{ T \cos \alpha + \frac{1}{2} \rho V^2 S_w c_{L2} \cos \phi \right\} \\ 0 & \frac{\rho V^2 S_w c_{L2} \sin \phi}{2mV \cos \gamma} \\ & 0 \\ & -\frac{1}{mV} \left\{ T \sin \alpha + \frac{1}{2} \rho V^2 S_w (c_{L1} + c_{L2} \alpha) \sin \phi \right\} \\ & \frac{\rho V^2 S_w (c_{L1} + c_{L2} \alpha) \cos \phi}{2mV \cos \gamma} \end{bmatrix} \quad (3.36)\end{aligned}$$

次に逆運動学では，Eq. (3.35)左辺の $\ddot{\mathbf{x}}$ に対して微少運動変化量を予め定めておく必要がある．力学フィルタでは， $\ddot{\mathbf{x}}$ に制御器を導入することで目標値に近づけるための変化量を与える．ここでは，一例として目標速度 V_{ref} ，目標経路角 γ_{ref} ，目標方位角 ψ_{ref} を用いて状態量 \mathbf{x} の誤差に対する P 制御系を構成する．ここで， \mathbf{K} は制御ゲインである．

$$\ddot{\mathbf{x}} = \mathbf{K}(\mathbf{x}_{ref} - \mathbf{x}) = \begin{bmatrix} K_v(V_{ref} - V) \\ K_\gamma(\gamma_{ref} - \gamma) \\ K_\psi(\psi_{ref} - \psi) \end{bmatrix} \quad (3.37)$$

Eq. (3.36)と Eq. (3.37)を用いると，Eq. (3.35)から与えるべき入力変化量が逆算できる．この過程で，描かれる軌道を実現するための入力時系列データが生成される．

$$\dot{\mathbf{u}} = \mathbf{J}^{-1}(\mathbf{x}, \mathbf{u}) \ddot{\mathbf{x}} = \mathbf{J}^{-1}(\mathbf{x}, \mathbf{u}) \mathbf{K}(\mathbf{x}_{ref} - \mathbf{x}) \quad (3.38)$$

次に， $\dot{\mathbf{u}}$ を積分して得た新たな \mathbf{u} を Eq. (3.32)に代入し，運動方程式を計算する．

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \frac{1}{m} \left(T \cos \alpha - \frac{1}{2} \rho V^2 S C_D - mg \sin \gamma \right) \\ \frac{1}{mV} \left(T \sin \alpha + \frac{1}{2} \rho V^2 S C_L \cos \phi - mg \cos \gamma \right) \\ \frac{1}{mV \cos \gamma} \left(\frac{1}{2} \rho V^2 S C_L \sin \phi \right) \end{bmatrix} \quad (3.39)$$

Eq. (3.37)を積分して得られる新たな \mathbf{x} の成分を Eq. (3.16), Eq. (3.17), Eq. (3.18)に代入し、微少時間に移動した機体の位置が算出できる．以上の計算過程で得られた情報をフィードバックし、逐次計算することで時間の経過とともに目標値に近づいていく運動が得られ、また必要となるコマンドの時系列データも同時に生成できる．以上が、力学フィルタの構成である．

本節の最後に、注意すべき点について述べる．逆運動学を解く際にヤコビアンの逆行列が計算できない（ランク落ちが発生する）場合、これを特異姿勢と呼び、そのような状態を通過とすると解が得られない問題が生じる．例えば、二足歩行ロボットが常に膝を曲げた姿勢で立っているのはこの特異姿勢を回避するためである．本研究の場合は、Eq. (3.36)に示すヤコビアンで逆行列が存在しないケースとして、質量 m や速度 V が無量大となる場合や、迎角 α が $90[\text{deg.}]$ かつバンク角 ϕ が $90[\text{deg.}]$ となる場合等が考えられるが、いずれも非現実的なケースであり特異姿勢に陥ることはないと思われる．また、迎え角やバンク角等のコマンドは、力学フィルタの計算ループ内にリミッターを組み込むことで、特異姿勢になる可能性を排除することができる．

また、本節の力学フィルタでは、Eq. (3.34)の右辺第一項を省略していることや、入力 \mathbf{u} を積分していることなどの課題を残す．そこで、これらの問題を解決する新たな力学フィルタの構成を付録に記載する．

3.5 ランダム探索と力学フィルタを組み合わせた基礎的な軌道プランナの構築

3.5.1 軌道プランナの概要

本節では、これまでに説明したランダム探索と力学フィルタを用いて障害物回避軌道を探索する基礎的な軌道プランナを構築する。本節の軌道プランナでは、はじめにランダム探索で障害物にぶつからない経路を探索し、その後に力学フィルタで飛べる軌道へと変換する。アルゴリズムの概念図を Fig. 3.5.1 に示す。

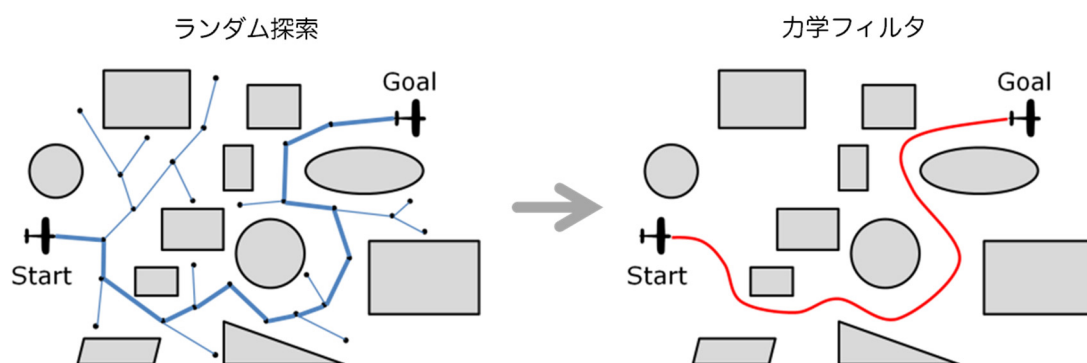


Fig. 3.5.1 Conceptual Diagram of Basic Trajectory Planner

3.5.2 アルゴリズムの流れ

基礎的な軌道プランナの計算手順を概念図を用いて順に説明する。まずランダム探索のフェーズでは、2章で説明したランダム探索アルゴリズムをそのまま実行する。ランダム探索によりゴールまで繋がるツリーの探索結果が得られたとき、ツリーを構成する各ノードはID、座標および距離の情報を持つ (Fig. 3.5.2)。ある地点のノードが持つ距離の情報は、図右下に示されるように、スタートからの各ノード間ユークリッド距離の総和である。次にそれらのツリーを目標経路として力学フィルタにより飛べる軌道を生成していく。はじめに適当な初期値 (リアルタイム誘導の際はセンサから取得した値) を与え、3.4節で取り上げた力学フィルタの計算手順に従って数値計算を行う。

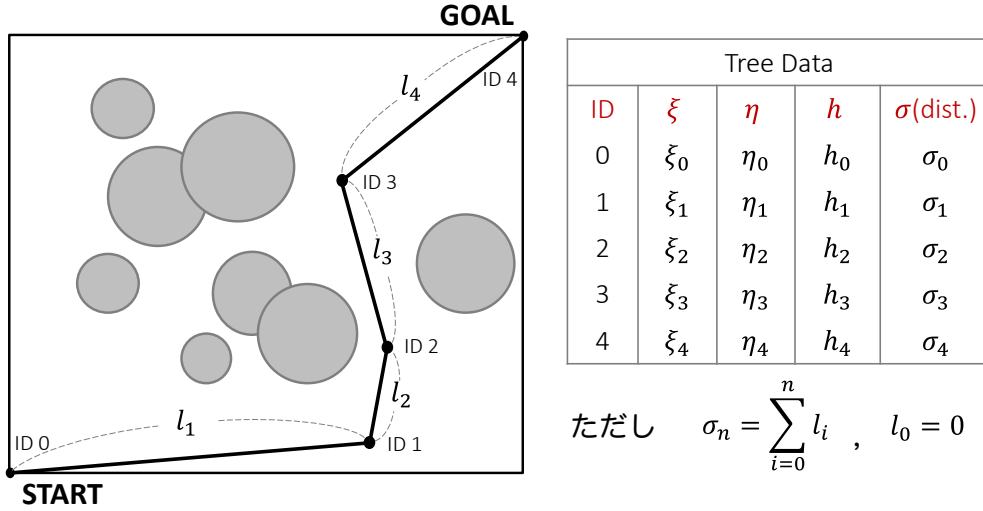


Fig. 3.5.2 Tree Construction of Basic Trajectory Planner

次に Fig. 3.5.3 には、力学フィルタに組み込む制御器を説明するため、力学フィルタの計算過程において ID1 を通過した後の状態を図示する。まず Eq. (3.35) の $\dot{\mathbf{x}}$ に与える値を生成する (Eq. (3.38))。ここで、Eq. (3.35) とは異なる目標値を用いているので \mathbf{r}_{ref} と書き換える。目標速度 V_{ref} は一定 (巡航速度) V_{const} とする。また、目標バンク角 ϕ_{ref} は Fig. 3.5.3 内に青丸で示される機体の現在位置から直線 ID1-ID2 に下ろした $\xi - \eta$ 平面における垂線の長さを d_{err} として算出し、 d_{err} およびその時間変化率を用いた PD 制御器により定める。Fig. 3.5.4 には目標 ID が $n+1$ のときの d_{err} の算出概念図を示す。 ID_n の $\xi - \eta$ 平面座標と機体位置 \mathbf{P} の $\xi - \eta$ 平面座標が成すベクトルを $\mathbf{t}_{\xi\eta}$ 、また ID_n の $\xi - \eta$ 平面座標と ID_{n+1} の $\xi - \eta$ 平面座標が成すベクトルを $\mathbf{s}_{\xi\eta}$ とし、さらにそれらの成す角を $\theta_{\xi\eta}$ とすると、 d_{err} は Eq. (3.39) によって定義される距離となる。目標バンク角 ϕ_{ref} と同様に、 h_{err} を用いた PD 制御器により目標迎え角 α_{ref} が定められる。 ID_n の $\xi - h$ 平面座標と機体位置 \mathbf{P} の $\xi - h$ 平面座標が成すベクトルを $\mathbf{t}_{\xi h}$ 、また ID_n の $\xi - h$ 平面座標と ID_{n+1} の $\xi - h$ 平面座標が成すベクトルを $\mathbf{s}_{\xi h}$ とし、それらの成す角を $\theta_{\xi h}$ とすると、 h_{err} は、Eq. (3.41) によって定義される。これらの各目標値と現在値の誤差をそれぞれ V_{err} 、 α_{err} 、 ϕ_{err} とする (Eq. (3.42))。各誤差を用いて、速度は PD 制御系、迎え角、バンク角は P 制御系 (Eq. (3.43)) として最終的に Eq. (3.35) に与える値 \mathbf{v} を生成する。このように、距離の誤差から一度姿勢角のコマンドを作り、そのコマンドに関して機体を制御させることでオーバーシュートしない安定した軌道を得ることができる。

Fig. 3.5.3 において目標 ID は 2 であるが，次の条件に基づき ID が更新される．目標 ID の更新は，現時点までの軌道計算結果として与えられる軌跡（青破線）から飛行距離（*Flight Dist.*）を算出し，スタートから現在の目標 ID までのエッジ長さ σ と比較する．飛行距離が σ より長くなったとき，目標 ID を次の ID に更新する（Fig. 3.5.3 右上）．

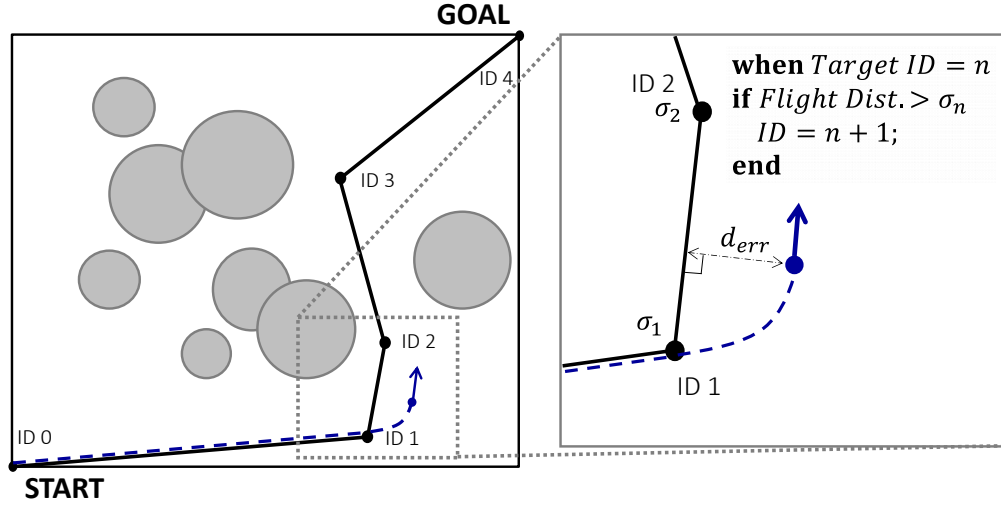


Fig. 3.5.3 Dynamics Filter Process of Basic Trajectory Planner

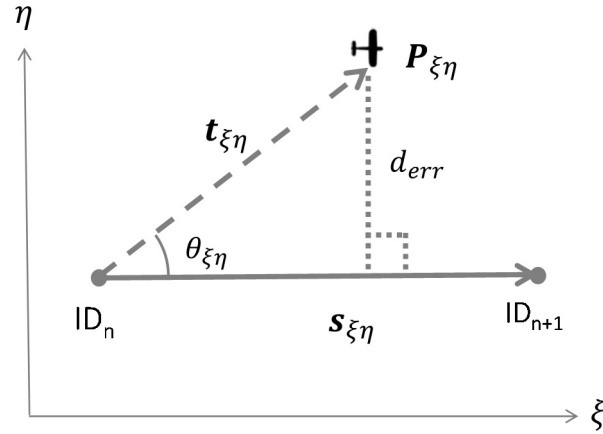


Fig. 3.5.4 Distance Error Estimation Algorithm

$$\mathbf{r}_{ref} = \begin{bmatrix} V_{ref} \\ \alpha_{ref} \\ \phi_{ref} \end{bmatrix} = \begin{bmatrix} V_{const} \\ K_{hp}h_{err} + K_{hd}\dot{h}_{err} \\ K_{dp}d_{err} + K_{dd}\dot{d}_{err} \end{bmatrix} \quad (3.38)$$

$$d_{err} = |\mathbf{t}_{\xi\eta}| \sin \theta_{\xi\eta} = \frac{|\mathbf{s}_{\xi\eta} \times \mathbf{t}_{\xi\eta}|}{|\mathbf{s}_{\xi\eta}|} \quad (3.39)$$

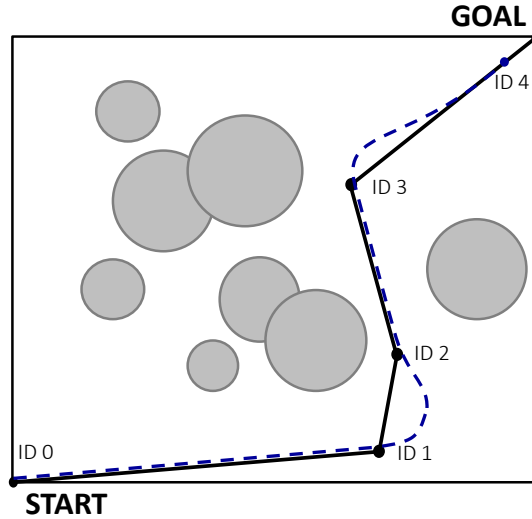
$$\mathbf{t}_{\xi\eta} = \begin{bmatrix} \xi_P - \xi_{ID_n} \\ \eta_P - \eta_{ID_n} \end{bmatrix}, \quad \mathbf{s}_{\xi\eta} = \begin{bmatrix} \xi_{ID_{n+1}} - \xi_{ID_n} \\ \eta_{ID_{n+1}} - \eta_{ID_n} \end{bmatrix} \quad (3.40)$$

$$h_{err} = |\mathbf{t}_{\xi h}| \sin \theta_{\xi h} = \frac{|\mathbf{s}_{\xi h} \times \mathbf{t}_{\xi h}|}{|\mathbf{s}_{\xi h}|} \quad (3.41)$$

$$\begin{bmatrix} V_{err} \\ \alpha_{err} \\ \phi_{err} \end{bmatrix} = \begin{bmatrix} V_{ref} - V \\ \alpha_{ref} - \alpha \\ \phi_{ref} - \phi \end{bmatrix} \quad (3.42)$$

$$\mathbf{v} = \begin{bmatrix} K_{vp}V_{err} + K_{vd}\dot{V}_{err} \\ K_{ap}\alpha_{err} \\ K_{pp}\phi_{err} \end{bmatrix} \quad (3.43)$$

Fig. 3.5.3 に示したように，ゴールノードの ID まで繰り返してフィルタリングすることで，目標経路に追従する飛行軌道を生成する．計算終了の条件は，軌道距離 (*Flight Dist.*) が目標経路の最後のノードの長さ σ_4 と等しくなった時点とする (Fig. 3.5.5)．



計算終了条件
if *Flight Dist.* $\geq \sigma_4$
 Break;
end

Fig. 3.5.5 Terminal Condition of Dynamics Filter

3.6 基礎的な軌道プランナの軌道生成シミュレーション

本節では、基礎的な軌道プランナのシミュレーション結果と考察および課題について述べる。また、アルゴリズムの評価を容易にするため、高度方向の変化を考慮しない $\xi-\eta$ 平面の2次元軌道生成のシミュレーション結果を示す。なお、シミュレーションに用いる機体モデルは3.3節に示した火星探査航空機とする。

3.6.1 シミュレーション条件

はじめに、軌道生成を行う探索域や障害物配置などの環境設定をTable 3.6.1に示す。

Table 3.6.1 Searching Environment

大気密度	[kg/m ³]	0.0118
重力加速度	[m/s ²]	3.2
スタート位置	[m]	[0, 0]
ゴール位置	[m]	[10000, 10000]
障害物数	[個]	0 ~ 50 (10 個刻みで増加させる)
障害物半径	[m]	200 ~ 1000 でランダム
障害物位置	[m]	探索域内でランダム
最大エッジ長	[m]	500
衝突回避マージン	[m]	12.1 (片翼長の 10 倍)

ここで、3.1 節で示した制御器において、本シミュレーションで使用する目標速度は $V_{ref} = 60$ [m/s]とし、各制御ゲインはTable 3.6.2を用いた。

Table 3.6.2 Control Gain of Dynamics Filter

K_{hp}	[1/m]	0.0	K_{hd}	[s/m]	0.0
K_{dp}	[1/m]	0.3	K_{dd}	[s/m]	4.0
K_{vp}	[1/s ²]	1.0	K_{vd}	[1/s]	10.0
K_{ap}	[1/s ²]	0.0	K_{pp}	[1/s ²]	0.5

また、力学フィルタには実用的な軌道を生成させるため、コマンド生成時に制限を設けている．設定した制限値を以下に示す．

Table 3.6.3 Command Limiters

$0 \leq T_{com} \leq 10$	[N]	$-10 \leq \dot{T}_{com} \leq 10$	[N/s]
$-10 \leq \alpha_{com} \leq 10$	[°]	$-10 \leq \dot{\alpha}_{com} \leq 10$	[°/s]
$-30 \leq \phi_{com} \leq 30$	[°]	$-30 \leq \dot{\phi}_{com} \leq 30$	[°/s]

力学フィルタの数値計算に与える初期条件は以下の通りである．ここで、速度、推力、迎角の初期値は、大山らが研究した火星探査航空機の 100km 滑空を目指した多目的最適設計解^[25]の結果に基づく．

Table 3.6.4 Initial Parameter

位置 $[\xi, \eta]$	[m]	[0, 0]
時間 t	[s]	0
速度 V	[m/s]	60
方位角 ψ	[°]	ツリーの初期方位と等しい
推力	[N]	1.37
迎角	[°]	6.79
バンク角	[°]	0

また、本シミュレーションは下記の計算環境で行った．

Table 3.6.5 Computational Environment

CPU	Intel Core i7-4790 (3.60 GHz)
RAM	8.00 GB
Software	MATLAB R2014a

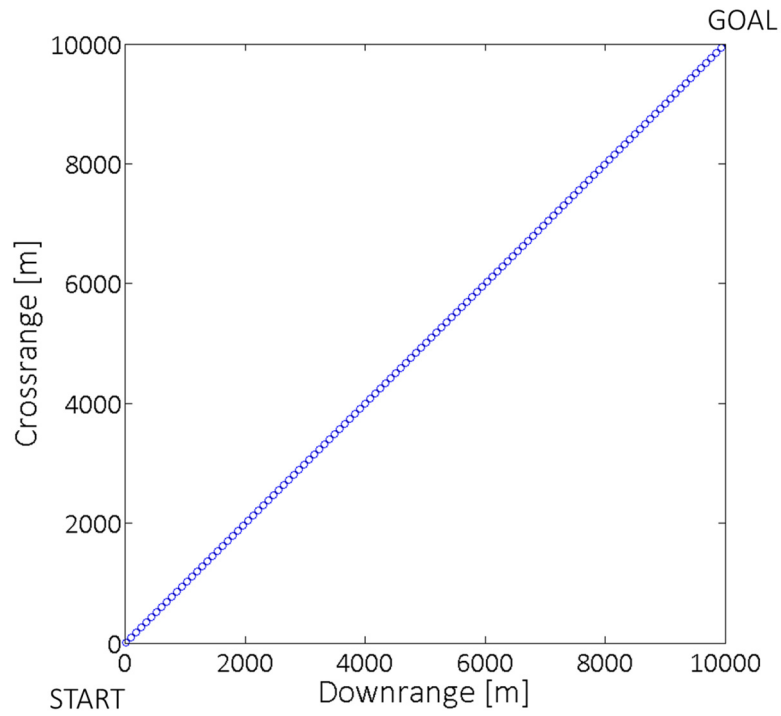
3.6.2 シミュレーション結果

3.6.2.1 成功例

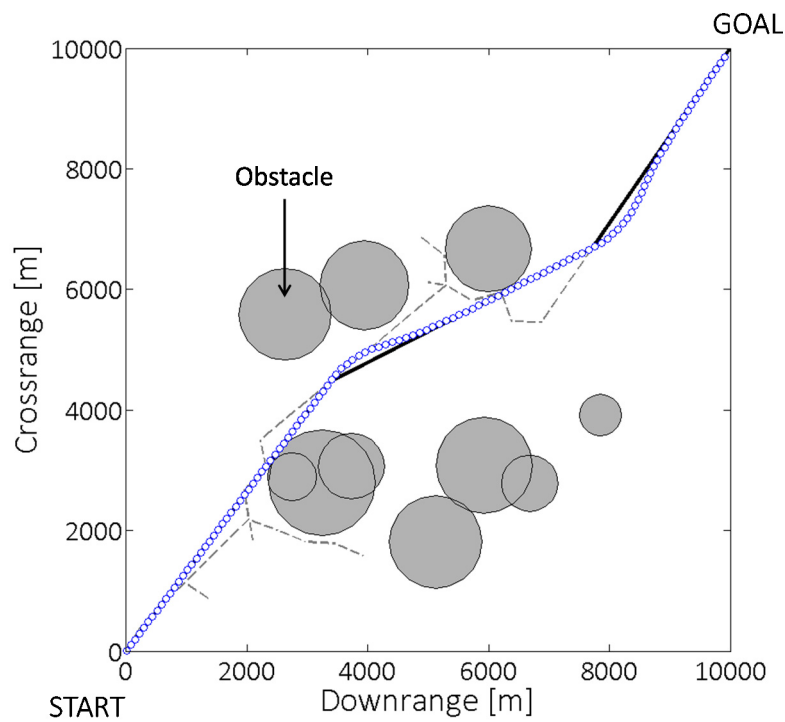
上述の条件に基づき，障害物の数を 0 個から 50 個まで 10 個刻みで増加させた場合の軌道生成シミュレーションを行った (Fig. 3.6.1). また，Fig. 3.6.2, Fig. 3.6.3 には，Fig. 3.6.1 (f) の軌道生成時に得られたコマンド時系列，速度，およびランダム探索で得られた経路と力学フィルタで得られた軌道との距離誤差を示す．これらの図において，灰色丸は障害物，破線は RRT で生成された全てのツリー，実線はゴールに到達する RRT の最終的な経路，青丸プロットは力学フィルタで得られた軌道を示す．なお，ここでは障害物に衝突することなくゴールまで到達する軌道を描けた“成功例”のみを載せており，失敗例は，本アルゴリズムで明らかとなる課題と併せて 3.6.2.3 条で考察する．

シミュレーションの結果より，アルゴリズムのコンセプト通りランダム探索により障害物回避経路を探索し，その後力学フィルタにより飛行可能な軌道に変換することに成功した．また，障害物の数が増加してもゴールまでの障害物回避軌道を描けることが分かった．さらに，力学フィルタにより得られた推力，迎角，バンク角はいずれも実現可能な滑らかなコマンド時系列となった．速度については，常に定常速度付近を維持しており，ランダム探索で得られる目標経路と力学フィルタで得られる軌道の距離誤差は，時間の経過とともにゼロに収束していることが分かる．つまり，軌道が目標経路に収束していることを意味する．なお，コマンドの時系列データにおいて，推力と迎角の変化に類似した傾向が見られるが，これは迎え角が大きくなると抗力が増して速度が低下するが，推力を増すことで速度一定を保つよう制御しているためと考えられる．

以上の結果より，力学フィルタに実際のモデルとの相違がないと仮定すると，得られた推力，迎角，バンク角を指令値として機体を制御することで，このような複雑な障害物環境下でも衝突を回避して目的地まで誘導することが可能であることが示された．



(a) Obstacle Number 0



(b) Obstacle Number 10

Fig. 3.6.1 Simulation Result of Basic Trajectory Planner
(to be continued)

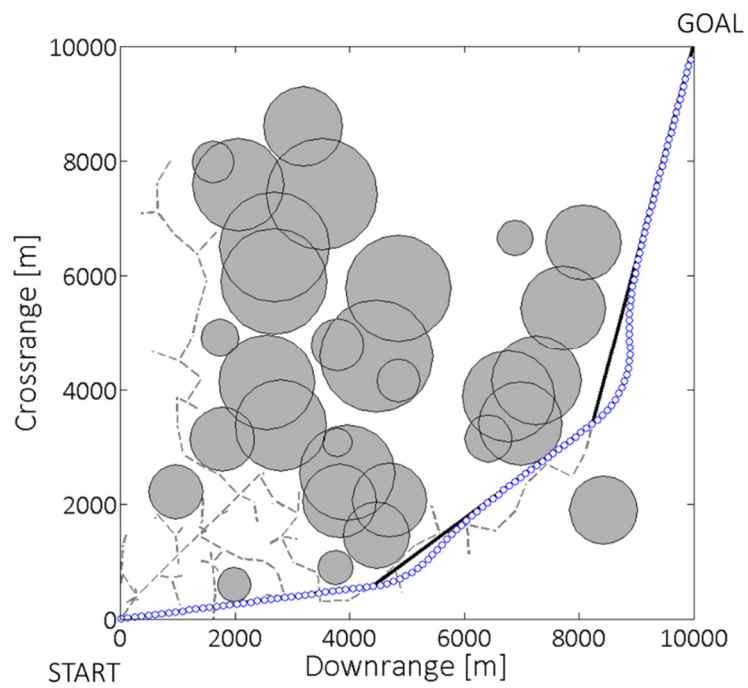
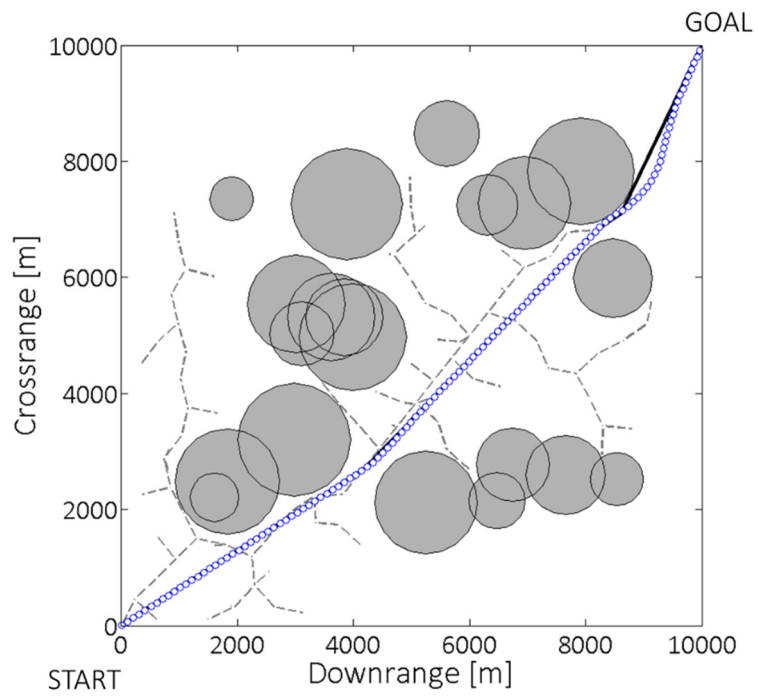
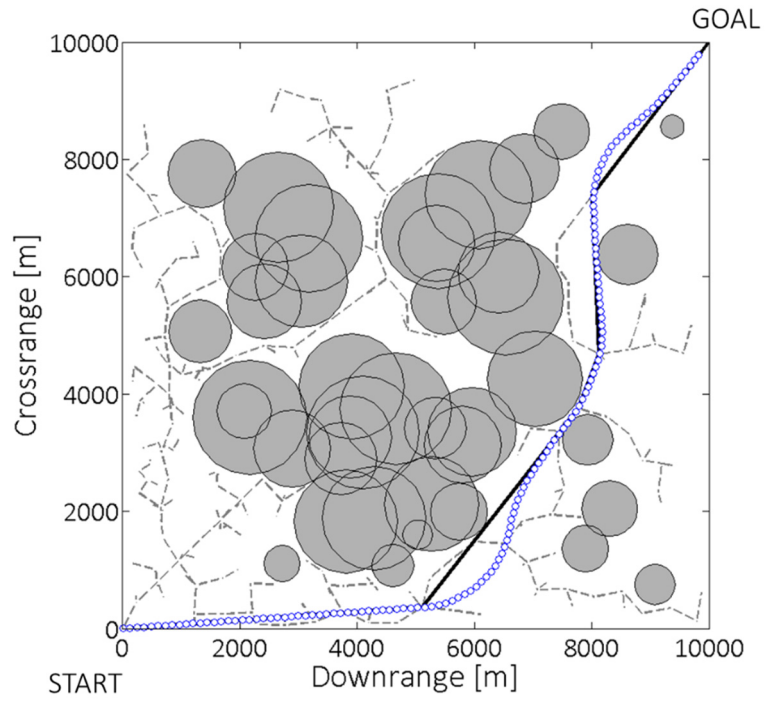
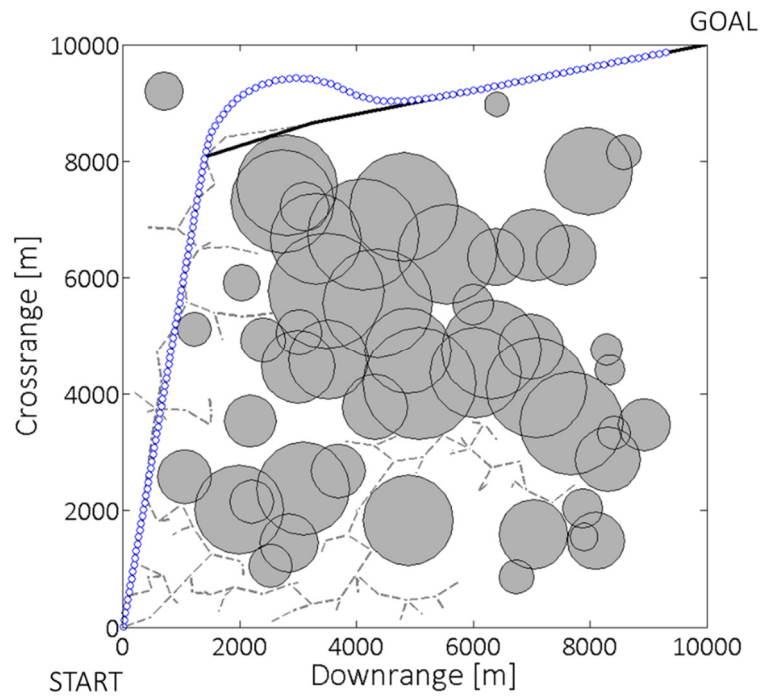


Fig. 3.6.1 Simulation Result of Basic Trajectory Planner
(to be continued)



(e) Obstacle Number 40



(f) Obstacle Number 50

Fig. 3.6.1 Simulation Result of Basic Trajectory Planner
(to be concluded)

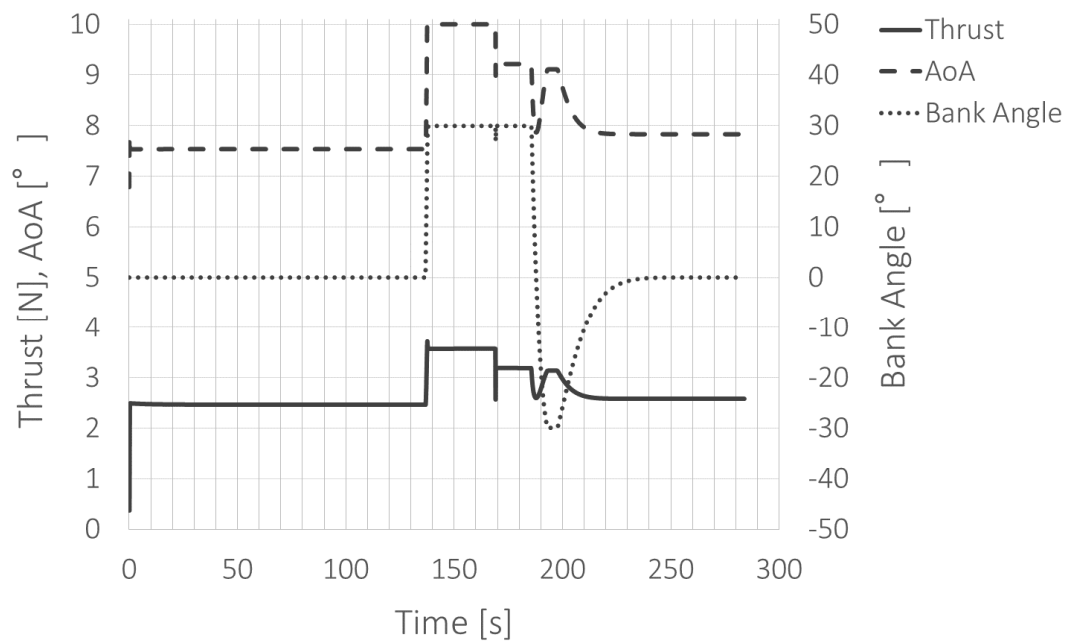


Fig. 3.6.2 Guidance Commands

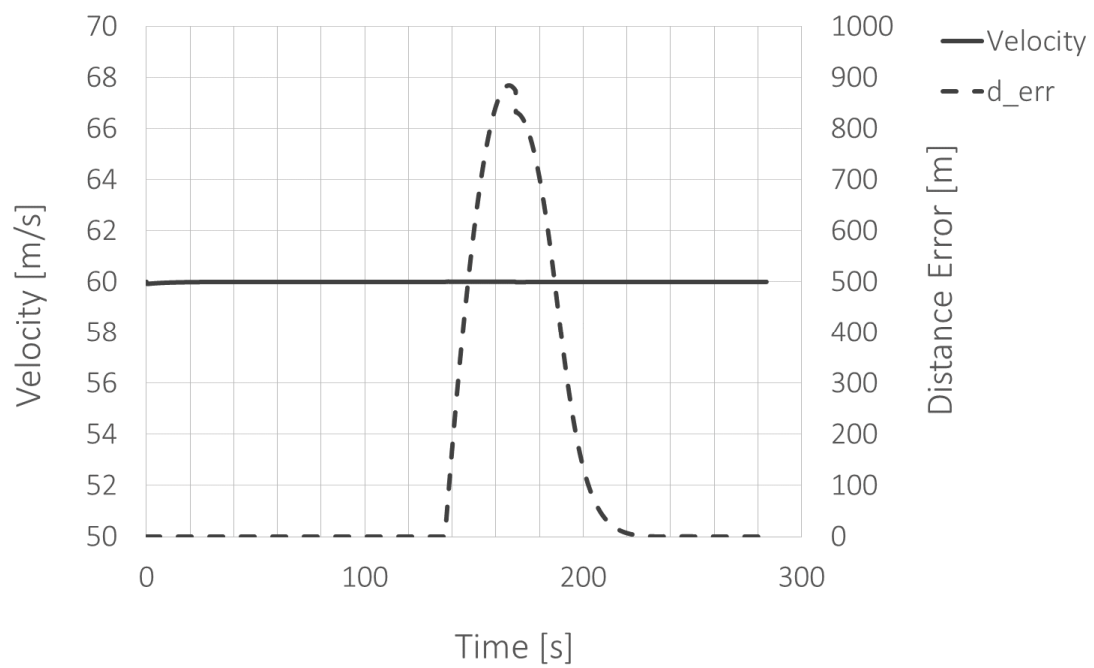


Fig. 3.6.3 Velocity and Distance Error

3.6.2.2 計算時間

これらの障害物環境における軌道プランナの計算時間を計測した。計測用 PC は、Table 3.6.5 と同様のもので、ランダム探索と力学フィルタのフェーズに分けてそれぞれ計測した。なお、本計測結果は機体の実装する CPU における計測ではないためリアルタイム性の議論には適さず、各フェーズが計算時間に与える影響度合いを評価する目的として実施した。

各障害物数に対して 10 回軌道プランナを実行し、その平均値を算出した。結果を Fig. 3.6.4 に示す。なお、描画などアルゴリズム以外にかかる時間は省いている。結果より、障害物の数が増加すると全体の計算時間が増加していることが分かる。特に、ランダム探索の時間増加が大きく影響しており、力学フィルタの時間増加は殆ど見られない。これは以下のように考察できる。ランダム探索では、障害物数の増加にともないツリーがゴールに到達しにくくなり反復回数が増加する。また、衝突判定の回数も増えることによる影響を受けて計算時間が増加している。一方で力学フィルタの計算時間は、飛行距離にのみ依存するため殆ど増えていない。しかし、障害物の増加によって経路が複雑になり飛行距離が伸びるため、その分僅かに増加傾向が見られる。

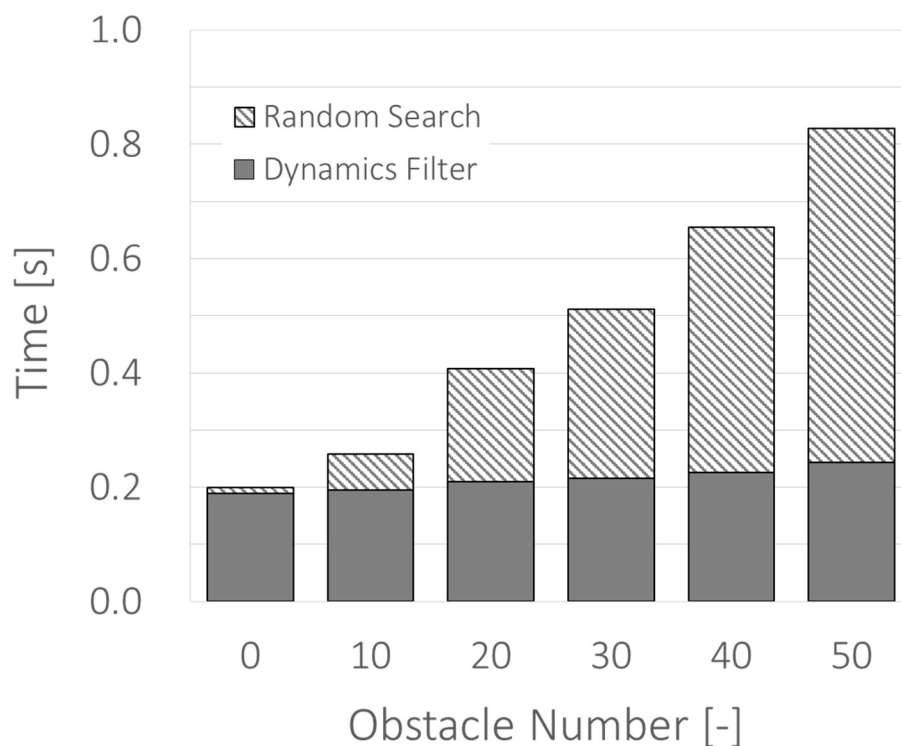


Fig. 3.6.4 Computer Calculation Time

3.6.2.3 失敗例と課題

ここまでは、成功例を複数取り上げたが、一方で多くの課題も見つかった。本節では障害物回避軌道の生成に失敗するケースの考察や、現状の基礎的な軌道プランナの課題について述べる。はじめに、シミュレーションで見つかった主な課題は以下の 5 項目である。

- I. 力学フィルタ後の軌道が障害物に衝突する。
- II. 力学フィルタの終了条件の不良により目的地まで軌道が描けない。
- III. 目標 ID の更新方法の不良により誤誘導が生じる。
- IV. 機体の初期方位角に対して目標経路の初期方位角に不整合がある。
- V. ゴール地点の軌道進入方位が不定である。

(1) 力学フィルタ後の軌道が障害物に衝突する。

本軌道プランナでは、ランダム探索の後に力学フィルタを行う。ランダム探索の際には衝突判定を行いツリーを成長させるため、衝突回避性を保障しているが、力学フィルタで得られる軌道においては衝突回避を保障していない。ランダム探索時に十分な衝突回避マージンを設けることにより、一定の衝突回避性を付与することは可能であるが、マージンの値を定量的に決定することは困難である。実際にシミュレーションの結果より、障害物に衝突する軌道が生成されるケースが多数確認された (Fig. 3.6.5)。衝突箇所を赤丸で示す)。よって、力学フィルタ後の軌道についても完全に衝突回避を保障できるアルゴリズムへの改善が必須課題である。

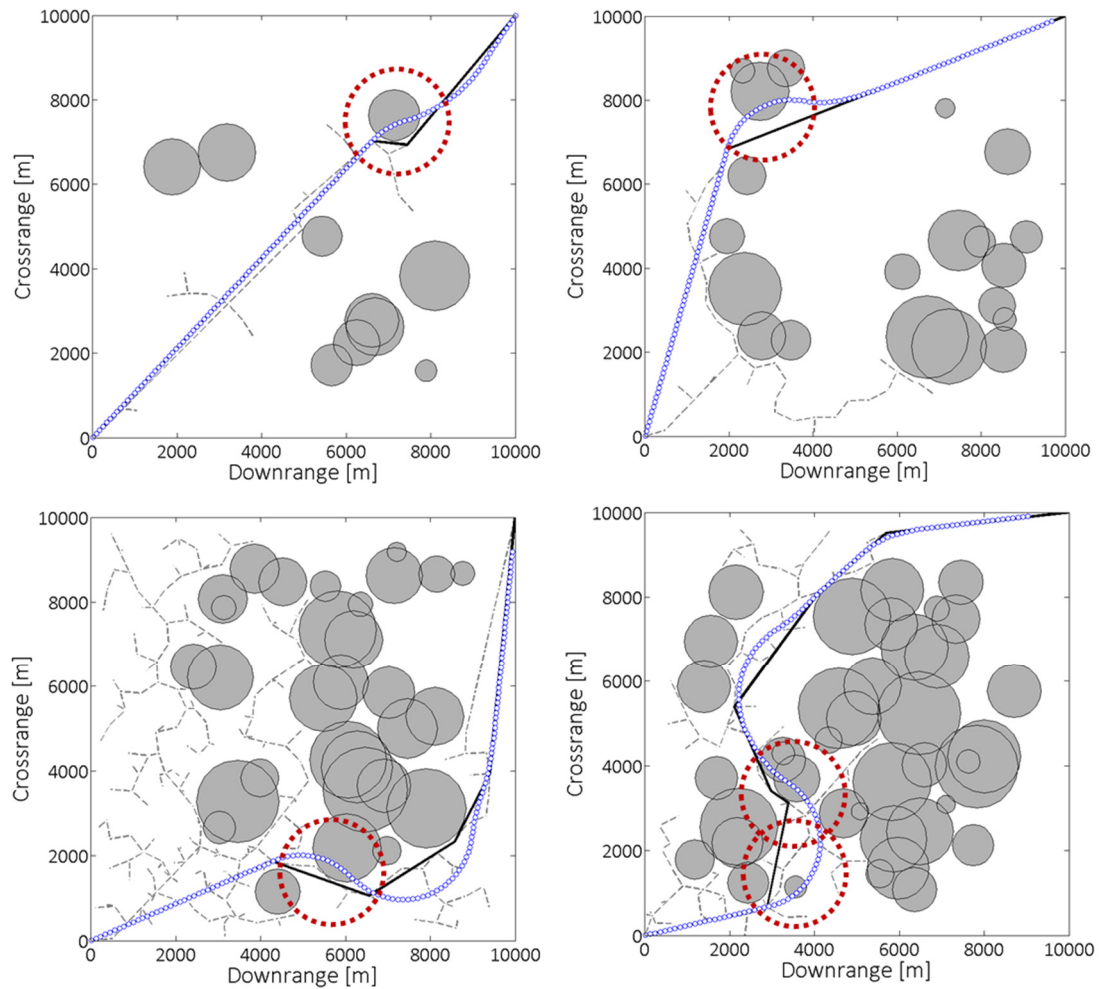


Fig. 3.6.5 Trajectory with Collisions

(2) 力学フィルタの終了条件の不良により目的地まで軌道が描けない。

現在の力学フィルタの終了条件は、ランダム探索で得られた目標経路のスタートからゴールまでのエッジ長さを飛行距離を超えた時点で終了としている。このとき、力学フィルタにより目標経路から大きく逸脱した場合、飛行距離が目標経路の長さより長くなる。すると、ゴールに到達する前に力学フィルタが終了してしまい、ゴールに到達するまでの軌道を描けない。シミュレーションでもその事象が確認できる (Fig. 3.6.6)。そこで、ゴールまで確実に到達するアルゴリズムに改善する必要がある。

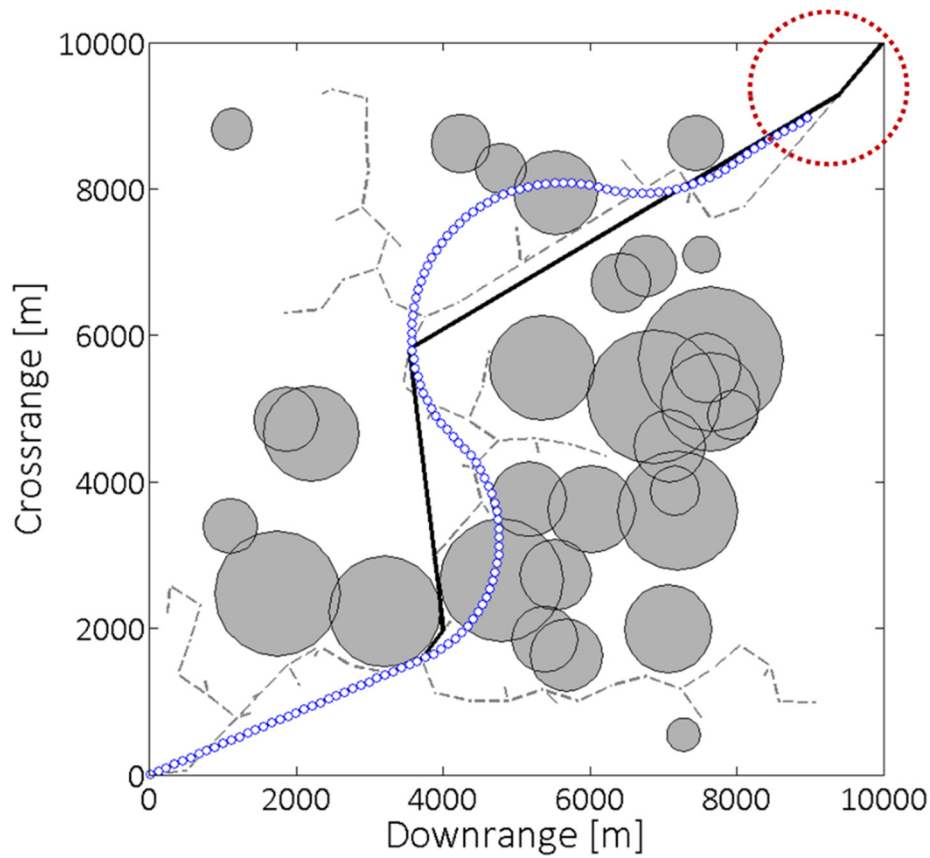


Fig. 3.6.6 Incompletion of Trajectory Generation

(3) 目標 ID の更新方法の不良により誤誘導が生じる．

目標 ID の更新方法は，スタートから目標 ID までの経路長さを軌道の距離が超えた場合に次の ID に更新するよう定めている．また，目標バンク角の符号（左にバンクするか右にバンクするか）は，現在目標としている経路の進行方向に対して機体が左右どちらに位置しているかで切り替えている．このとき，目標経路から大きく逸脱した場合，軌道が追従できないケースが稀にある．この事象について，実際に誤誘導が生じたシミュレーション結果（Fig. 3.6.7）を用いて解説する．図内には，経路長さと軌道距離が一致する時点をそれぞれ黒丸と青丸のプロットで図示している（ σ_1 は黒丸と青丸が重なっている）．いま，P1 の地点まで力学フィルタにより軌道が作られたとする．このとき，目標 ID は 4 であり，目標経路の ID3 と ID4 の成す進行方向（赤矢印）に対して機体は右側に存在する，そこで軌道を修正するため左にバンクを行う．その後しばらく左バンクで軌道を描いていくが，軌道距離が σ_4 に達したとき，目標 ID は 5（ゴール地点）

に更新される．ここで、力学フィルタが P2 の地点にいるとき、ID4 と ID5 の成す進行方向に対して機体は右側に位置するため、さらに左にバンクを指令する．そして、そのまま左バンクを続けても軌道が目標経路に沿うことは無く、常に経路右側に位置することになるため、左バンクを続けることとなる．これが誤誘導の原因であり、目標 ID の更新アルゴリズムに関して改善が必要である．

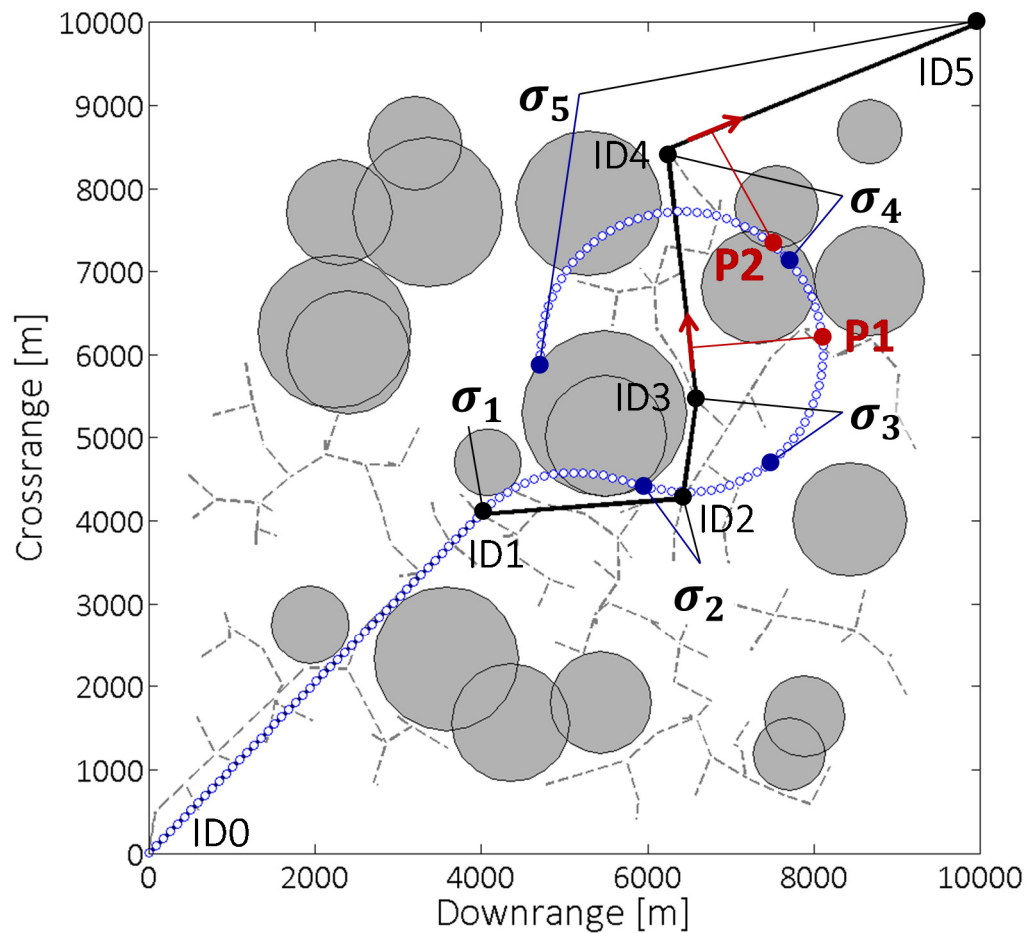


Fig. 3.6.7 Trajectory Misgeneration

(4) 機体の初期方位角に対して目標経路の初期方位角に不整合がある。

これまでのシミュレーションでは、力学フィルタの初期値として用いられる機体の初期方位角は経路の初期方位角と等しいとして行った。しかし、実際にリアルタイム誘導を行う場合には、ランダム探索から得られる経路の方位角と機体が現在向いている方位角が異なる。これは、現在のアルゴリズムにおいて、ランダム探索は機体の初期方位角を考慮せず、ランダムにツリーを成長させているためである。その事象を表したシミュレーション結果を Fig. 3.6.8 に示す。いま、機体の初期位置を $[\xi, \eta] = [3000, 3000]$ として、初期方位角が -45° (赤矢印) であるとする。この位置からランダム探索を開始すると、図のようにほぼ正対する方向 (方位角約 135°) に目標経路を導く場合がある。このとき、機体の初期方位角と目標経路の方位角の乖離が大きく、力学フィルタの軌道は目標経路から大きく逸脱して衝突する軌道を描くことになる。このように、初期状態量を考慮せずにランダム探索を実行すると力学フィルタの描く軌道との乖離が大きくなり、衝突軌道を描く大きな原因となるため、改善が必要である。

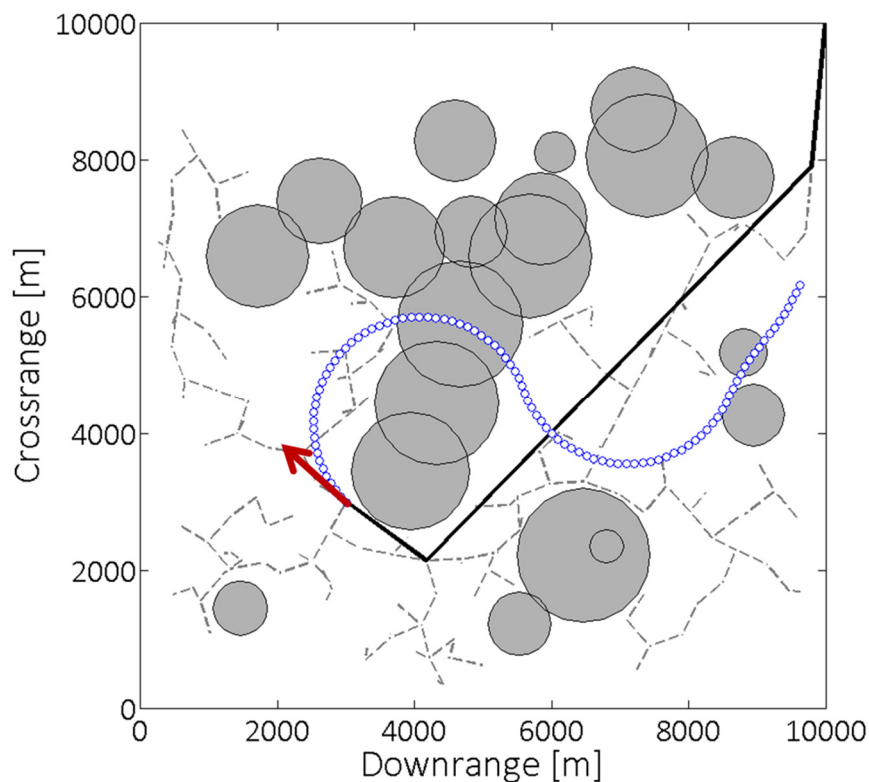


Fig. 3.6.8 Initial Angle Mismatch

(5) ゴール地点の軌道進入方位が不定である。

UAV の自律航行では、ミッションによっては目標地点へ到達するだけでなく、目標地点に到達時の経路角、方位角をコントロールしたい場合が考えられる。例えば、有翼ロケットの宇宙空間到達後における帰還飛行フェーズでは、目標地点（宇宙港など）に対して滑走路に沿うような向きで帰還する軌道生成が望ましい。このような問題を想定するとき、現状のアルゴリズムでは、そのランダム特性のため終端の軌道入射角度を定めることができない。Fig. 3.6.9 において、有翼ロケットは緑色破線のように滑走路に侵入してくることが望ましいが、本軌道プランナでは青色破線のような軌道を描く可能性がある。

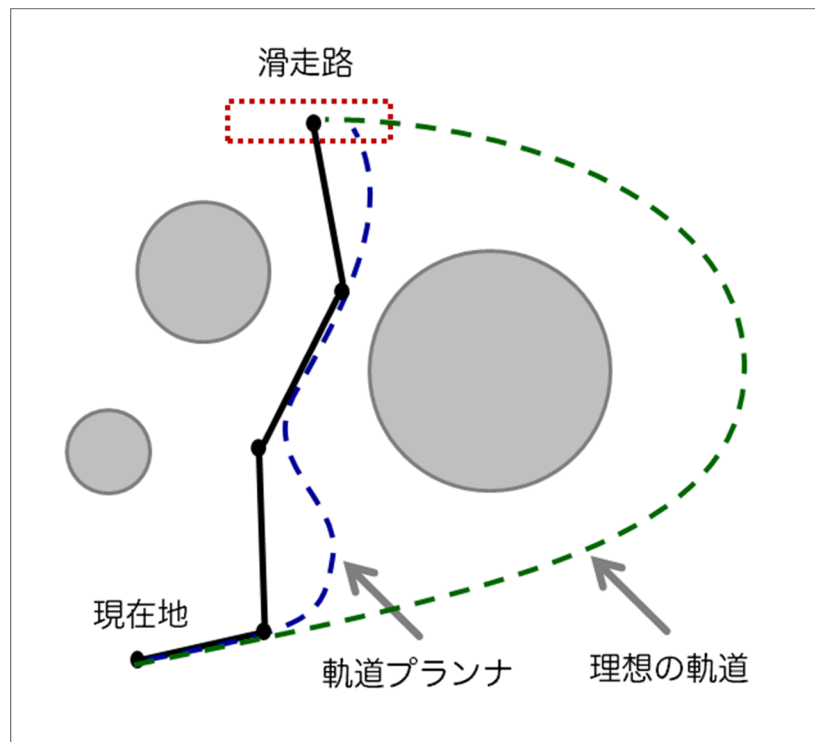


Fig. 3.6.9 Indeterminate Terminal Trajectory Direction

3.7 本章のまとめ

本章では、はじめに力学フィルタの構成を行い、ランダム探索と力学フィルタを組み合わせた基礎的な軌道プランナを構築した。また、軌道プランナによる軌道生成シミュレーションを行い、ランダム探索で得られた経路に基づいた力学フィルタにより滑らかな軌道が描けることが分かり、障害物を増加させても同様に軌道生成が行えた。一方で、障害物に衝突する軌道が生成されたり、誤誘導が生じるなど多くの課題が明らかとなった。次章では、これらの課題を克服する新しい軌道プランナを検討する。

第 4 章 ランダム探索と力学フィルタによる融合型軌道プランナの提案

4.1 融合型軌道プランナ

4.1.1 融合型軌道プランナの概要

前章では、ランダム探索と力学フィルタを組み合わせた軌道プランナによって障害物回避軌道が生成できる可能性が示された。しかし一方で、同手法には多くの課題が見つかった。ここで、本章の方針を示すために前章で見つかった課題を改めて以下に示す。

- I. 力学フィルタ後の軌道が障害物に衝突する。
- II. 力学フィルタの終了条件の不良により目的地まで軌道が描けない。
- III. 目標 ID の更新方法の不良により誤誘導が生じる。
- IV. 機体の初期方位角に対して目標経路の初期方位角に不整合がある。
- V. ゴール地点の軌道進入方位が不定である。

この中で最も重要な課題は I であり、力学フィルタ後の軌道に衝突回避性を保障することは前章の軌道プランナでは困難である。生成された軌道に対して衝突判定を行うことは可能であるが、そこで衝突と判断された場合、衝突しない軌道が描けるまで何度も軌道生成を試みるなどの策は計算時間が掛かり、リアルタイム性の観点から好ましくない。

そこで、I の課題を克服するため、障害物回避経路を探索する問題と軌道を生成する問題を同時に解くという着想に至った。本章では、ランダム探索に力学フィルタのアルゴリズムを組み込むことで、障害物回避軌道を生成しながらランダムに探索することを可能にする融合型軌道プランナを提案する。融合型軌道プランナ概念図を Fig. 4.1.1 に示す。また、本手法では、課題 I だけでなく、上記全ての課題の解決に取り組む。

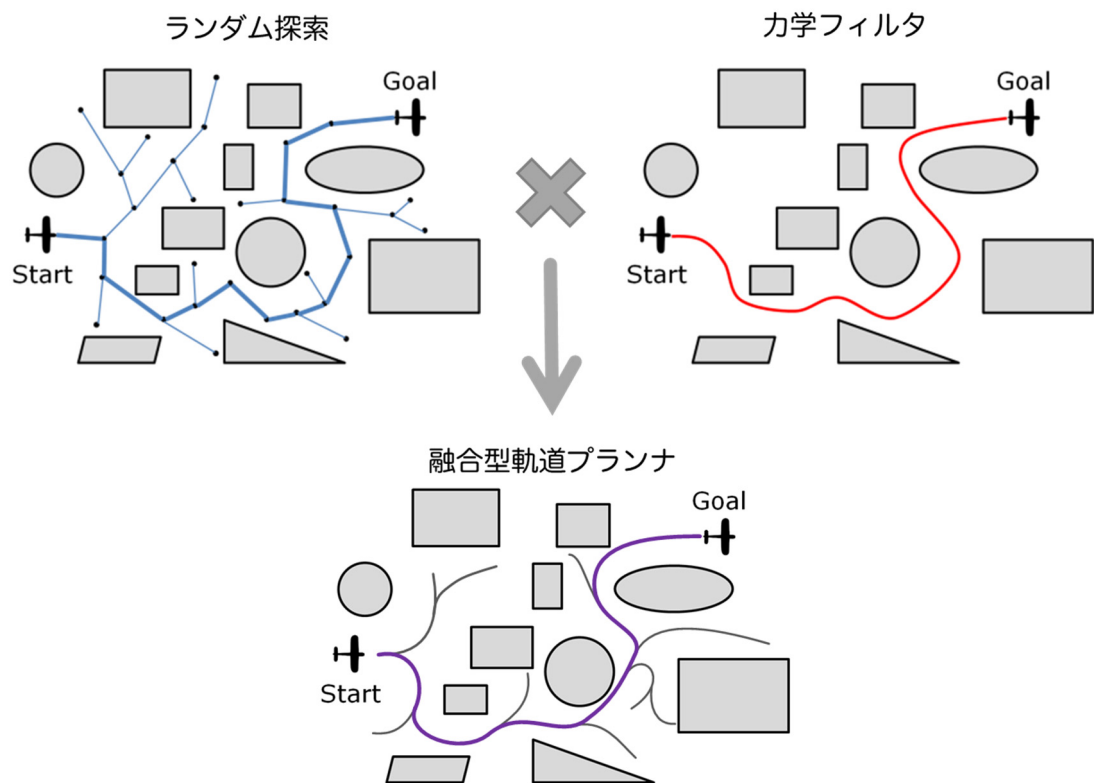


Fig. 4.1.1 Conceptual Diagram of Fused Trajectory Planner

4.1.2 アルゴリズムの構成

4.1.2.1 概念図

融合型軌道プランナのアルゴリズム概念図を Fig.4.1.2 に示し，以下は概念図の説明を記す．

(1) Initialization

いま，Fig. 4.1.2 (a)に示すような障害物環境があると仮定する．スタートとゴールの矢印はそれぞれの地点における軌道角度を表現している．つまり，スタート地点の軌道角度は現在の状態量で決まる初期軌道角度であり，ゴール地点は誘導アルゴリズムに要求される目標の終端軌道角度である．

(2) Step 1.

本アルゴリズムでは、4.1.1 節で述べた課題 V（ゴール地点の軌道進入方位が不定である）の解決策として、近似曲線を用いた擬似軌道生成を行う（Fig. 4.1.2 (b)）。これは、初期角度と終端角度を制御可能な近似曲線を描いて、力学フィルタに対してゴール位置とその進入方位を満たす指令を与えることを目的とする。ここで用いる近似曲線は、任意の 4 つの制御点によって初期角度と終端角度が制御可能となる 3 次ベジエ曲線（Quadratic Bezier Curve）とする。ベジエ曲線の定義や適用方法などの詳細は 4.1.3 節で述べる。そして今、与えられたベジエ曲線（破線）に対して衝突判定を行う。Fig. 4.1.2 (b)では障害物に衝突しており、この場合 Step2 に移行する。ここで衝突が無いときは、ベジエ曲線から得られる指令角を用いて Step7 に移行する。

(3) Step 2.

Step1 で生成されたベジエ曲線の情報は、障害物に衝突していることから破棄され、探索域内にランダム点 P_{rand} が 1 点打たれる（Fig. 4.1.2 (c)）。ランダム点は、枝を成長させる位置（ノード）をツリー内から無作為に選び出すために用いられる。

(4) Step 3.

ランダム点からノードの最近傍点(Nearest Node)を選ぶ。初期段階ではスタートノードが選ばれる（Fig. 4.1.2 (d)）。この位置からツリーの新しい枝が成長する。

(5) Step 4.

Step3 で選ばれた最近傍のノードに対して、ランダムに方位角指令 ψ_{rand} 、経路角指令 γ_{rand} が与えられる（Fig. 4.1.2 (e)）。ここで、それぞれの指令値は軌道プランナの定める制限内の値を取る。これは経路角 100° といった非現実的な指令値を与えないようにするためである。

(6) Step 5.

Step4 で与えられた角度指令値に収束するように、予め決められた T 秒間（枝成長時間と呼ぶ）だけ力学フィルタの処理を行い枝を成長させていく（Fig. 4.1.2 (f)）。力学フィルタ処理の過程では、計算刻み δt 秒毎にノードが生成され、それら全てのノードに対して ID が割り振られる。また各ノードはそれぞれ自身の ID、力学フィルタで計算されるパラメータ、および接続元（一つ手前）ID の情報が格納される。このとき、Step3 で選ばれたノードが持つ各情報を力学フィルタの初期値として用いることで、成長元のツ

リーと滑らかに接合する軌道を描くことができる。また、 δt 秒毎に障害物との衝突判定を行い、衝突した場合には即座にフィルタ処理を終了する。

本手法で用いる力学フィルタの計算過程は 3.4.4 項で説明したものと同一であるが、適用する制御器が 3.5.2 項のものと異なる。3.5.2 項では、ランダム探索で得られた経路情報に基づいて目標値 \mathbf{r}_{ref} を生成し、制御器を構成した。しかし、融合型軌道プランナでは、目標値 \mathbf{r}_{ref} そのものをランダムに与えることで制御器を構成する。融合型軌道プランナで用いる目標値 \mathbf{r}_{ref} を Eq. (4.1) に示す。ここで、目標速度 V_{ref} は、一定（巡航速度） V_{const} を与える。また、目標経路角 γ_{rand} および目標方位角 ψ_{rand} は、ランダム値により無作為に γ_{rand} 、 ψ_{rand} を与える。次に、Eq. (4.2)により現在の速度 V 、経路角 γ 、方位角 ψ との誤差を算出する。そして、Eq. (4.3)により誤差とその時間変化率を用いて PD 制御器を構成し、入力 \mathbf{v} を生成する。以上のように、ランダムに目標値を与えてフィルタ処理を行う。

$$\mathbf{r}_{ref} = \begin{bmatrix} V_{ref} \\ \gamma_{ref} \\ \psi_{ref} \end{bmatrix} = \begin{bmatrix} V_{const} \\ \gamma_{rand} \\ \psi_{rand} \end{bmatrix} \quad (4.1)$$

$$\begin{bmatrix} V_{err} \\ \gamma_{err} \\ \psi_{err} \end{bmatrix} = \begin{bmatrix} V_{ref} - V \\ \gamma_{ref} - \gamma \\ \psi_{ref} - \psi \end{bmatrix} \quad (4.2)$$

$$\mathbf{v} = \begin{bmatrix} K_{vp}V_{err} + K_{vd}\dot{V}_{err} \\ K_{gp}\gamma_{err} + K_{gd}\dot{\gamma}_{err} \\ K_{pp}\psi_{err} + K_{pd}\dot{\psi}_{err} \end{bmatrix} \quad (4.3)$$

(7) Step 6.

Step5 でツリーの成長が終わると、その地点からゴールまでを結ぶベジエ曲線を描いて衝突判定を行う (Fig. 4.1.2 (g))。衝突する場合は Step2 へ移行し、衝突しない場合は Step7 へ移行する。

(8) Step 2.~Step 6.

Step2~6 で述べた処理を反復し、ランダムに探索を進めていく。反復は、ベジエ曲線が衝突することなくゴールと結合されるまで繰り返される (Fig. 4.1.2 (h))。

(9) Step 7.

障害物に衝突しないベジエ曲線が描けたとき、軌道プランナはベジエ曲線を用いてゴールに向かって力学フィルタの処理を行う。ここで、力学フィルタに与えられる指令値は、ベジエ曲線を分割して得られる座標の差分から導いた指令角である。ただし、ベジエ曲線は簡易的な疑似軌道であり、運動モデルとの誤差が大きいため、力学フィルタをかけると軌道が徐々にベジエ曲線から外れていく。そこで、ベジエ曲線を一定周期（枝成長時間 T 秒）毎に逐次更新することにより誤差を修正しながらゴールへアプローチしていく（Fig. 4.1.2 (i)）。

(10) Step 8.

ゴールへの到達判定は、位置誤差と軌道角度誤差で判定される。ミッションで定める閾値以内に収まるとき、Step7 が終了される。ゴールまで到達した場合、ゴール方向から接続元の ID を逆順することにより、ゴールとスタートを結ぶ最終的な障害物回避軌道が得られる（Fig. 4.1.2 (j)）。

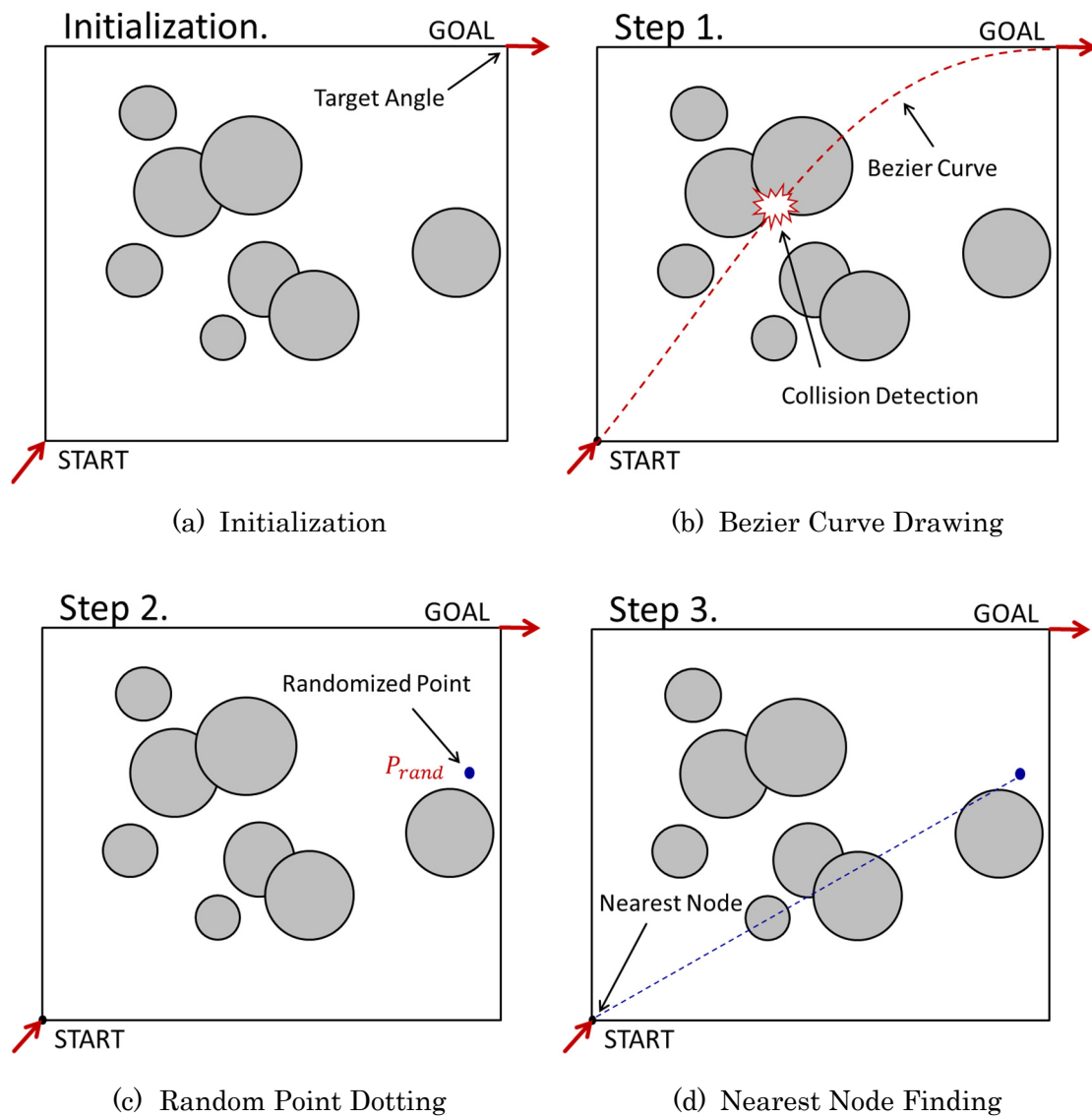


Fig. 4.1.2 Algorithm Flow of Coordinated Trajectory Planner
(to be continued)

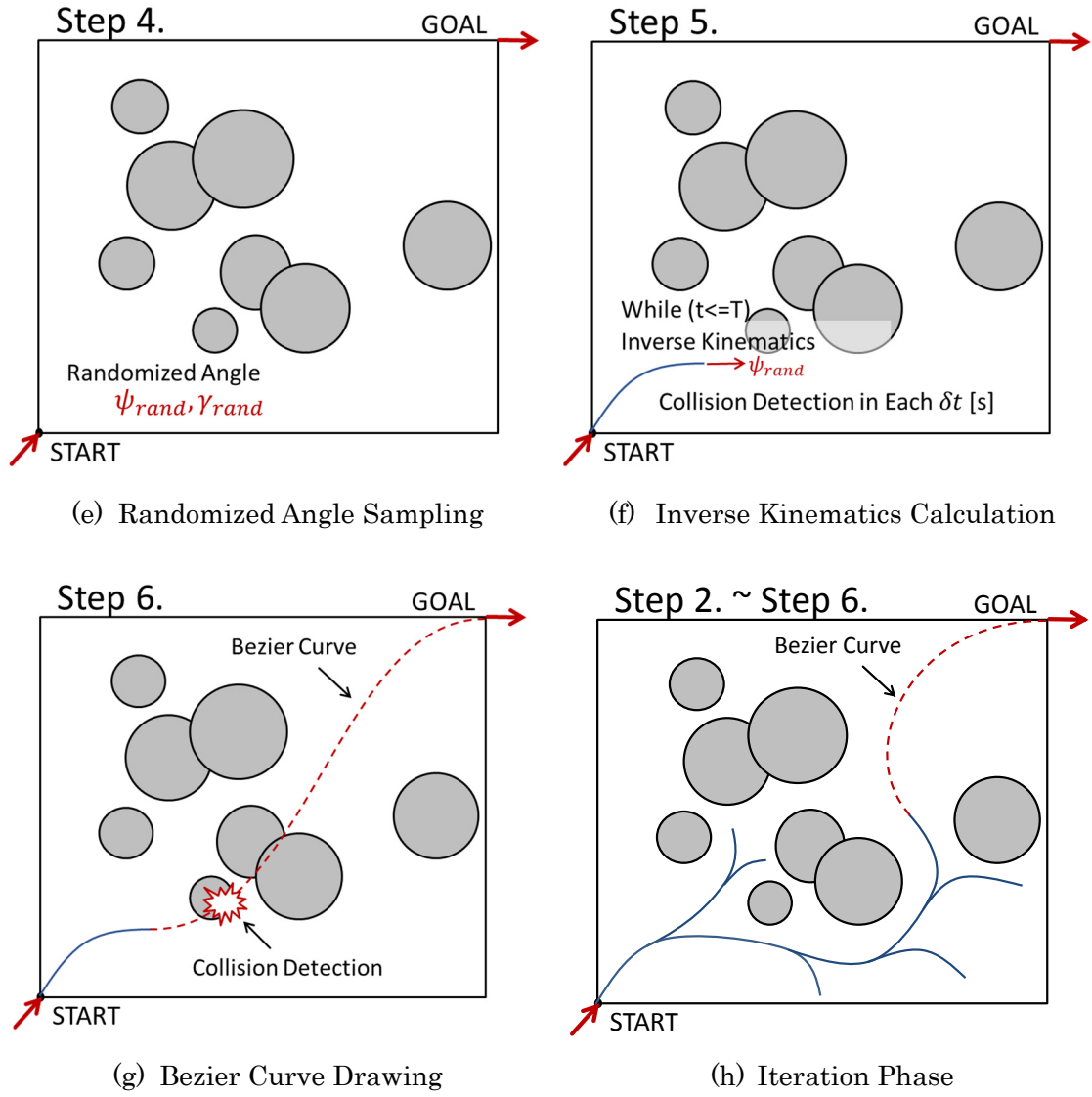


Fig. 4.1.2 Algorithm Flow of Coordinated Trajectory Planner
(to be continued)

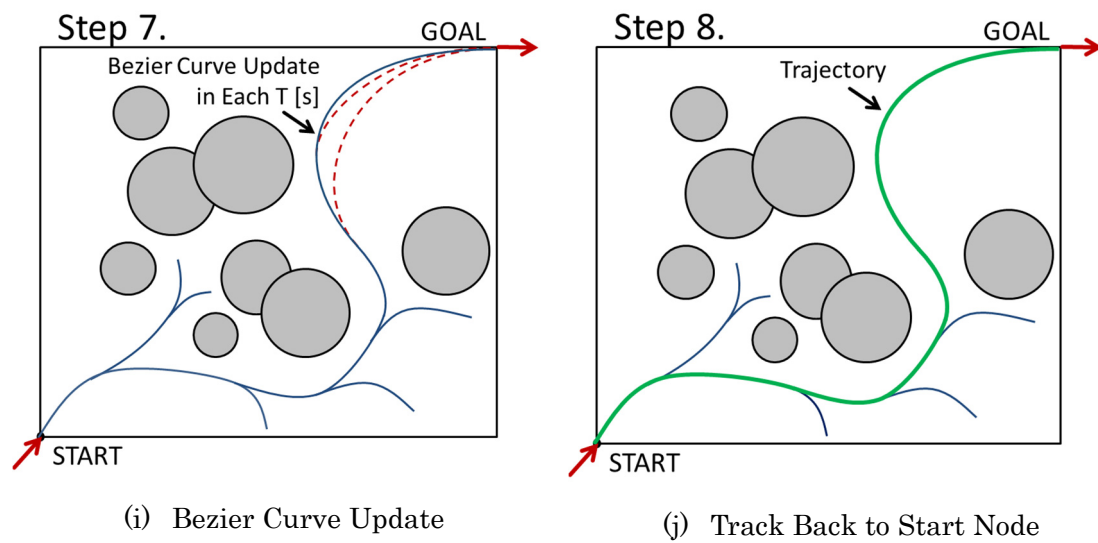


Fig. 4.1.2 Algorithm Flow of Coordinated Trajectory Planner
(to be concluded)

4.1.2.2 フローチャート

本アルゴリズムのフローチャートを Fig. 4.1.3 に示す.

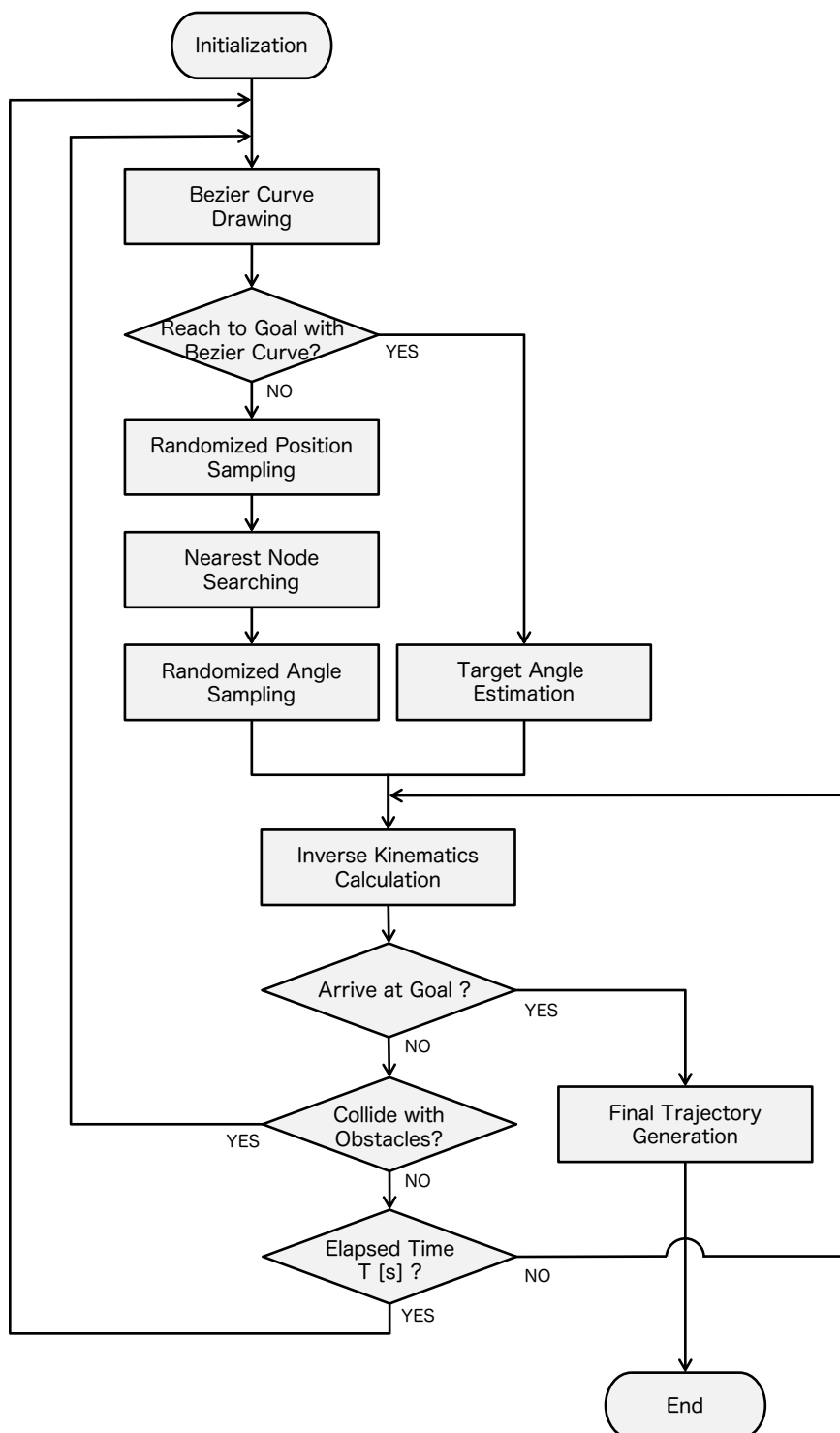


Fig. 4.1.3 Algorithm Flow Chart

4.1.3 ベジエ曲線の適用

4.1.3.1 ベジエ曲線概念と定義

ベジエ曲線^[60]は Pierre Bézier によって考案された N 個の制御点を用いて $N-1$ 次の曲線を描く手法であり, コンピュータ上で滑らかな曲線を描くグラフィック技術などで一般的に利用される. 本軌道プランナに用いる 3 次ベジエ曲線概念図を Fig. 4.1.4 に, また定義式を Eq. (4.4), Eq. (4.5)に記す. 図に示すように, スタートの前方とゴールの手前に, それぞれの軌道角度を満たす位置に制御点を配置してベジエ曲線を描く.

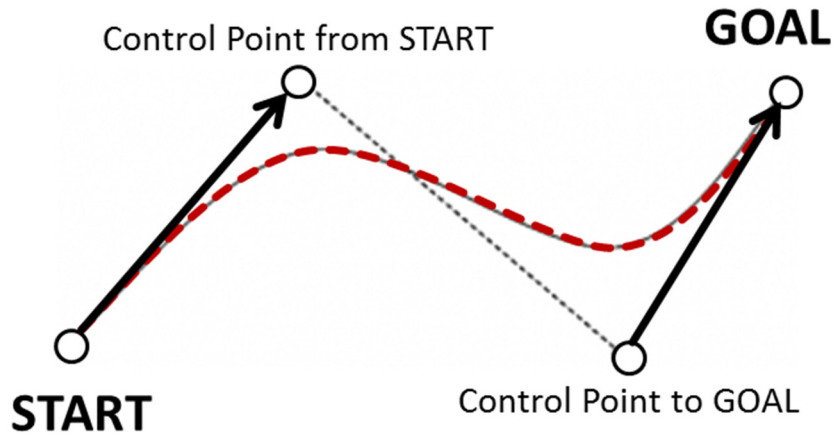


Fig. 4.1.4 Conceptual Diagram of Trajectory Planner with Bezier Curve

$$P(t) = \sum_{i=0}^{N-1} B_i J_{(N-1)i}(t) \quad (4.4)$$

$$J_{ni}(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad (4.5)$$

上式において, B_0, B_1, \dots, B_{N-1} は制御点を表し, $J_{ni}(t)$, ($0 < t < 1$)はバーンスタイン基底関数 (Bernstein Basis Polynomials) ^[61]であり $\binom{n}{i}$ は二項係数を表す. この定義に基づくと, 3 次ベジエ曲線は Eq. (4.6)のように表せる.

$$P(t) = (1-t)^3 B_0 + 3(1-t)^2 t B_1 + 3(1-t) t^2 B_2 + t^3 B_3 \quad (4.6)$$

ここで、3 次ベジエ曲線の作図法を簡潔に説明する． Fig. 4.1.5 において B_0, B_1, B_2, B_3 はそれぞれ制御点である． ここで、ベジエ曲線上のある点 $P(t)$ は下記手順で求まる．

1. まず線分 B_0B_1 ，線分 B_1B_2 ，線分 B_2B_3 を $t : (1 - t)$ に分割する点 B'_0, B'_1, B'_2 を求める．
2. 次に線分 $B'_0B'_1$ ，線分 $B'_1B'_2$ を $t : (1 - t)$ に分割する点 B''_0, B''_1 を求める．
3. さらに線分 $B''_0B''_1$ を $t : (1 - t)$ に分割する点が $P(t)$ となる．

このとき、 t を $0 < t < 1$ の範囲で変化させて得られる $P(t)$ の軌跡が 3 次ベジエ曲線となる．

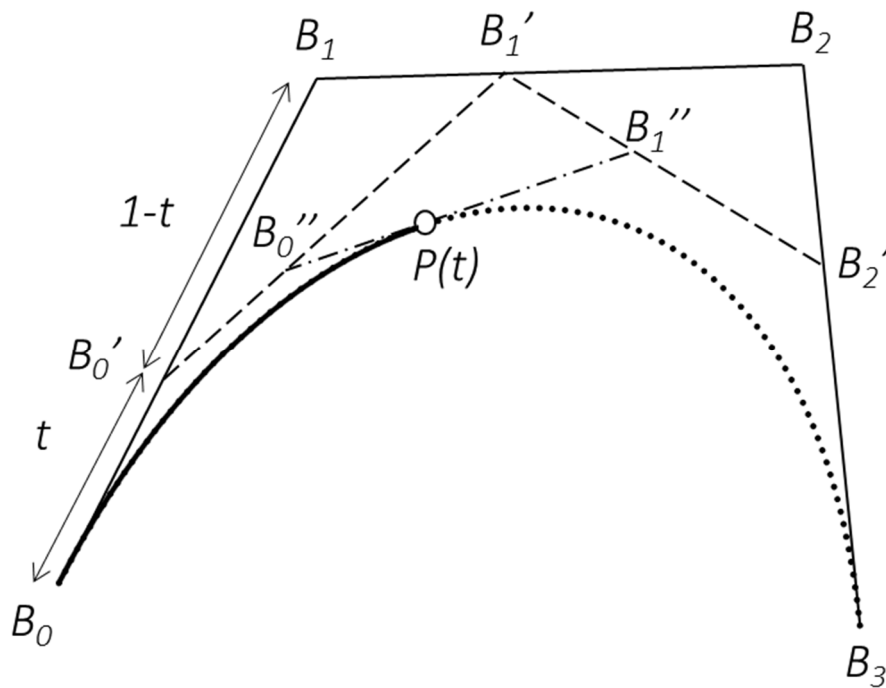


Fig. 4.1.5 Bezier Curve Drawing Method

4.1.3.2 制御点の配置

3 次ベジエ曲線は 4 つの制御点で描かれ、両端の 2 点は Fig. 4.1.4 に示したとおり、それぞれ START 地点（始点）と GOAL 地点（終点）であるが、その他の 2 点については始点と終点が要求する方向ベクトル成分を持ち、かつ運動モデルに無理のない曲率を成す位置に置かれることが望ましい。つまり、力学フィルタにより追従性の高い曲率となるような制御点を選ぶ必要がある。本節では、制御点配置の工夫について述べる。

ここで説明のため、始点、始点の次の点、終点の手前の点、終点をそれぞれ B_0, B_1, B_2, B_3 とする。このとき、 B_1, B_2 の制御点はそれぞれ始点、終点に近すぎても遠すぎても良好な擬似軌道を描くことができない。 B_0B_3 間の距離に対して B_0B_1, B_2B_3 間の距離が過度に近い場合はベジエ曲線の両端付近の曲率が過大となり、遠すぎる場合はベジエ曲線の間接点付近の曲率が過大となるため、機体の追従性が悪化してしまう。このような問題から、 B_0 の位置が探索状況で時々刻々と変化する軌道プランナでは、制御点 B_0B_1, B_2B_3 の長さをある固定値に定めることは好ましくない (Fig. 4.1.6)。

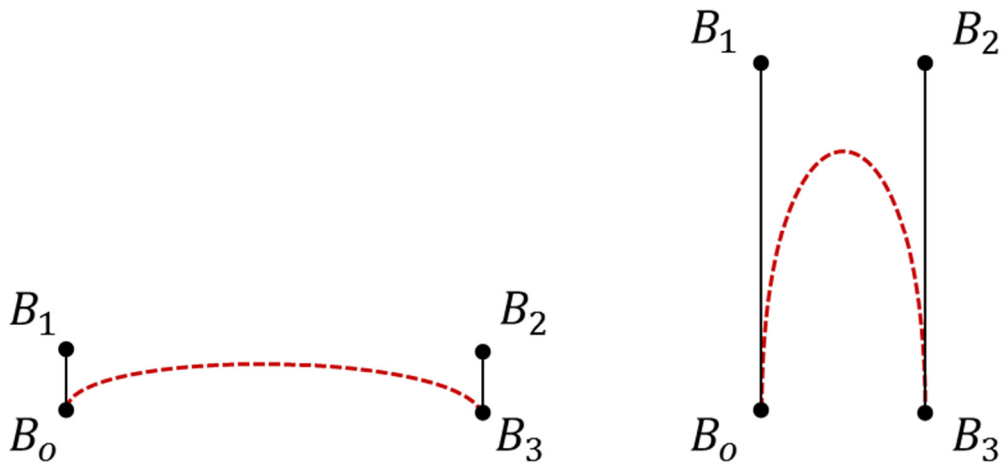


Fig. 4.1.6 Control Point Arrangement Issue

この曲率過大となる問題を解決するため、本手法では制御点を B_0B_3 間の距離に比例して可変とする。Fig. 4.1.7 に制御点の配置法に関する模式図を示す。まず、 B_0B_3 間の距離 $d_{B_0B_3}$ を求める (Eq. (4.7))。次に B_0 地点のノードが持つ単位方向ベクトルに対して、 $d_{B_0B_3}$ および事前に定めた比例定数 λ をかけた値と B_0 との和を B_1 と定義する (Eq. (4.8))。同様に、 B_3 からの差を B_2 とする (Eq. (4.9))。このように B_0 と B_3 の距離に比例して制御点を定めることにより、常に一定の割合で制御点を配置し、曲率過大に陥ることを防ぐ。

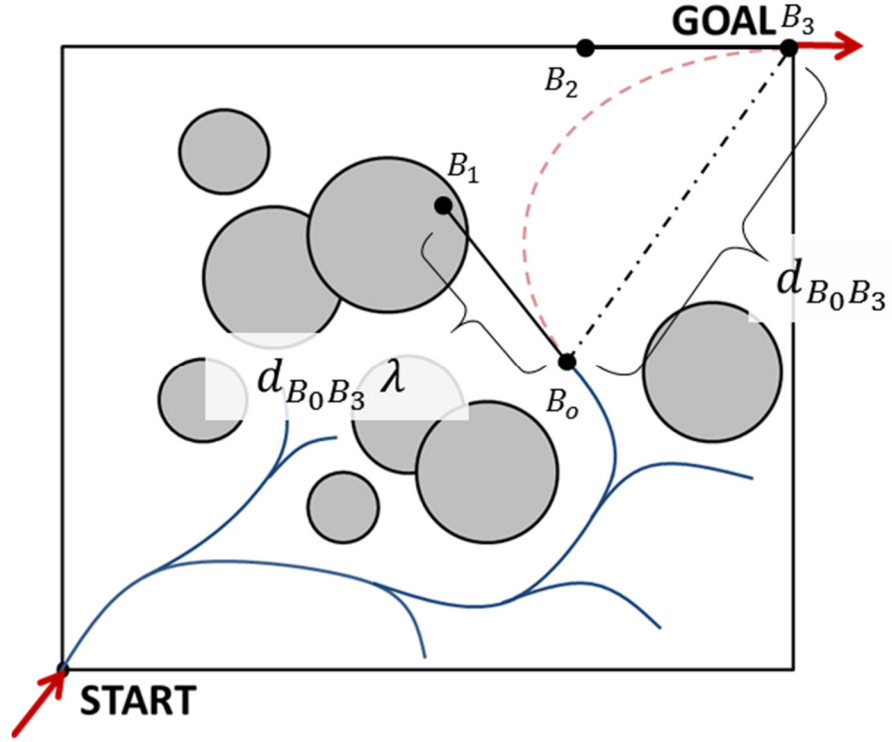


Fig. 4.1.7 Control Points Allocation

$$d_{B_0B_3} = \sqrt{(\xi_{B_3} - \xi_{B_0})^2 + (\eta_{B_3} - \eta_{B_0})^2 + (h_{B_3} - h_{B_0})^2} \quad (4.7)$$

$$\mathbf{B}_1 = \begin{bmatrix} \xi_{B_1} \\ \eta_{B_1} \\ h_{B_1} \end{bmatrix} = \begin{bmatrix} \xi_{B_0} \\ \eta_{B_0} \\ h_{B_0} \end{bmatrix} + \begin{bmatrix} \cos \gamma_{B_0} \sin \psi_{B_0} \\ \cos \gamma_{B_0} \cos \psi_{B_0} \\ \sin \gamma_{B_0} \end{bmatrix} d_{B_0B_3} \lambda \quad (4.8)$$

$$\mathbf{B}_2 = \begin{bmatrix} \xi_{B_2} \\ \eta_{B_2} \\ h_{B_2} \end{bmatrix} = \begin{bmatrix} \xi_{B_3} \\ \eta_{B_3} \\ h_{B_3} \end{bmatrix} - \begin{bmatrix} \cos \gamma_{B_3} \sin \psi_{B_3} \\ \cos \gamma_{B_3} \cos \psi_{B_3} \\ \sin \gamma_{B_3} \end{bmatrix} d_{B_0B_3} \lambda \quad (4.9)$$

4.1.3.3 比例定数 λ の決定

比例定数 λ は、理論的に機体モデルの追従性の高さを評価することは困難なため、飛行シミュレーションにより決定する。この飛行シミュレーションでは、初期位置、初期軌道角度および終端位置、終端軌道角度を事前に設定し、一度だけベジエ曲線を引き、力学フィルタの計算を行う。つまり、ベジエ曲線の更新は行わずにベジエ曲線が描く擬

似軌道と力学フィルタの描く真の軌道との誤差を計測することで、追従性の高さを数値的に評価する．なお、火星探査航空機と有翼ロケット実験機の両モデルで、方位角と経路角を各々独立で変化させた場合のシミュレーションを行い、それらの結果をもとに総合的に λ を決定する．

(1) 火星探査航空機のシミュレーション

火星探査航空機のモデルを用いて λ を変更しながらシミュレーションを行う．ここで、機体の初期状態および力学フィルタに用いた制御ゲインを Table 4.1.1 に記す．

Table 4.1.1 Initial States and Control Gains of Mars Aircraft

初期速度	[m/s]	70
初期推力	[N]	5
初期迎角	[°]	6
初期バンク角	[°]	0
P ゲイン $[K_{vp}, K_{gp}, K_{pp}]$	$[1/s^2, 1/s^2, 1/s^2]$	[1, 1, 1]
D ゲイン $[K_{vd}, K_{gd}, K_{pd}]$	$[1/s, 1/s, 1/s]$	[1, 1, 1]

まず、方位角変化に対する誤差を評価するための初期条件、終端条件を下表の通り定める．

Table 4.1.2 Initial and Terminal Trajectory States for Evaluating Direction Angle

初期位置 $[\xi_s, \eta_s, h_s]$	[m]	[0, 3000, 5000]
初期軌道角度 $[\gamma_s, \psi_s]$	[°]	[0, 90]
終端位置 $[\xi_g, \eta_g, h_g]$	[m]	[7000, -5000, 5000]
終端軌道角度 $[\gamma_g, \psi_g]$	[°]	[0, 180]

Fig. 4.1.8 に λ を 0.0~1.0 まで 0.1 毎に変化させた場合のベジエ曲線（破線）と、力学フィルタによって生成された軌道（実線）を示す. $\lambda = 0.0$ のとき、ベジエ曲線はスタートとゴールを結ぶ直線となり、急な初期の角度変化に軌道が追従できていない. その後、 $\lambda = 0.1 \sim 0.2$ においても初期の大きな角度変化に付いて行けず誤差が大きいが、 $\lambda = 0.3 \sim 0.6$ ではベジエ曲線が緩やかになるため機体が良く追従していることが分かる. そして、 $\lambda = 0.7$ 以降において今度は中間点付近での曲率が大きくなっているため再び追従性が低下しており、擬似軌道から逸脱している.

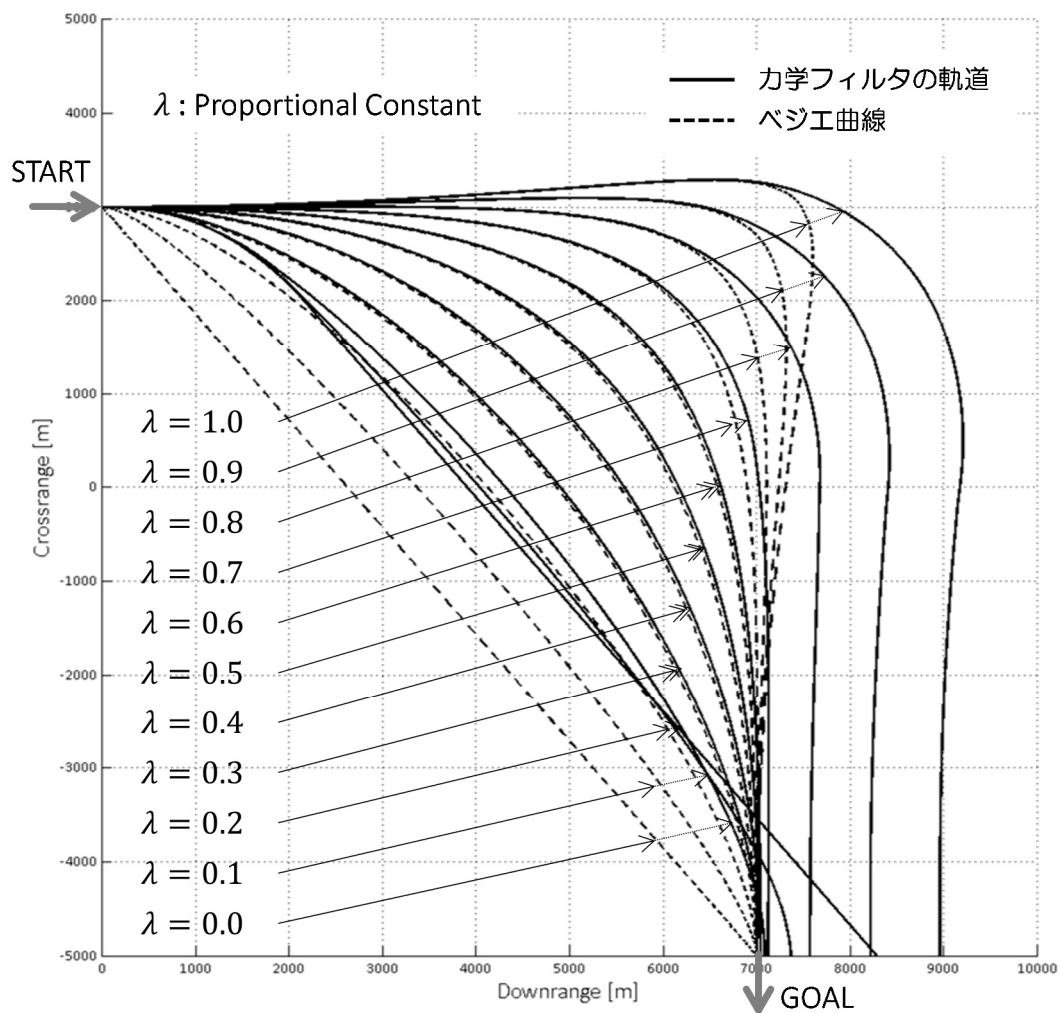


Fig. 4.1.8 Bezier Curve and Resultant Trajectory of λ

ここで得られた軌道を数値的に評価するため、時々刻々の角度誤差の累積和を計算した。ここでは、累積誤差が小さくなる λ を良好な結果とするが、飛行距離が長い経路は累積和が増加してしまうため全軌道を公平に評価できない。そこで、飛行距離の影響を消すため単位飛行距離あたりの誤差として評価する。評価式を Eq. (4.10)に示す。

$$E = \frac{\sum_{ID=0}^{end} \sqrt{(\psi_{refID} - \psi)^2}}{Flight\ Distance} \quad (4.10)$$

上式に基づいて軌道誤差を計算し比較した結果を Fig. 4.1.9 に示す。結果は軌道の図を用いた考察と同様に、 $\lambda = 0.3 \sim 0.6$ のとき良好な結果を示し、さらに $\lambda = 0.6$ となると、最も追従性が高いことが分かった。

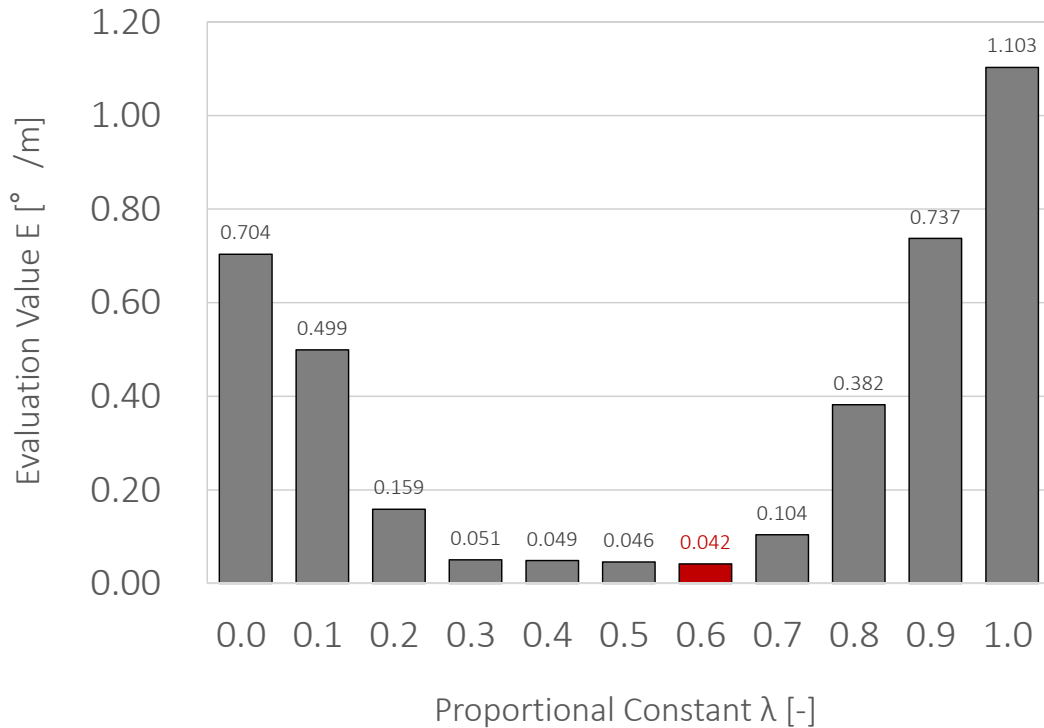


Fig. 4.1.9 Evaluation Value E of λ

次に，経路角変化に対する追従性も同様に評価する．用いた初期条件，終端条件を Table 4.1.3 に示す．なお，機体の初期状態およびゲインは方位角の評価時と同様である．

Table 4.1.3 Initial and Terminal Trajectory States for Evaluating Flight Path Angle

初期位置 $[\xi_s, \eta_s, h_s]$	[m]	[0, 0, 4000]
初期軌道角度 $[\gamma_s, \psi_s]$	[°]	[0, 90]
終端位置 $[\xi_g, \eta_g, h_g]$	[m]	[10000, 0, 6000]
終端軌道角度 $[\gamma_g, \psi_g]$	[°]	[0, 90]

Fig. 4.1.10 に軌道生成の結果を示す． $\lambda = 0.0 \sim 0.1$ において初期の急激な角度変化に付いて行けず誤差が大きくなっている． $\lambda = 0.3 \sim 0.5$ ではベジエ曲線が緩やかになるため機体が良く追従している．そして， $\lambda = 0.6$ 以降は中間点付近での曲率が大きくなっているため再び追従性が低下している．

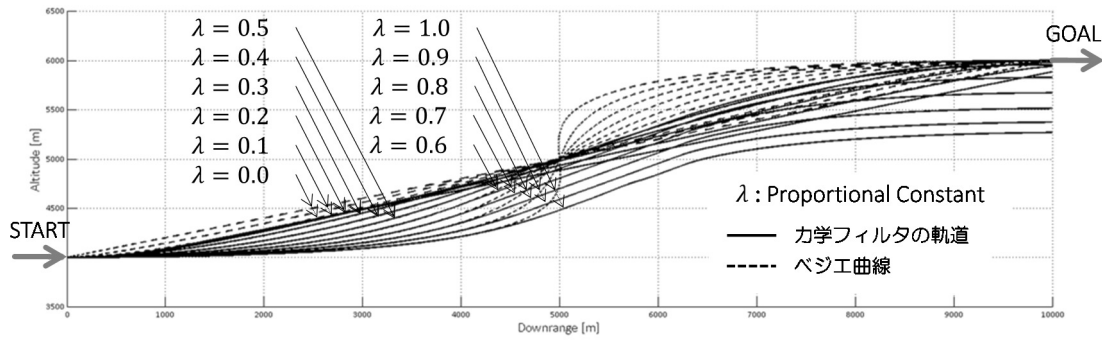


Fig. 4.1.10 Bezier Curve and Resultant Trajectory of λ

さらに，先ほどと同様の評価値を用いた比較を行う．評価式は Eq.(4.10)を経路角誤差に変更した Eq. (4.11)を用いる．

$$E = \frac{\sum_{ID=0}^{end} \sqrt{(\gamma_{refID} - \gamma)^2}}{Flight Distance} \quad (4.11)$$

上式に基づいてそれぞれの軌道を評価し比較した結果を Fig. 4.1.11 に示す。結果は $\lambda = 0.3 \sim 0.6$ 程度のとき良好な結果を示し、 $\lambda = 0.3$ となるとき、最も追従性が高いことが分かった。

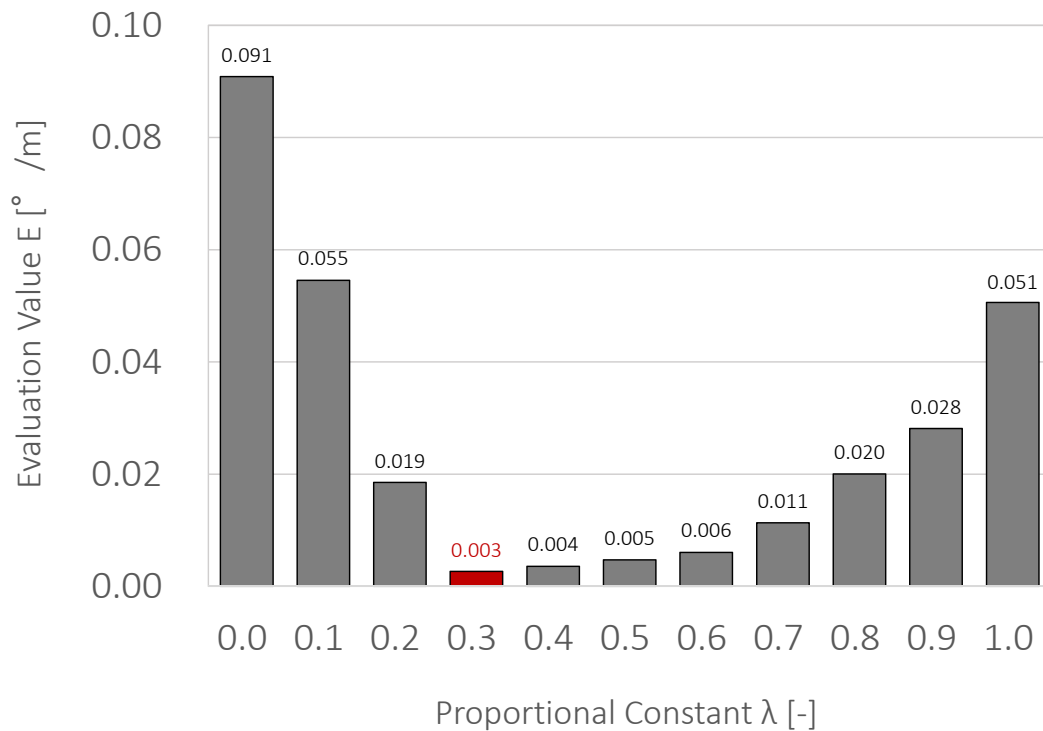


Fig. 4.1.11 Evaluation Value E of λ

以上の結果より、方位角変化に対する評価値は $\lambda = 0.3 \sim 0.6$ において殆ど変わらないこと、経路角変化への追従性は $\lambda = 0.3$ において良好であり、 λ が小さければ小回り軌道を描けることから、火星探査航空機のモデルにおいてベジエ曲線に用いる制御点比例定数を $\lambda = 0.3$ と定める。ただし、この値は本シミュレーション条件の結果をもとに決定されており、条件が変わると最適解も多少変化することが考えられるため、 λ の決定論については、改善の余地がある。

(2) 有翼ロケット実験機のシミュレーション

次に、有翼ロケット実験機のモデル（WIRES#015）を用いて同様のシミュレーションを行う．機体の初期状態および制御ゲインを Table 4.1.4 に記す．

Table 4.1.4 Initial States and Control Gains of WIRES#015

初期速度	[m/s]	70
初期推力	[N]	0
初期迎角	[°]	12
初期バンク角	[°]	0
P ゲイン $[K_{vp}, K_{gp}, K_{pp}]$	$[1/s^2, 1/s^2, 1/s^2]$	$[1, 1, 1]$
D ゲイン $[K_{vd}, K_{gd}, K_{pd}]$	$[1/s, 1/s, 1/s]$	$[1, 1, 1]$

まず、方位角変化に対する誤差を評価するための初期条件、終端条件を Table 4.1.5 の通り定める．

Table 4.1.5 Initial and Terminal Trajectory States for Evaluating Direction Angle

初期位置 $[\xi_s, \eta_s, h_s]$	[m]	$[0, 1500, 2500]$
初期軌道角度 $[\gamma_s, \psi_s]$	[°]	$[0, 90]$
終端位置 $[\xi_g, \eta_g, h_g]$	[m]	$[4000, -2500, 2500]$
終端軌道角度 $[\gamma_g, \psi_g]$	[°]	$[0, 180]$

Fig. 4.1.12 にシミュレーション結果を示す． $\lambda = 0.0 \sim 0.1$ において誤差が大きいが， $\lambda = 0.2 \sim 0.9$ 付近ではいずれも機体が良く追従している．そして， $\lambda = 1.0$ は中間点付近での旋回時に追従性が低下し，擬似軌道から大きく逸脱している．

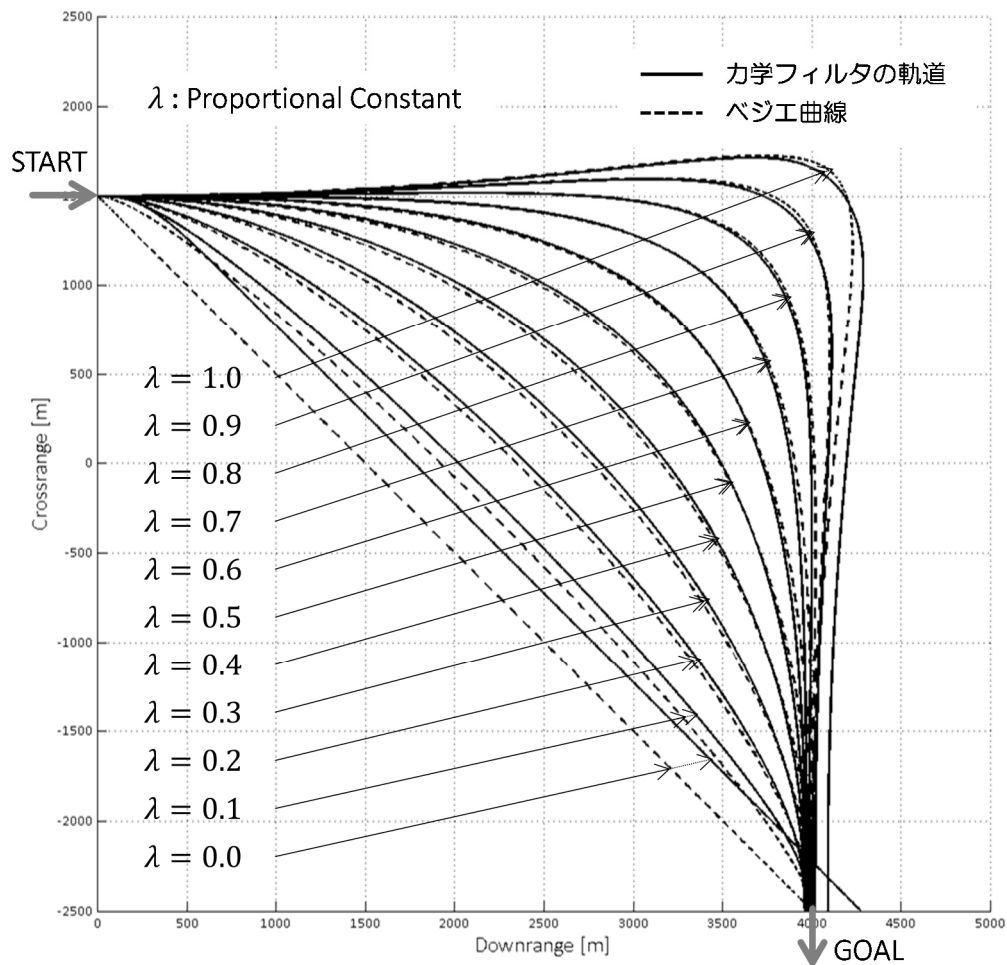


Fig. 4.1.12 Bezier Curve and Resultant Trajectory of λ

火星探査航空機モデル同様に，数値的に評価を行う．評価値の比較結果を Fig. 4.1.13 に示す．結果より， $\lambda = 0.2 \sim 0.8$ 程度のときいずれも良好な結果を示しているが， $\lambda = 0.7$ となると，最も追従性が高いことが分かった．

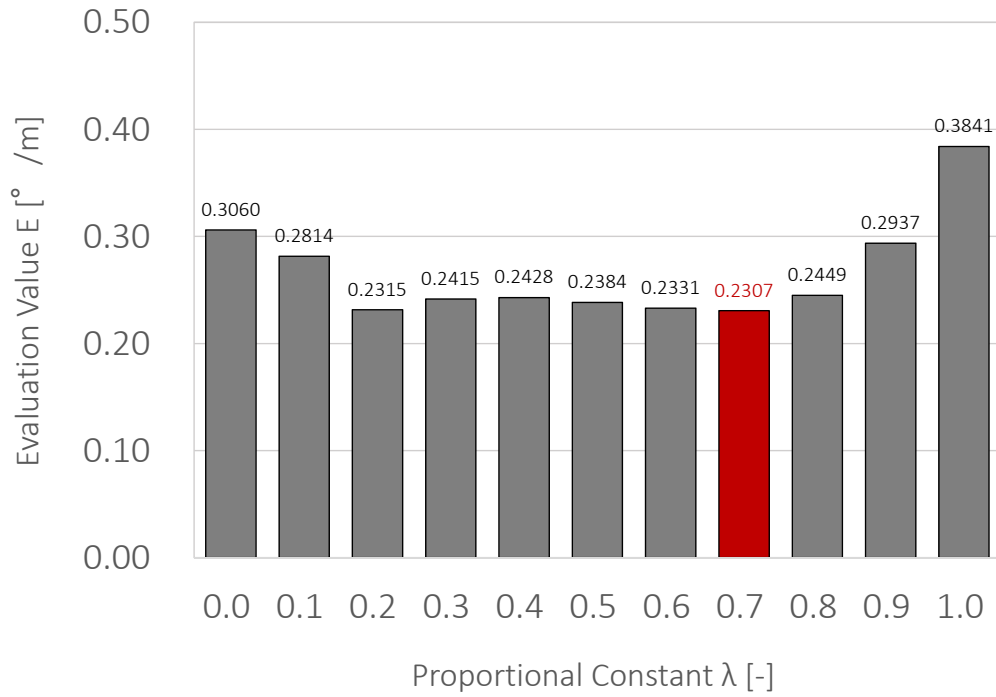


Fig. 4.1.13 Evaluation Value E of λ

次に，経路角変化に対する追従性も評価する．用いた初期条件，終端条件を Table 4.1.6 に示す．なお，機体の初期状態およびゲインは，方位角の評価時と同様である．

Table 4.1.6 Initial and Terminal Trajectory States for Evaluating Flight Path Angle

初期位置 $[\xi_s, \eta_s, h_s]$	[m]	[0, 0, 3500]
初期軌道角度 $[\gamma_s, \psi_s]$	[°]	[0, 90]
終端位置 $[\xi_g, \eta_g, h_g]$	[m]	[5000, 0, 1500]
終端軌道角度 $[\gamma_g, \psi_g]$	[°]	[0, 90]

Fig. 4.1.14 に軌道生成の結果を示す． $\lambda = 0.0 \sim 0.1$ において初期位置付近の急激な角度変化に対する誤差が大きくなっている． $\lambda = 0.2 \sim 0.4$ ではベジエ曲線が緩やかになり機体が良く追従している．そして， $\lambda = 0.5$ 以降は中間点付近での曲率が大きくなっているため再び追従性が低下している．

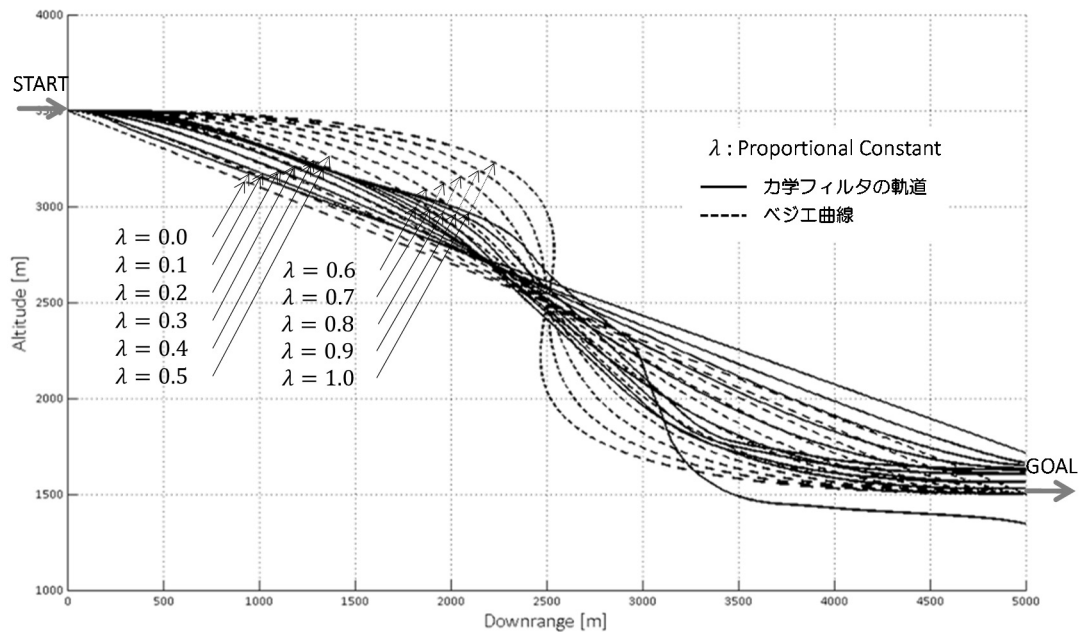


Fig. 4.1.14 Bezier Curve and Resultant Trajectory of λ

さらに，評価値を用いた比較を行う．評価値の比較結果を Fig. 4.1.15 に示す．結果は $\lambda = 0.2 \sim 0.3$ のとき良好な結果を示し， $\lambda = 0.3$ となるとき，最も追従性が高いことが分かった．

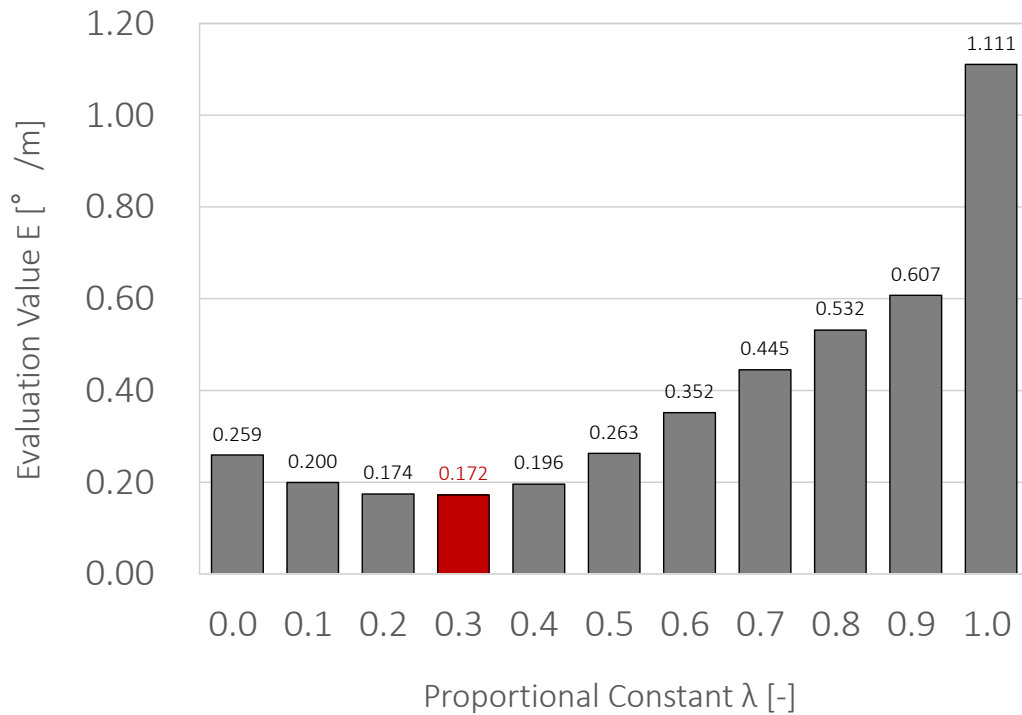


Fig. 4.1.15 Evaluation Value of λ

以上の結果を踏まえて、方位角変化については $\lambda = 0.2 \sim 0.8$ でいずれも良好な追従性を示し、経路角変化への追従性は $\lambda = 0.2 \sim 0.3$ において良好となった。よって、 λ を小さくして小回り軌道を得るため、有翼ロケット実験機モデルにおいてベジエ曲線に用いる制御比例定数を $\lambda = 0.2$ と定める。ただし、この値は本シミュレーション条件の結果をもとに決定されており、条件が変わると最適解も多少変化することが考えられるため、 λ の決定論については、改善の余地がある。

4.2 軌道生成シミュレーション

4.2.1 火星探査航空機

4.2.1.1 シミュレーション条件

火星探査航空機のシミュレーション条件は以下の通りである．火星に存在する山々を想定して，障害物の形状は円錐形状として扱い，障害物の数を 10~50 個まで 20 個刻みで増加させた場合のシミュレーションを行う．

Table 4.2.1 Simulation Conditions for Mars Aircraft

大気密度	[kg/m ³]	0.0118
重力加速度	[m/s ²]	3.2
スタート位置 $[\xi, \eta, h]$	[m]	[0, 5000, 2500]
スタート軌道角度 $[\psi, \gamma]$	[°]	[135, 0]
ゴール位置 $[\xi, \eta, h]$	[m]	[10000, -5000, 2500]
ゴール軌道角度 $[\psi, \gamma]$	[°]	[135, 0]
障害物の形状	[-]	円錐
障害物寸法	[m]	ランダム
障害物位置	[m]	ランダム
障害物数	[個]	10, 30, 50

ここで，軌道プランナに用いられる各種パラメータは以下のとおりである．

Table 4.2.2 Trajectory Planner Parameters for Mars Aircraft

ランダム方位角指令値の範囲	[°]	$-180 \leq \psi_{rand} \leq 180$
ランダム経路角指令値の範囲	[°]	$-20 \leq \gamma_{rand} \leq 20$
枝成長時間 T	[s]	30
ベジエ制御点比例定数 λ	[-]	0.3
衝突回避マージン	[m]	12.1 (片翼長の 10 倍)
探索終了条件：許容距離誤差	[m]	100 以下
探索終了条件：許容角度誤差	[°]	10 以下

また，機体に与える初期条件は以下の通りである．

Table 4.2.3 Initial States for Simulation

時間 t	[s]	0
速度 V	[m/s]	70
方位角 ψ	[°]	135
推力	[N]	2.5
迎角	[°]	6
バンク角	[°]	0

4.1 節で示した制御器において，本シミュレーションで使用する目標速度は $V_{ref} = 70$ [m/s]とし，各制御ゲインは下表を用いた．

Table 4.2.4 Control Gains

K_{vp}	[1/s ²]	1.0	K_{vd}	[1/s]	1.0
K_{gp}	[1/s ²]	1.0	K_{gd}	[1/s]	1.0
K_{pp}	[1/s ²]	1.0	K_d	[1/s]	1.0

力学フィルタには実用的な軌道を生成させるため、各コマンドに制限値を設けている。設定した制限値を以下に示す。

Table 4.2.5 Command Limiters

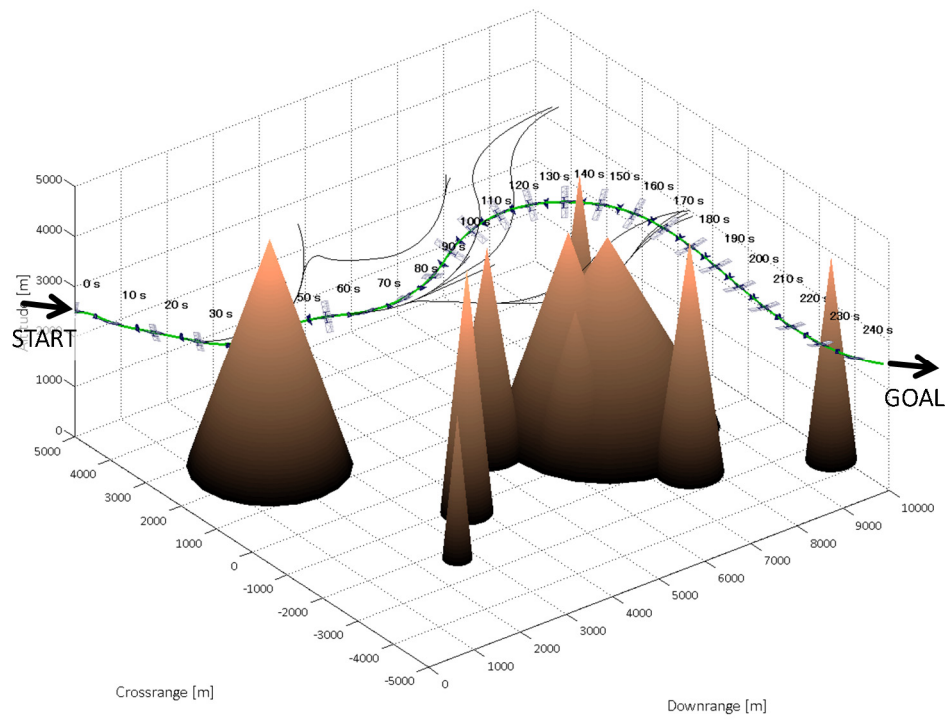
$0 \leq T_{com} \leq 5$	[N]	$-5 \leq \dot{T}_{com} \leq 5$	[N/s]
$-7 \leq \alpha_{com} \leq 7$	[°]	$-7 \leq \dot{\alpha}_{com} \leq 7$	[°/s]
$-30 \leq \phi_{com} \leq 30$	[°]	$-30 \leq \dot{\phi}_{com} \leq 30$	[°/s]

計算環境は Table 3.6.6 と同様である。

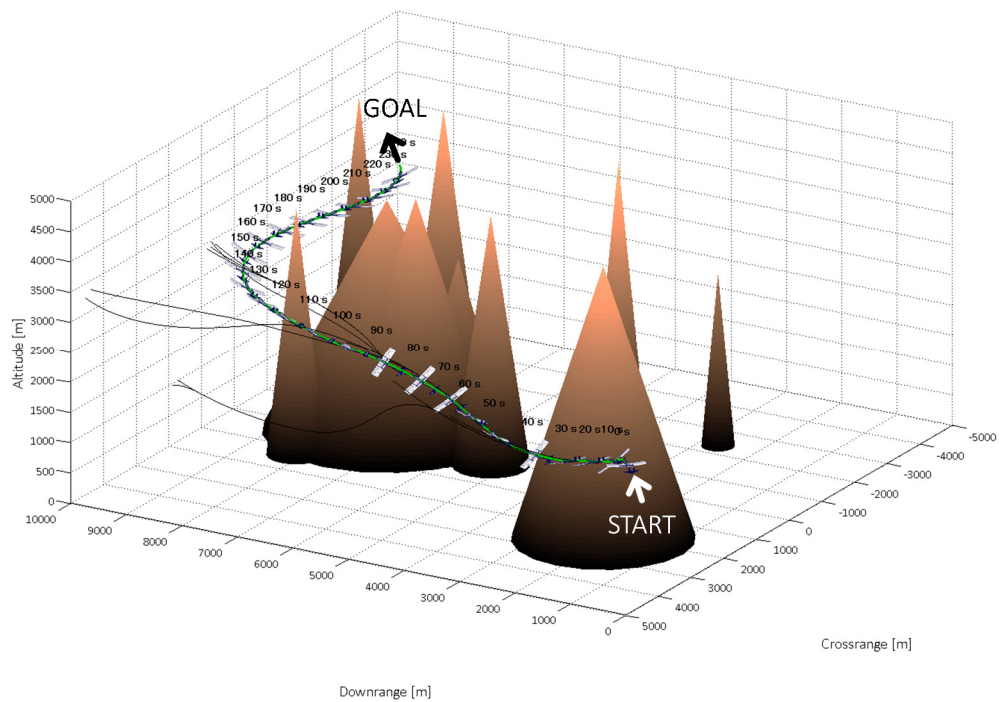
4.2.1.2 シミュレーション結果

(1) 障害物 10 個

障害物の数を 10 個とした軌道生成結果を Fig. 4.2.1 に示す。黒の実線は、融合型軌道プランナで探索された全てのツリーを示し、最終的にゴールへ到達できた軌道は緑の実線で示す。また、機体モデルを図示し機体上部の数字は、スタートからの飛行時間を表している。ただし、視認性向上のため機体モデルは、実寸の 100 倍にスケールアップして表示している。結果より障害物 10 個の環境において、探索初期の段階(70 秒以前)で複数のランダム探索が実行されツリーが成長していることが分かり、その後、山と山の間を抜けて障害物を回避し目的地へ到達する良好な軌道が生成できていることが確認できる。

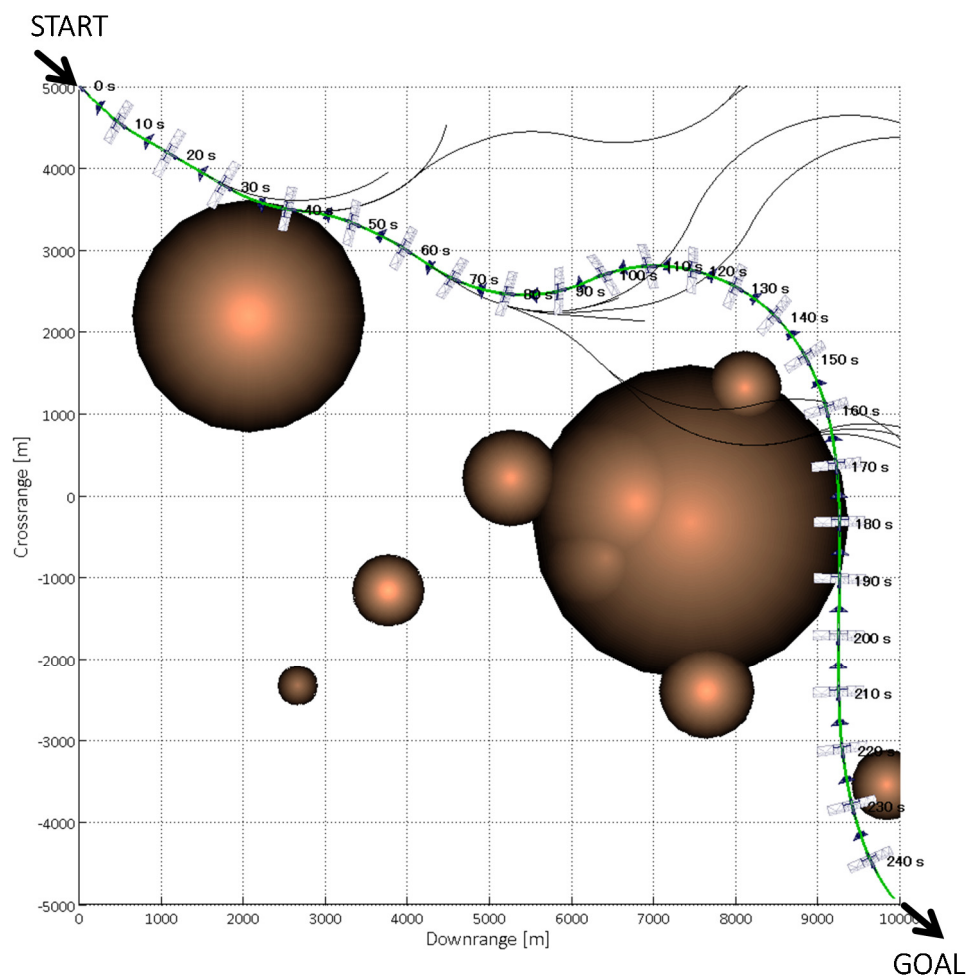


(a) Bird's-Eye View 1



(b) Bird's-Eye View 2

Fig. 4.2.1 Simulation Result (Obstacle Number 10)
(to be continued)



(c) Top View

Fig. 4.2.1 Simulation Result (Obstacle Number 10)
(to be concluded)

Fig. 4.2.2 には、得られたコマンド時系列データを示す．その結果より、推力、迎角、バンク角の各コマンドが制限範囲内の実用的なコマンドとして生成されていることが分かる．また Fig. 4.2.3 には、速度と軌道角度の指令値（破線）および実際の機体の状態（実線）を示しており、各目標値に対して機体は概ね追従している．約 65 秒から 95 秒の間など、目標値（特に方位角）が大きく変化している箇所では枝生成時間 T (30 秒) 以内に十分に収束していないが、ランダムに軌道を探索するという軌道プランナのココンセプトにおいて収束性は問題にはならない．95 秒以降は、方位角、経路角の目標値が滑らかに遷移しているが、これはベジエ曲線の疑似軌道によって与えられた目標値を表

している。その後、125 秒で目標値が不連続に変化するが、ここでは、枝生成時間 T 秒の運動計算を終えて、疑似軌道が機体の現在値を使って新しく更新されたことを示す。なお、軌道の終点における目標位置との誤差は 97.0 [m]であり、また方位角および経路角の誤差はそれぞれ 2.35 [°], 1.38 [°]となり、探索終了条件を満たす軌道が生成された。

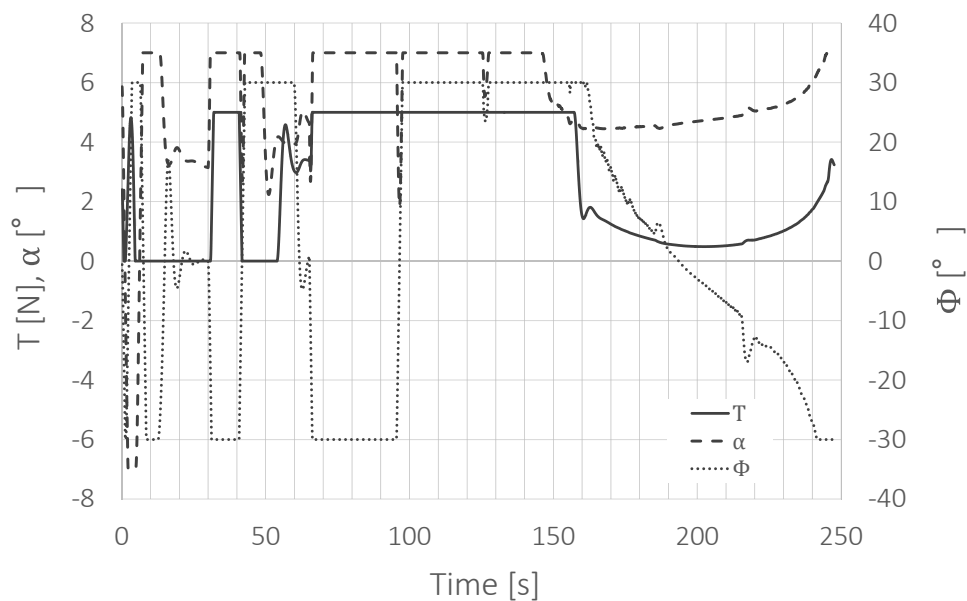


Fig. 4.2.2 Command Time Histories

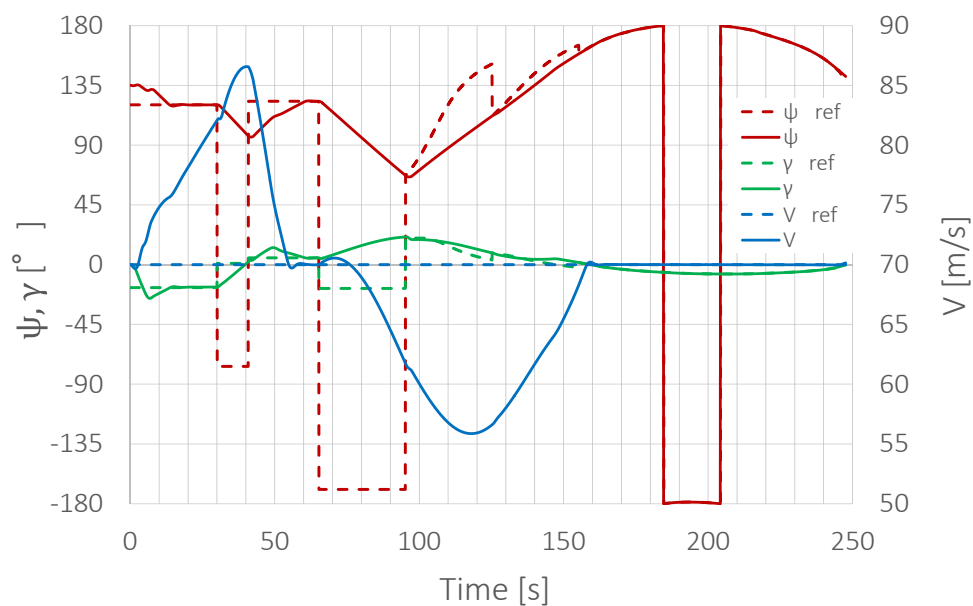
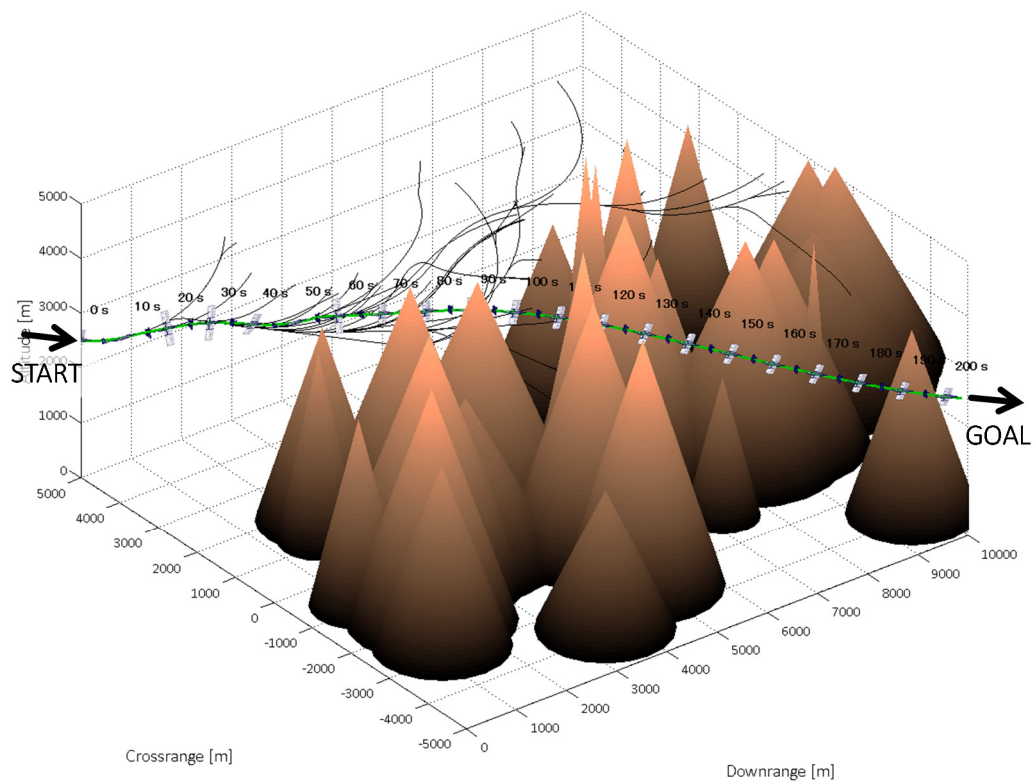


Fig. 4.2.3 Reference and Response State Variables

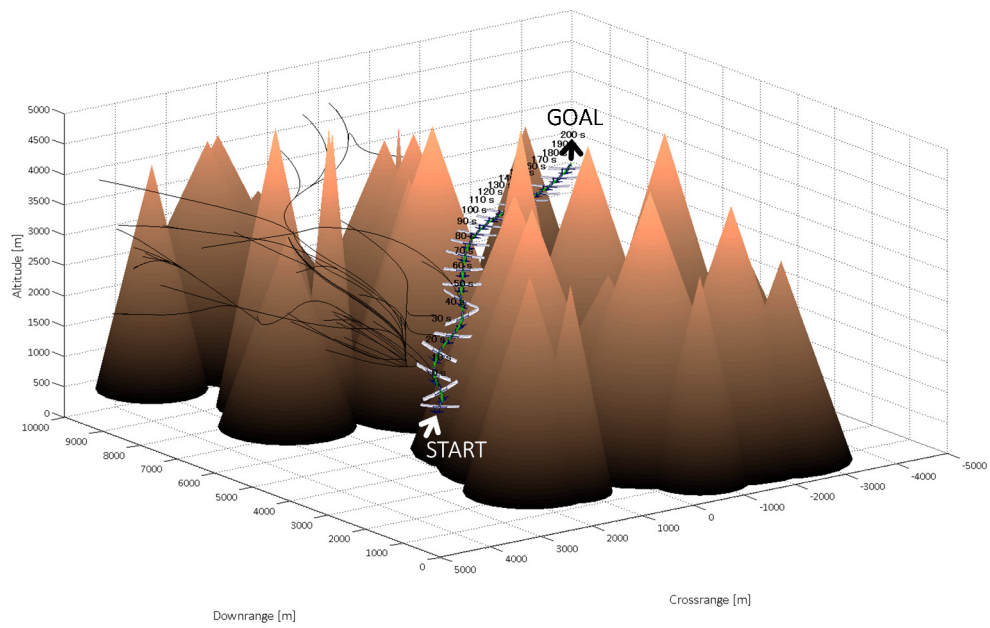
(2) 障害物 30 個

次に障害物の数を 30 個に増やした場合の軌道生成結果を示す (Fig. 4.2.4). 中央付近に高い山々が多く, それらを避けるように周囲から迂回して目的地まで到達する探索結果を得た. しかし Fig. 4.2.4(c) より, ランダム探索において左旋回が多く行われていることが分かる. これは, 制御器とランダムに与えられる方位角の関係に起因する. つまり, 現在の機体の方位角が 135 度のときは, ランダムに与えられる方位角が $135 < \psi_{rand} \leq 180$ の場合のみ右旋回となり, $-180 < \psi_{rand} \leq 135$ の場合には左旋回を行う. そのためランダム探索は, 左旋回を多く試みている探索結果となった.

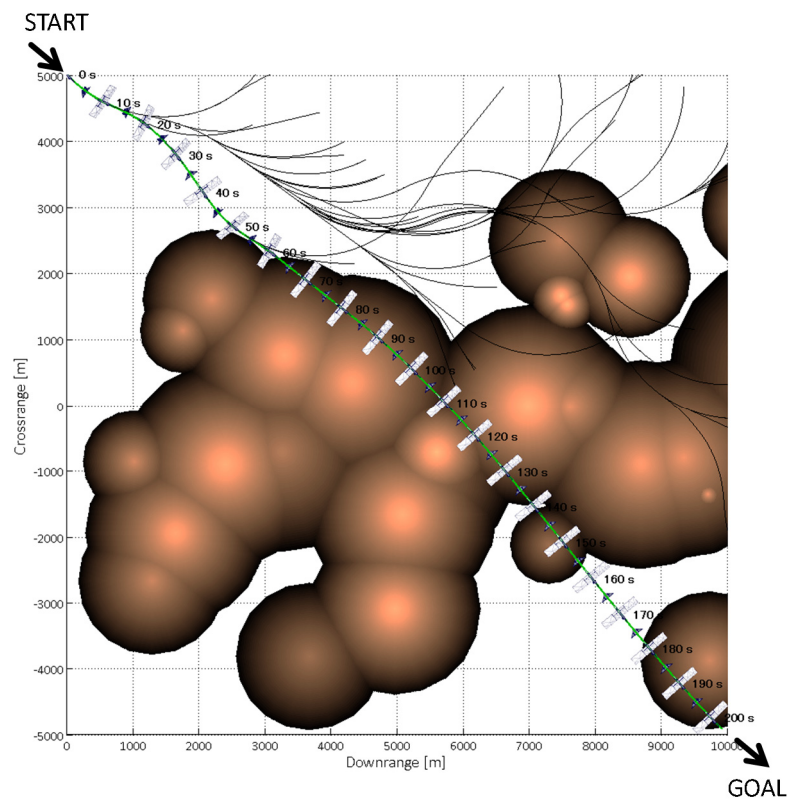


(a) Bird's-Eye View 1

Fig. 4.2.4 Simulation Result (Obstacle Number 30)
(to be continued)



(b) Bird's-Eye View 2



(c) Top View

Fig. 4.2.4 Simulation Result (Obstacle Number 30)
(to be concluded)

また本ケースも同様に，コマンド時系列データおよび指令値と機体状態量のグラフを Fig. 4.2.5, Fig. 4.2.6 にそれぞれ示す．ランダムに探索が進められた段階（85 秒以前）では，大きなコマンドの変動が見られるものの，障害物 10 個で得られた軌道と比較して直線的な軌道が生成されているため概ね安定したコマンド履歴となっている．また，方位角，経路角もランダムに与えられた値に追従する傾向が見られる．

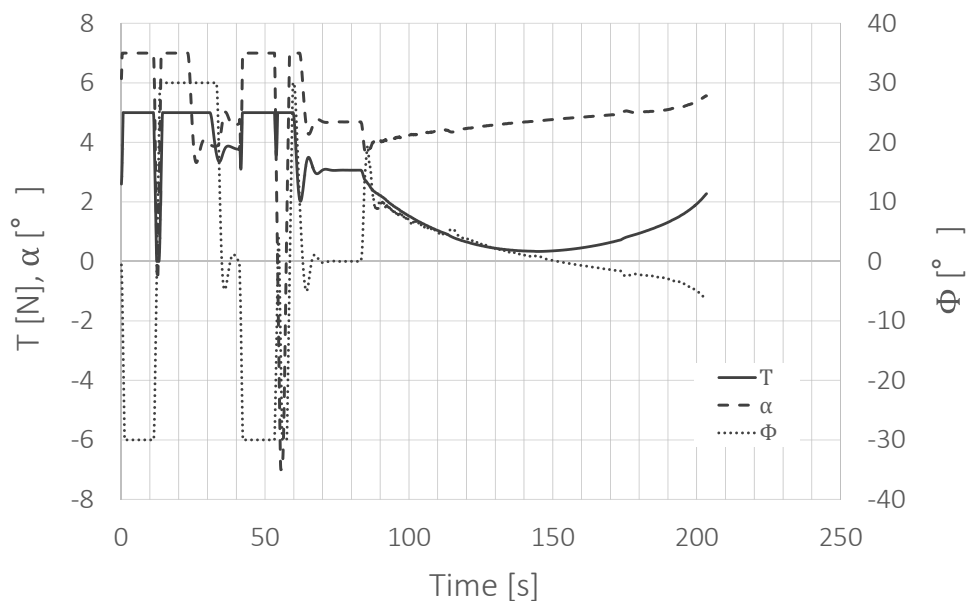


Fig. 4.2.5 Command Time Histories

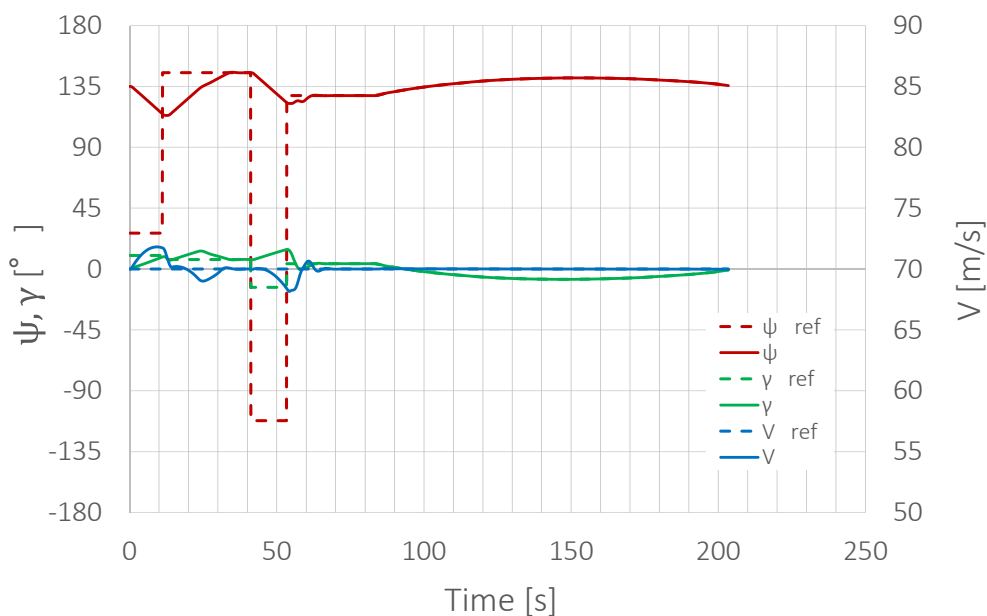
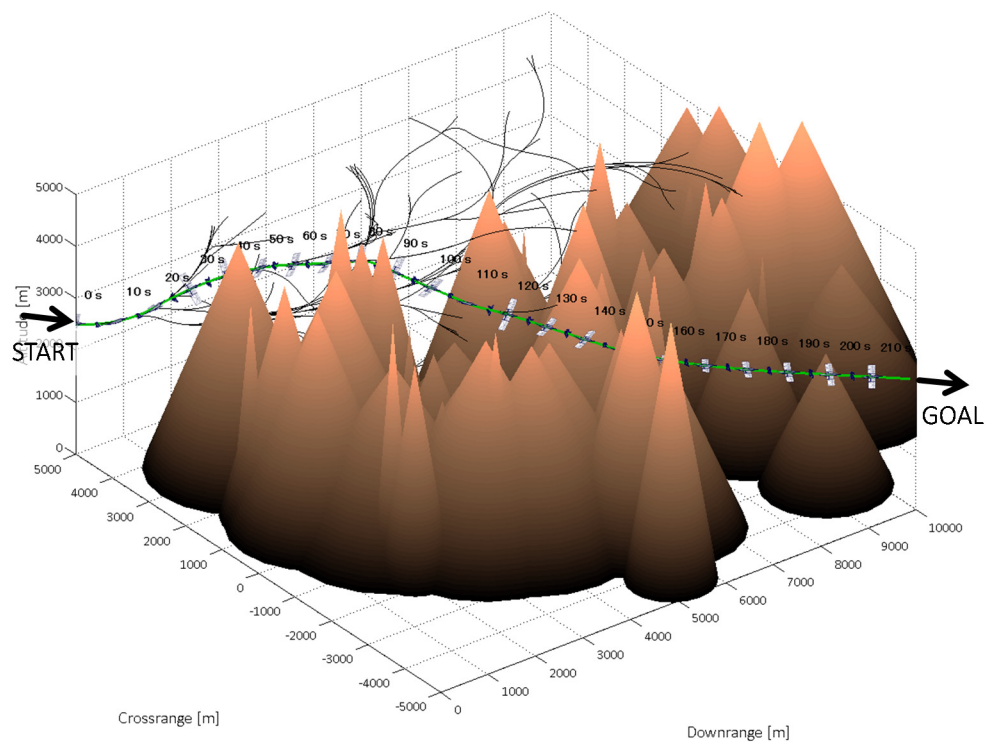


Fig. 4.2.6 Reference and Response States Variables

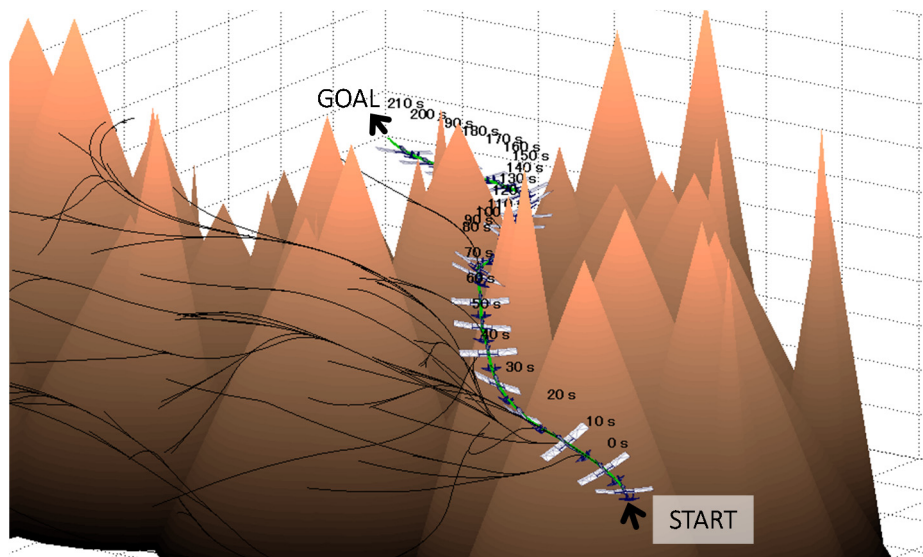
(3) 障害物 50 個

さらに、障害物の数を 50 個に増やした場合の軌道生成結果を示す (Fig. 4.2.7). 障害物数が 10 個や 30 個と比較して多くのランダム探索が行われたことが分かる. 本結果においても左旋回が多く行われているのは, 先述の制御器とランダムに与えられる方位角の関係性に起因する不具合である. しかし, このように障害物環境が極めて複雑な状況下でも, 探索解を得ることができた.

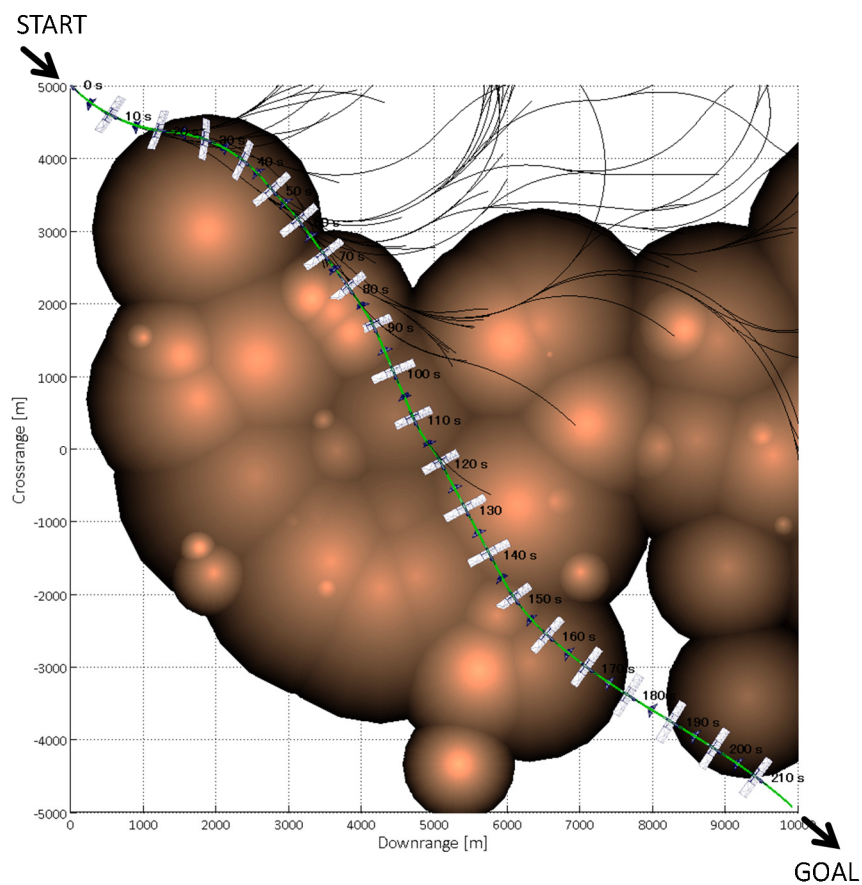


(a) Bird's-Eye View 1

Fig. 4.2.7 Simulation Result (Obstacle Number 50)
(to be continued)



(b) Bird's-Eye View 2



(c) Top View

Fig. 4.2.7 Simulation Result (Obstacle Number 50)
(to be concluded)

本ケースも同様に、コマンド時系列データおよび、指令値と機体状態量のグラフを Fig. 4.2.8, Fig. 4.2.9 にそれぞれ示す。それぞれのコマンド値は制限内で良好に生成されているが、軌道が複雑に蛇行しているため変動が激しい結果となった。方位角、経路角についてはランダムに与えられた値に追従する傾向が見られる。速度については、迎え角とバンク角の影響を受けて大きな増減が見られる。

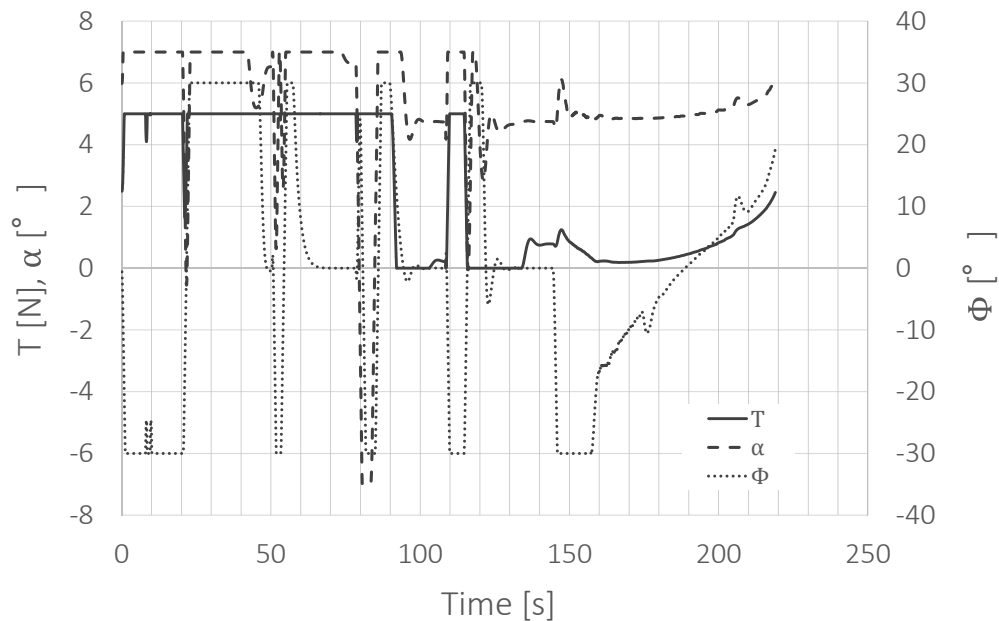


Fig. 4.2.8 Command Time Histories

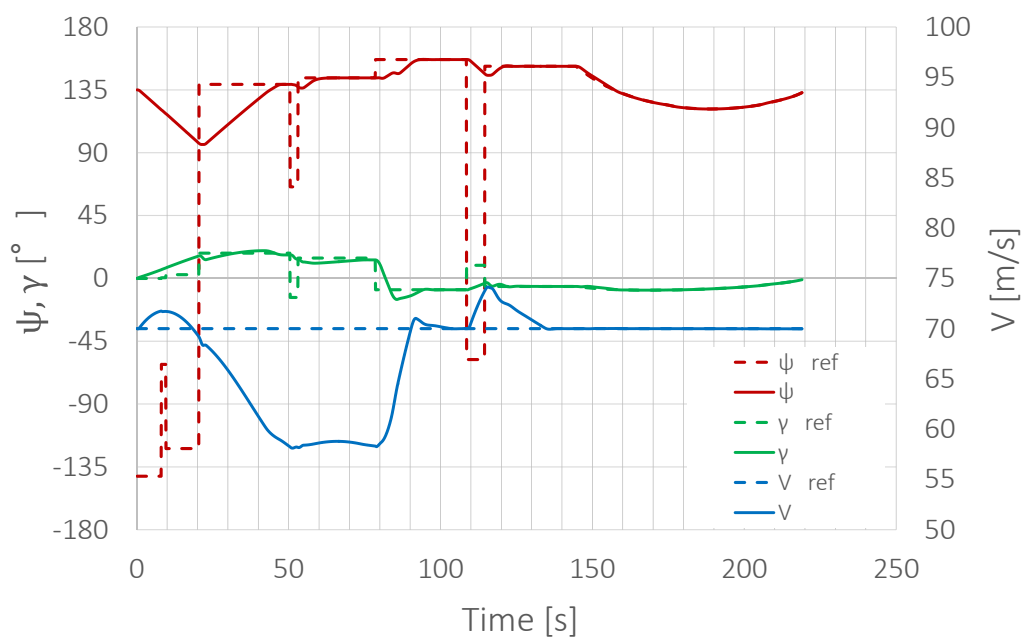


Fig. 4.2.9 Reference and Response State Variables

4.2.2 有翼ロケット実験機

4.2.2.1 シミュレーション条件

有翼ロケット実験機 **WIRES#015** の軌道生成シミュレーション条件を表に示す．本シミュレーションでは，飛行禁止区域を円形で設定し，その上空も全て通過禁止とした．飛行禁止区域（障害物）の数を 5~15 個まで 5 個刻みで増加させた場合のシミュレーションを行う．

Table 4.2.6 Simulation Conditions for WIRES

大気密度	[kg/m ³]	1.12
重力加速度	[m/s ²]	9.807
スタート位置 $[\xi, \eta, h]$	[m]	[500, -2500, 4500]
スタート軌道角度 $[\psi, \gamma]$	[°]	[0, 0]
ゴール位置 $[\xi, \eta, h]$	[m]	[5000, 2000, 1500]
ゴール軌道角度 $[\psi, \gamma]$	[°]	[90, 0]
障害物の形状	[-]	無限に長い円柱
障害物寸法	[m]	ランダム
障害物位置	[m]	ランダム
障害物数	[個]	5, 10, 15

軌道プランナに用いる各種パラメータは次のとおりである．本シミュレーションでは有翼ロケットの帰還軌道生成を行う．帰還時は無推力の滑空飛行であるため，ランダム経路角指令値の上限値は 0 [°]としている．

Table 4.2.7 Trajectory Planner Parameters for WIRES

ランダム方位角指令値の範囲	[°]	$-180 \leq \psi_{rand} \leq 180$
ランダム経路角指令値の範囲	[°]	$-30 \leq \gamma_{rand} \leq 0$
枝成長時間 T	[s]	10
ベジエ制御点比例定数 λ	[-]	0.2
衝突回避マージン	[m]	12.59 (片翼長の 10 倍)
探索終了条件：許容距離誤差	[m]	50 以下
探索終了条件：許容角度誤差	[°]	10 以下

機体に与える初期条件は以下の通りである.

Table 4.2.8 Initial States for Simulation

時間 t	[s]	0
速度 V	[m/s]	70
方位角 ψ	[°]	0
推力 T	[N]	0
迎角 α	[°]	12
バンク角 ϕ	[°]	0

本シミュレーションで使用する目標速度は $V_{ref} = 70$ [m/s]とし, 各制御ゲインは Table 4.2.9 を用いた.

Table 4.2.9 Control Gains

K_{vp}	$[1/s^2]$	1.0	K_{vd}	$[1/s]$	1.0
K_{gp}	$[1/s^2]$	1.0	K_{gd}	$[1/s]$	1.0
K_{pp}	$[1/s^2]$	1.0	K_d	$[1/s]$	1.0

また、各コマンドに設定した制限値を Table 4.2.10 に示す.

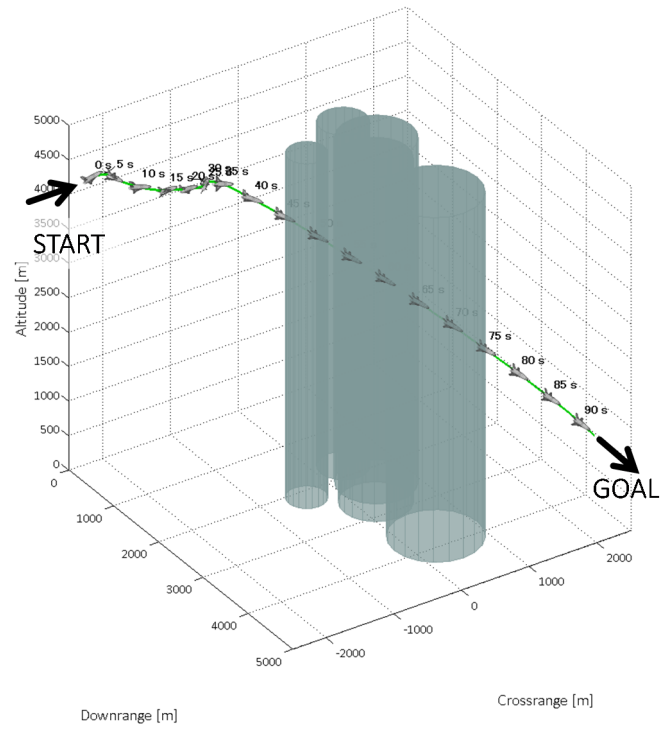
Table 4.2.10 Command Limiters

$0 \leq T_{com} \leq 0$	$[N]$	$-0 \leq \dot{T}_{com} \leq 0$	$[N/s]$
$-15 \leq \alpha_{com} \leq 15$	$[^\circ]$	$-15 \leq \dot{\alpha}_{com} \leq 15$	$[^\circ/s]$
$-60 \leq \phi_{com} \leq 60$	$[^\circ]$	$-60 \leq \dot{\phi}_{com} \leq 60$	$[^\circ/s]$

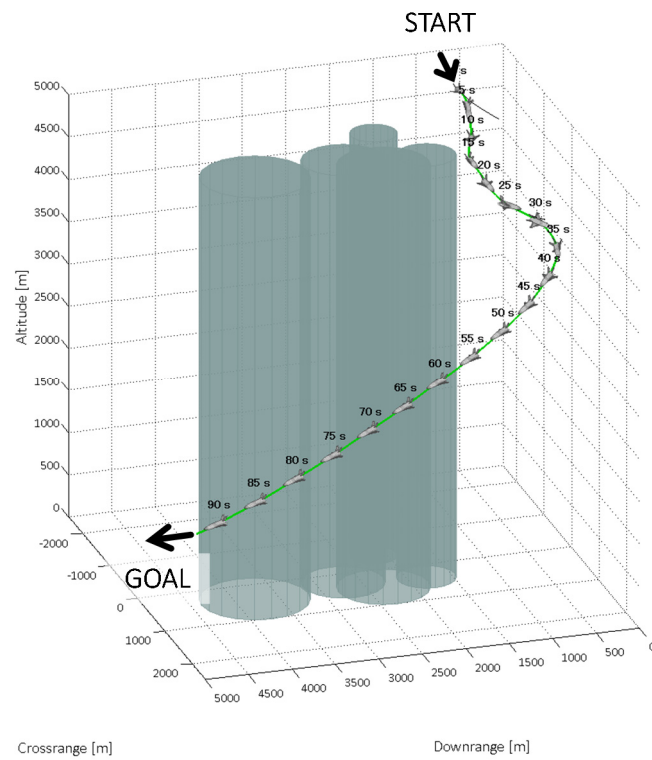
4.2.2.2 シミュレーション結果

(1) 障害物 5 個

障害物の数を 5 個とした軌道生成結果を Fig. 4.2.10 に示す. 視認性向上のため機体モデルは実寸の 100 倍にスケールアップして表示している. シミュレーションより, 5 つの飛行禁止区域を避けてゴールに到達する良好な滑空軌道が生成できた.

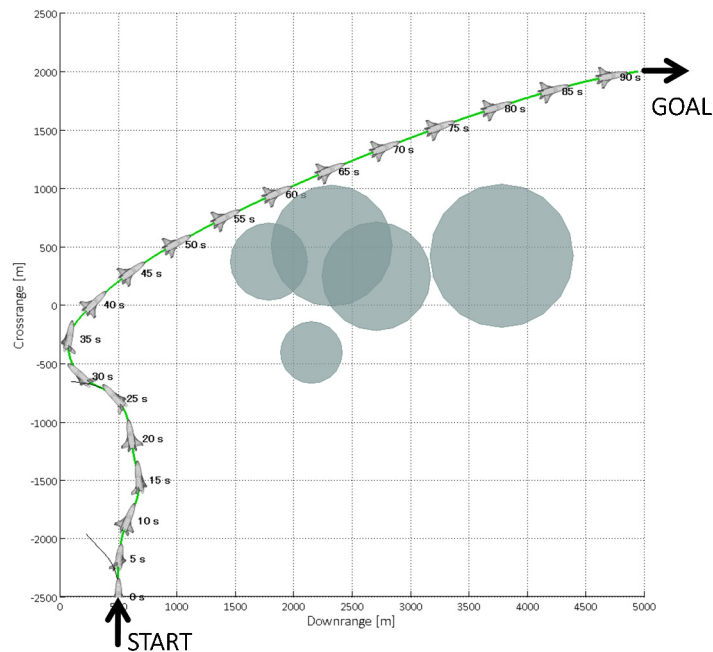


(a) Bird's-Eye View 1



(b) Bird's-Eye View 2

Fig. 4.2.10 Simulation Result (Obstacle Number 5)
(to be continued)



(c) Top View

Fig. 4.2.10 Simulation Result (Obstacle Number 5)
(to be concluded)

コマンド時系列データおよび、指令値と機体状態量のグラフを Fig. 4.2.11, Fig. 4.2.12 にそれぞれ示す. Fig. 4.2.11 の結果より, いずれのコマンドも制限値内で大きな振動のない実用的なコマンドが生成されていることが分かる. ベジエ曲線を用いた疑似軌道での飛行時 (約 40 秒以降) には, 迎角を低く取り経路角を下げて高度を急激に低下させているため, 機体の速度は目標値 (70m/s) を大きく超えて約 105 [m/s] まで増速している. しかし, 探索終了条件に速度誤差の評価項はなく, 条件を全て満たしている (距離誤差 46.9 [m], 方位角誤差 7.0 [°], 経路角誤差 9.5 [°]) ため解軌道となった.

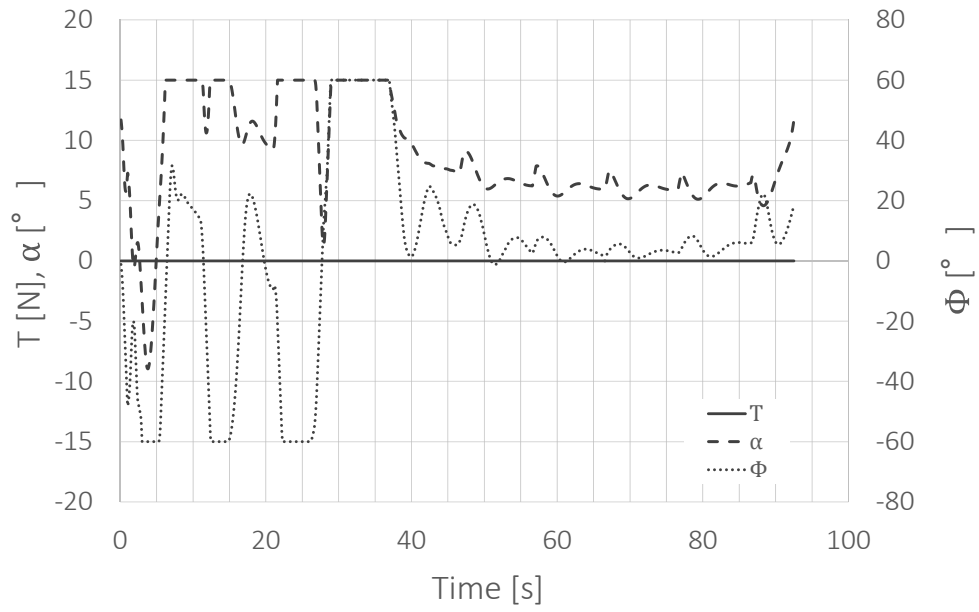


Fig. 4.2.11 Command Time Histories

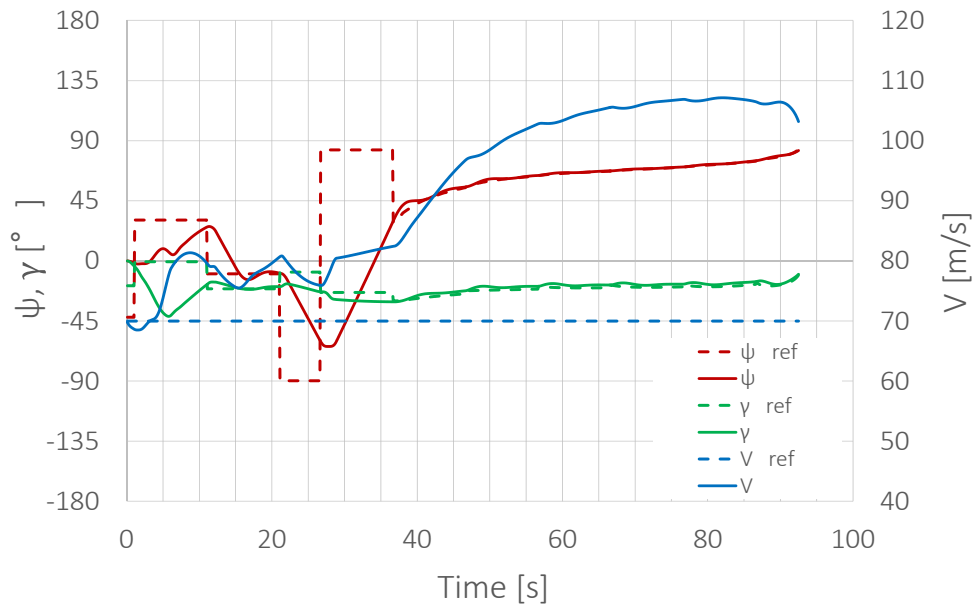
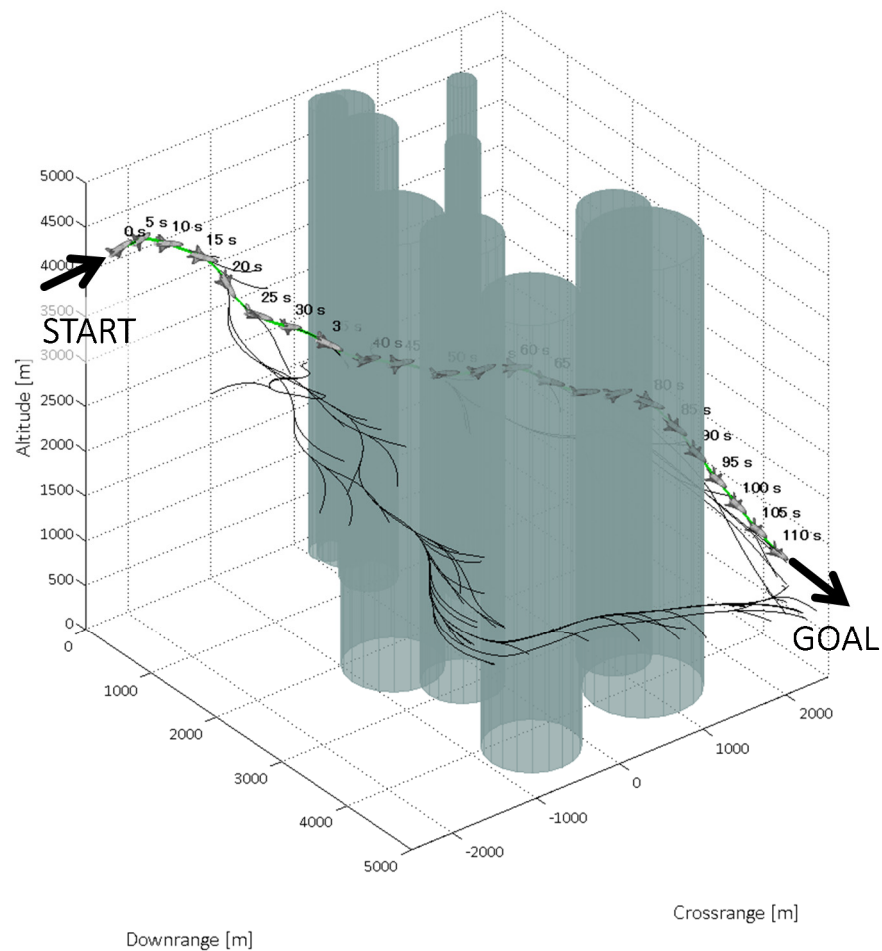


Fig. 4.2.12 Reference and Response State Variables

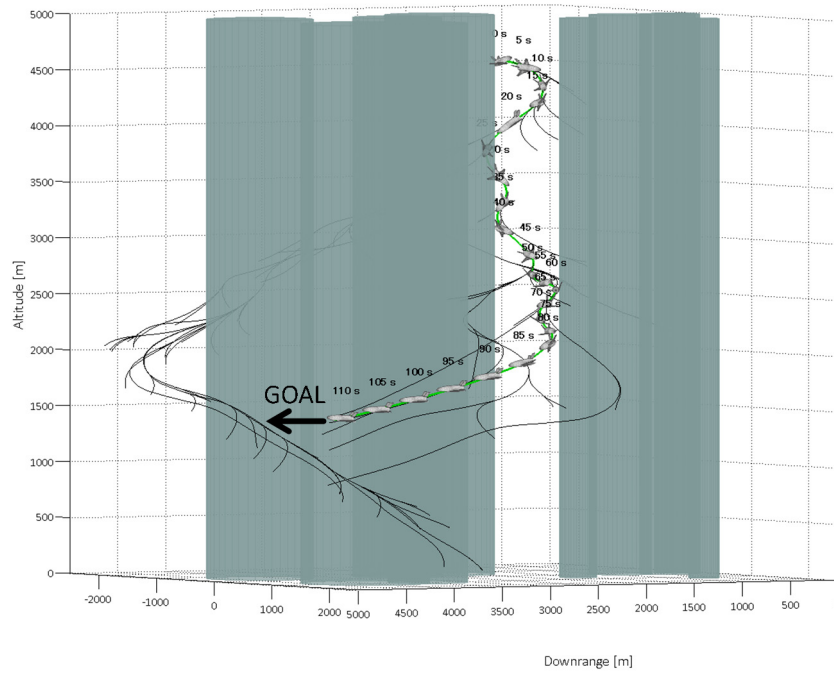
(2) 障害物 10 個

障害物の数を 10 個に増やして軌道生成を行った。結果を Fig. 4.2.13 に示す。機体進行方向に対して右側から迂回して目的地へ向かう探索が多く行われたが、いずれもゴールでの目標軌道角度を満たせずに失敗した軌道となった。一方で左側から迂回してくる軌道は、目標軌道角度を満たしやすい進入方向であるため、終了条件を満たして解軌道となることができた。このように障害物 10 個に増加しても良好な探索が行えた。なお、今回の有翼ロケットの探索シミュレーションは、初期方位を $0[^\circ]$ としているため火星探査航空機で生じた探索方位の偏りは現れにくいケースとなっている。

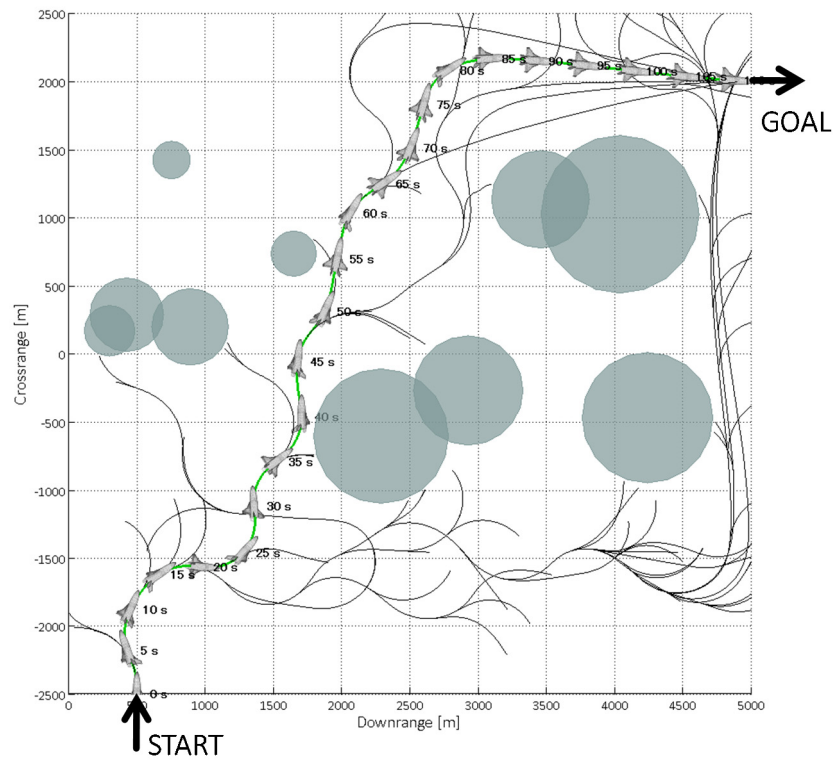


(a) Bird's-Eye View 1

Fig. 4.2.13 Simulation Result (Obstacle Number 10)
(to be continued)



(b) Bird's-Eye View 2



(c) Top View

Fig. 4.2.13 Simulation Result (Obstacle Number 10)
(to be concluded)

コマンド時系列データおよび、指令値と機体状態量のグラフを Fig. 4.2.14, Fig. 4.2.15 にそれぞれ示す。それぞれのコマンド値は制限内で良好に生成されているが、障害物 5 個の場合と比較して複雑な軌道であることから変動が激しい。方位角，経路角についてはランダムに与えられた値に追従する傾向が見られる。速度については、迎え角とバンク角の影響を受けて大きな増減が見られる。

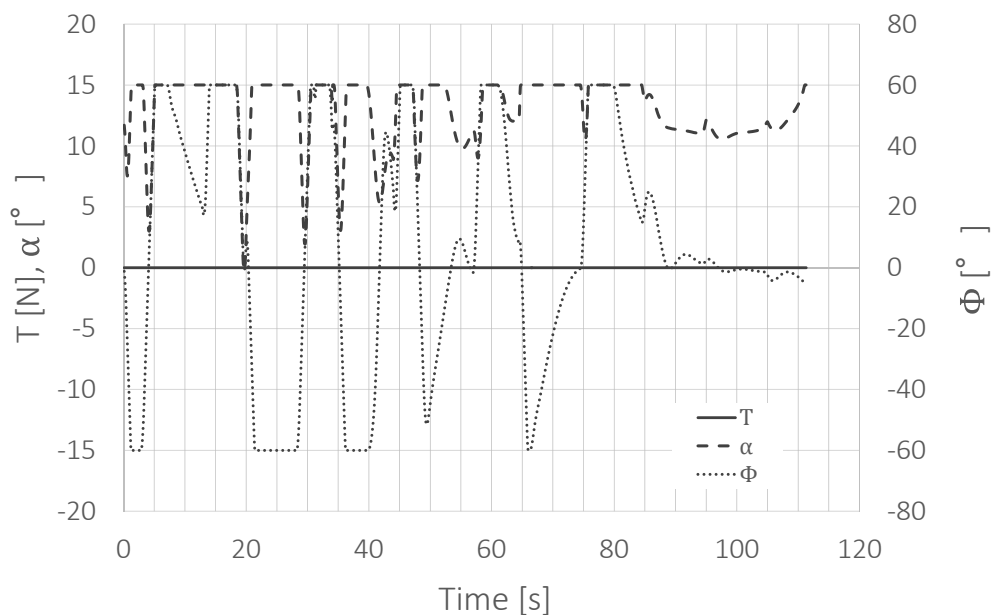


Fig. 4.2.14 Command Time Histories

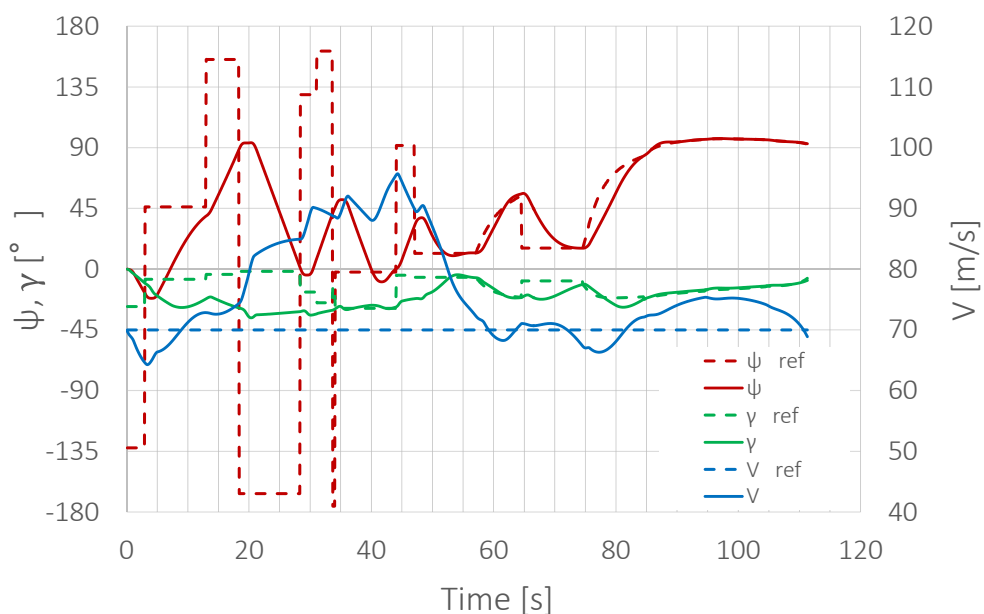
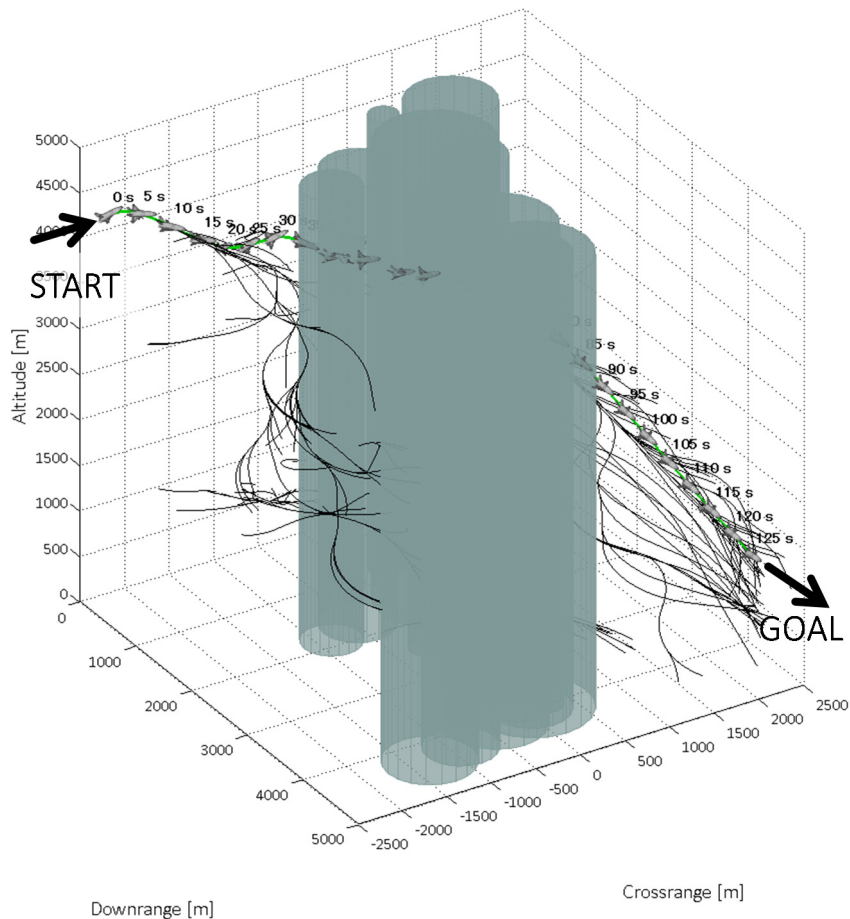


Fig. 4.2.15 Reference and Response State Variables

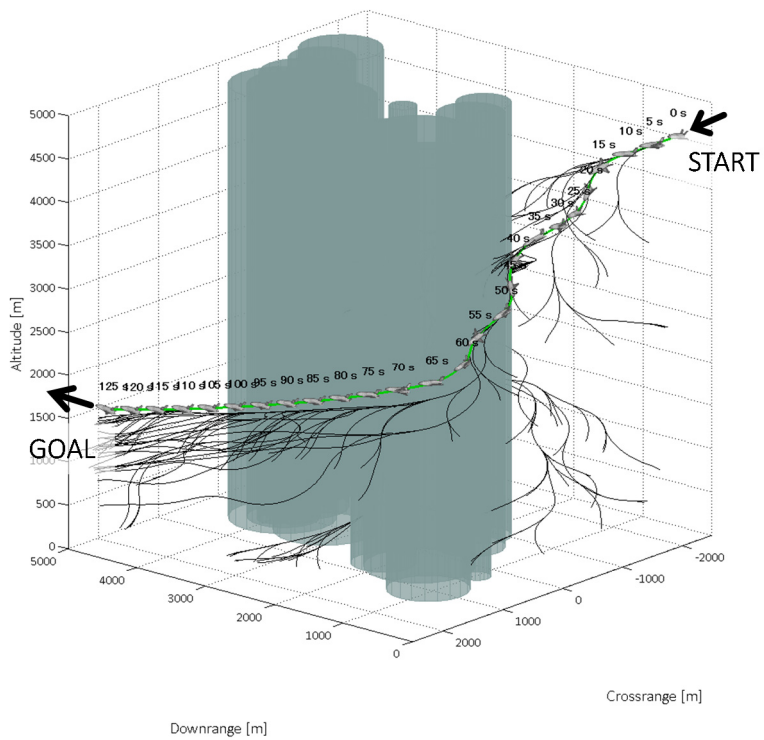
(3) 障害物 15 個

障害物の数を 15 個に増やして探索を行った．結果を Fig. 4.2.16 に示す．本環境では，通過できる領域が極めて狭いため多くの探索が試みられた．軌道が大回りとなり，長距離を飛行する必要があり多くの軌道が高度不足により到達できていない．また，滑空飛行を前提とした探索のためランダムに与える経路角範囲を $-30\sim0$ [°]としていたことも，高度を下げる方向のみに探索を進める要因となった．しかし，多くの探索を要したものの，結果として障害物 15 個に増加しても障害物を回避できる良好な軌道を探索することができた．

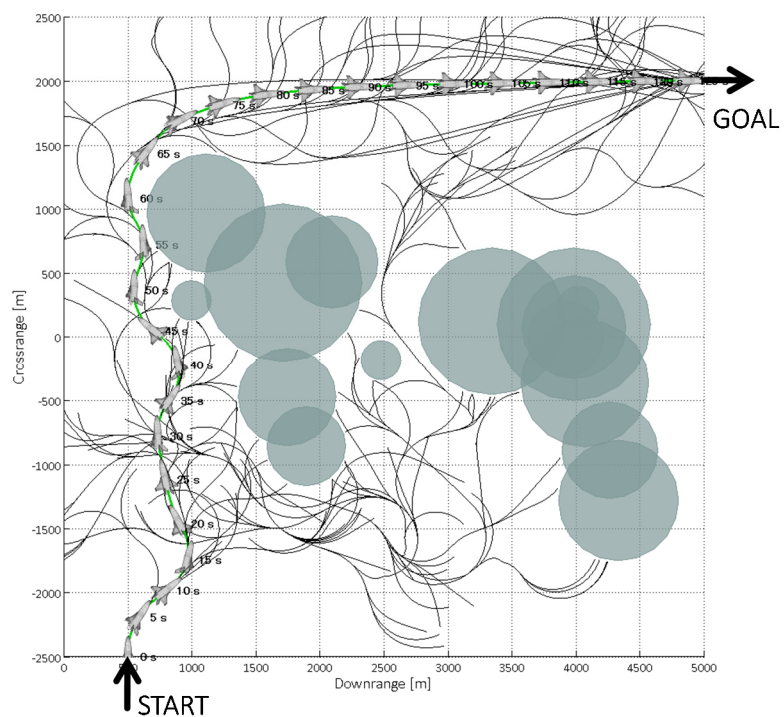


(a) Bird's-Eye View 1

Fig. 4.2.16 Simulation Result (Obstacle Number 15)
(to be continued)



(b) Bird's-Eye View 2



(c) Top View

Fig. 4.2.16 Simulation Result (Obstacle Number 15)
(to be concluded)

コマンド時系列データおよび、指令値と機体状態量のグラフを Fig. 4.2.17, Fig. 4.2.18 にそれぞれ示す。 Fig. 4.2.17 から分かるように、高度の低下を防ぐために迎角を常に大きく取る軌道が解軌道として生成されている。

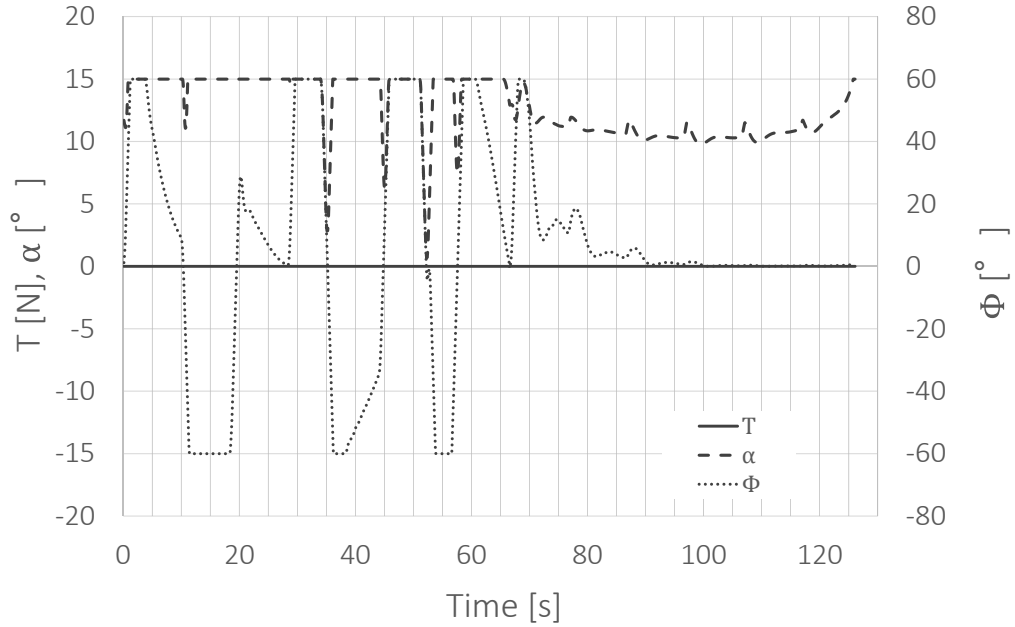


Fig. 4.2.17 Command Time Histories

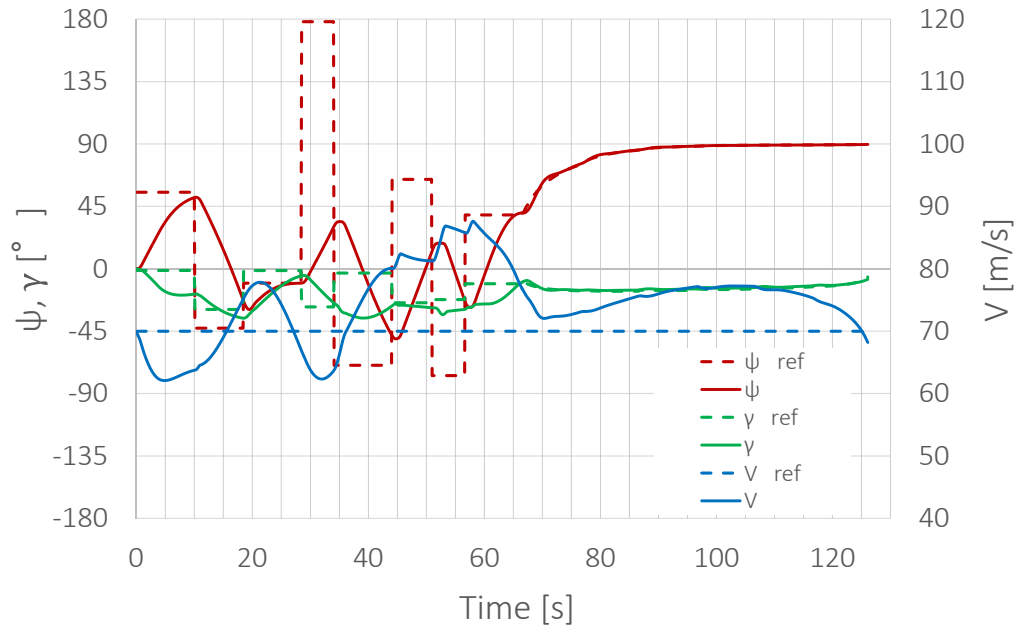


Fig. 4.2.18 Reference and Response State Variables

4.3 本章のまとめ

本章では、ランダム探索と力学フィルタによる融合型軌道プランナを提案した。シミュレーションの結果より、火星探査航空機および有翼ロケット実験機の両モデルにおいて、障害物が多数存在する複雑な環境においても良好な回避軌道を生成できることを示した。ただし、ランダム探索において目標方位角に偏りが生じることが分かった。この問題については、現在の機体方位角を基準方位として、そこからの差分でランダム値を与えるなどの工夫が必要である。

また、本手法は3章で取り上げたランダム探索と力学フィルタを組み合わせた基礎的な軌道プランナの課題を全て克服した。本章の最後に、基礎的な軌道プランナで列挙した課題に対する融合型軌道プランナの解決策について述べる。

(1) 力学フィルタ後の軌道が障害物に衝突する。

基礎的な軌道プランナでは、ランダム探索経路と力学フィルタの軌道に乖離があったため、障害物回避性を十分に保障することができなかった。一方、融合型軌道プランナはランダム探索と力学フィルタを同時に実行する。つまり、ダイナミクスを考慮しながらランダムに探索を進めていくため、確実に衝突しないツリーのみが成長していく。よって、衝突回避を保障する解軌道を得ることが可能となった。

(2) 力学フィルタの終了条件の不良により目的地まで軌道が描けない。

基礎的な軌道プランナでは、ランダム探索経路の距離と力学フィルタの距離を比較して、ランダム探索経路の総距離と等しい地点で軌道生成を終了していた。そのため、ランダム探索経路より大回りとなり飛行距離が伸びた軌道は目的地に到達する前に軌道計算が終了してしまう課題があった。一方、融合型軌道プランナでは、ベジエ曲線において指令角の更新時に飛行距離を用いているが、ベジエ曲線は一定時間（枝成長時間）毎に更新されるため生じた誤差はその都度オフセットされる。また、計算終了条件に目標位置との誤差を評価するため、一定距離以内に到達しない限り計算は終了しない。

(3) 目標 ID の更新方法の不良により誤誘導が生じる.

基礎的な軌道プランナでは、目標 ID の更新タイミングによって、バンク角が意図しない方向を指示し、誤誘導が生じる問題があった。これは目標経路の方位と機体の位置を考慮してバンクの向きを決定していたことが大きな要因であった。融合型軌道プランナでは、目標経路そのものが存在せず目標角度との誤差のみを用いているため、この問題は発生しない。なお、ベジエ曲線においても、疑似軌道から得られる角度を目標値として力学フィルタを解いているため同様に問題とならない。

(4) 機体の初期方位角に対して目標経路の初期方位角に不整合がある。

基礎的な軌道プランナでは、ランダム探索は機体の初期状態を考慮せずツリーを成長させるため、力学フィルタにより得られる軌道が目標経路と大きく乖離することが問題となった。しかし、融合型軌道プランナにおいては、初期方位角に基づきダイナミクスが計算され、また与えられる目標値も経路ではなく軌道角度であるため、初期方位角の不整合という問題は発生しない。

(5) ゴール地点の軌道進入方位が不定である。

基礎的な軌道プランナでは、軌道入射角を定めるアルゴリズムが組み込まれておらず、帰還軌道など入射角に制約があるケースに対応不可能であった。一方、融合型軌道プランナでは、ベジエ曲線を用いた疑似軌道という概念を導入することにより軌道入射角を定めることが可能となった。また、終了条件には角度誤差の評価を行うため、規定の誤差範囲に収まった軌道のみを解軌道として見なすことができる。

以上より、3 章で挙げた課題は融合型軌道プランナにおいて全て解決された。ただし、融合型軌道プランナを備えた複数の機体が互いを避けて飛行する場合を想定すると、互いの機体がランダムに飛行することにより衝突が発生する可能性を排除できないなどの課題が残されている。

第5章 リアルタイム誘導飛行シミュレーション

5.1 はじめに

本章では、融合型軌道プランナを用いたリアルタイム誘導の飛行シミュレーションを行う。自律誘導システムでは、アルゴリズムに用いたモデルと実際の運動に誤差がある場合や、突風などの外乱の影響で目標軌道から外れたとき、または障害物環境が時々刻々と変化する場合などに、その都度軌道修正を行うリアルタイム誘導性能が重要となる。

本シミュレーションでは、機体の運動を並進運動と回転運動の6自由度運動方程式として厳密なモデルで扱い、軌道プランナで生成される指令値に従い飛行させる。そこで、軌道プランナで得られる目標軌道を全く更新しない場合と、随時更新する場合とで結果を比較評価し、リアルタイム誘導の必要性和有効性を検証する。ただし、リアルタイム誘導を実現する上で重要な課題の一つである軌道の計算時間に関しては無限に短い時間に終わると仮定する。つまり、軌道プランナにより目標軌道を更新している際に、機体は一時静止しているものとみなす。また、本シミュレーションは長距離誘導探査を目的とする火星探査航空機と、ロケットの軌道頂点からの帰還誘導飛行を行う有翼ロケット実験機の両ケースで行う。

5.26 自由度運動方程式と制御則

本節では，飛行シミュレーションに用いる 6 自由度運動方程式と制御則を記載する．各空力係数および空力微係数は火星探査航空機，有翼ロケット（実際は HIMES の模型）それぞれの風洞試験で得られた結果を用いる．

(1) 記号の定義

m	: 質量	θ	: ピッチ角
g	: 重力	u, v, w	: 軸方向の速度
ρ	: 大気密度	p, q, r	: 軸周りの角速度
S	: 翼面積	I_{xx}, I_{yy}, I_{zz}	: 慣性モーメント
\bar{b}	: 主翼スパン	F_{Ax}, F_{Ay}, F_{Az}	: 軸方向の空気力
\bar{c}	: 平均空力翼弦	L, M, N	: 軸周りのモーメント
V	: 対気速度	$\delta_a, \delta_e, \delta_r$: エルロン，エレベータ， ラダー
T	: 推力	C_L, C_D	: 揚力，抗力係数
α	: 迎角	C_x, C_y, C_z	: 軸方向の空力係数
β	: 横滑り角	C_l, C_m, C_n	: 軸周りのモーメント係数
ϕ	: バンク角		

(2) 並進運動方程式

$$\dot{u} = -g \sin \theta + \frac{F_{Ax}}{m} + \frac{T}{m} - qw + rv \quad (5.1)$$

$$\dot{v} = g \cos \theta \sin \phi + \frac{F_{Ay}}{m} - ru + pw \quad (5.2)$$

$$\dot{w} = g \cos \theta \cos \phi + \frac{F_{Az}}{m} - pv + qu \quad (5.3)$$

(3) 各軸方向に働く空気力

$$F_{Ax} = \frac{1}{2} \rho V^2 S (-C_d \cos \alpha + C_l \sin \alpha + C_{x\delta_e} \delta_e) \quad (5.4)$$

$$F_{Ay} = \frac{1}{2} \rho V^2 S (C_{y\beta} \beta + C_{y\delta_a} \delta_a + C_{y\delta_r} \delta_r) \quad (5.5)$$

$$F_{Az} = \frac{1}{2} \rho V^2 S (-C_L \cos \alpha + C_D \sin \alpha + C_{z\delta_e} \delta_e) \quad (5.6)$$

(4) 回転運動方程式

$$I_{xx} \dot{p} + (I_{zz} - I_{yy}) qr = L \quad (5.7)$$

$$I_{yy} \dot{q} + (I_{xx} - I_{zz}) pr = M \quad (5.8)$$

$$I_{zz} \dot{r} + (I_{yy} - I_{xx}) pq = N \quad (5.9)$$

(5) 各軸周りのモーメント

$$L = \frac{1}{2} \rho V^2 S \bar{b} \left\{ C_{l\beta} \beta + C_{l\delta_a} \delta_a + C_{l\delta_r} \delta_r + \frac{\bar{b}}{2V} (C_{lp} p + C_{lr} r) \right\} \quad (5.10)$$

$$M = \frac{1}{2} \rho V^2 S \bar{c} \left\{ C_m + C_{m\delta_e} \delta_e + C_{m\beta} \beta + \frac{\bar{c}}{2V} (C_{mq} q + C_{m\dot{\alpha}} \dot{\alpha}) \right\} \quad (5.11)$$

$$N = \frac{1}{2} \rho V^2 S \bar{b} \left\{ C_{n\beta} \beta + C_{n\delta_a} \delta_a + C_{n\delta_r} \delta_r + \frac{\bar{b}}{2V} (C_{np} p + C_{nr} r) \right\} \quad (5.12)$$

次に、6 自由度シミュレーションの制御則について述べる。機体は推力および 3 つの空力舵面（エルロン、エレベータ、ラダー）の計 4 つの入力によって制御される。軌道プランナによって得られる推力、バンク角、迎角の時系列コマンドを指令値として用いる。まず、推力には指令値を直接入力する。エルロンおよびエレベータの操舵角は、それぞれバンク角と迎角の誤差により、ラダーの操舵角は、横滑り角を 0 にするよう PD 制御系を構成する。これらの制御則を以下に示す。ただし、エレベータに関しては姿勢を安定させるため、風洞試験で得られたトリム舵角を加えている。

$$\begin{bmatrix} T \\ \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} = \begin{bmatrix} T_{com} \\ K_{ap}\phi_{err} + K_{ad}\dot{\phi}_{err} \\ \delta_{e_{trim}} + K_{ep}\alpha_{err} + K_{ed}\dot{\alpha}_{err} \\ K_{rp}\beta_{err} + K_{rd}\dot{\beta}_{err} \end{bmatrix} \quad (5.13)$$

$$\begin{bmatrix} \phi_{err} \\ \alpha_{err} \\ \beta_{err} \end{bmatrix} = \begin{bmatrix} \phi_{com} - \phi \\ \alpha_{com} - \alpha \\ 0 - \beta \end{bmatrix} \quad (5.14)$$

5.3 誘導飛行シミュレーション

5.3.1 火星探査航空機

5.3.1.1 シミュレーション条件

本シミュレーションでは、プロジェクトで実際に想定され得るミッションを模擬して探索を行う。探索場所は候補地の一つとして検討される火星のマリネリス峡谷の一部とし、ダウンレンジ 100km の飛行探査を行う。探索域の地図（Google Earth の火星地図より引用）および、そのシミュレーションモデルを Fig. 5.3.1 と Fig. 5.3.2 にそれぞれ示す。探索域はダウンレンジ 100km、クロスレンジは 20km とする（太枠）。

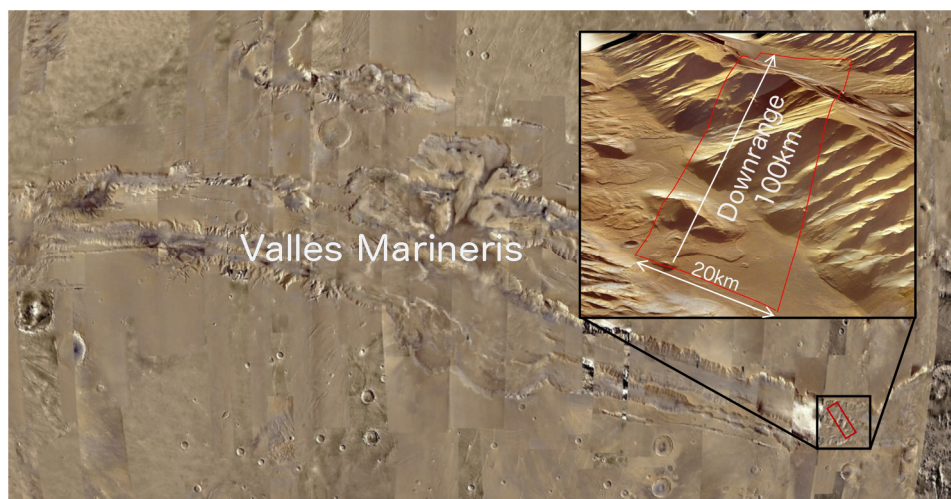


Fig. 5.3.1 Valles Marineris Image

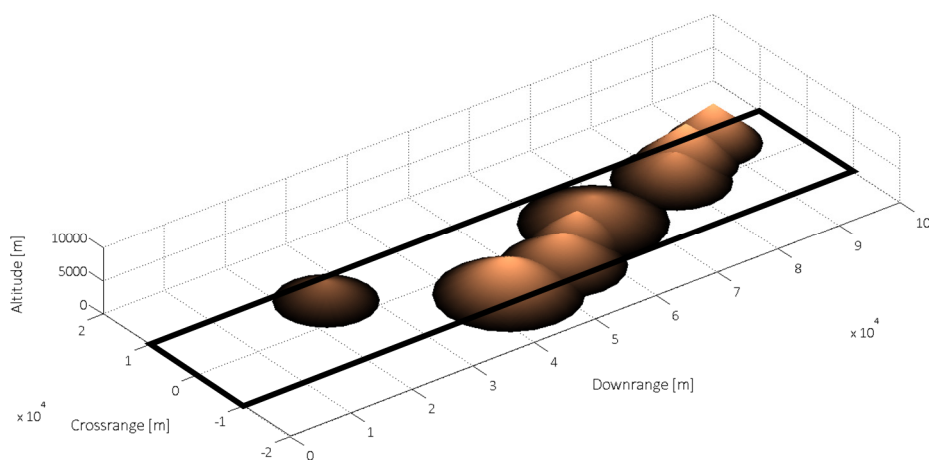


Fig. 5.3.2 Terrain Map Model

また，初期位置と初期軌道角度，および目標位置と目標軌道角度を以下に示す．その他機体の初期パラメータは前章と同じとする．

Table 5.3.1 Initial and Terminal Trajectory Conditions

初期位置 $[\xi, \eta, h]$	[m]	[0, 0, 3000]
初期軌道角度 $[\psi, \gamma]$	[°]	[90, 0] (※特別に Crossrange 正方向を 0 とし，火星の磁北が 0 でない点に留意)
目標位置 $[\xi, \eta, h]$	[m]	[100000, 0, 1000]
目標軌道角度 $[\psi, \gamma]$	[°]	[90, 0] (※特別に Crossrange 正方向を 0 とし，火星の磁北が 0 でない点に留意)

ここで，軌道プランナの各パラメータは前章と同じとするが，バンク角と迎角のコマンド制限値のみ，それぞれ ± 20 [°]（前章は ± 30 [°]）と ± 6 [°]（前章は ± 7 [°]）に変更する．これは 100km という長距離飛行を想定するにあたり，急旋回を行わなくても障害物回避は十分に可能であり，安定した姿勢で飛行を続けるためである．さらに，6 自由度シミュレーションの制御ゲインとトリム舵角を Table 5.3.2 に示す．

Table 5.3.2 Control Gains and Trim Angles for 6DOF Simulation

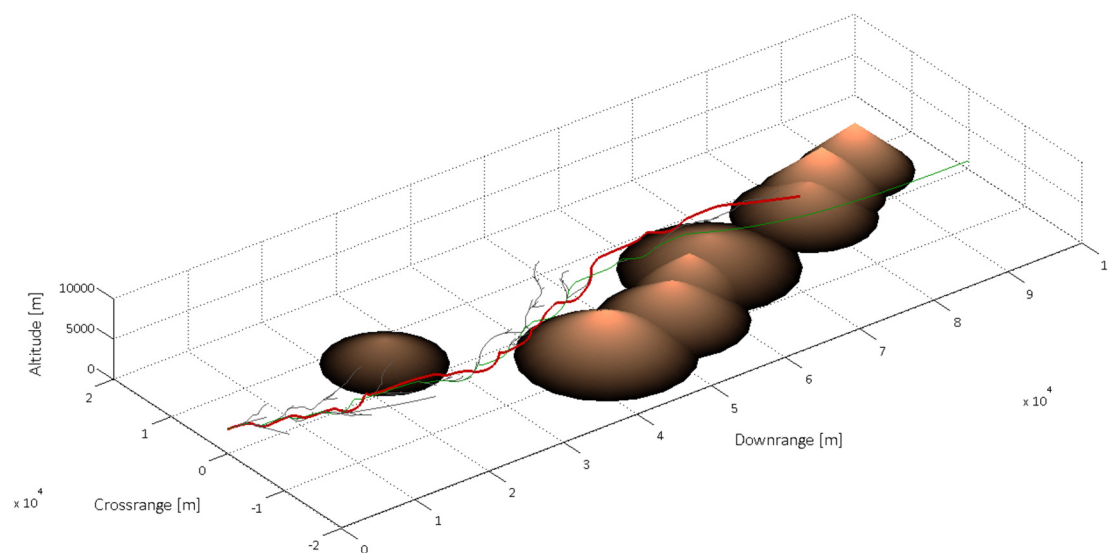
K_{ap}	-0.5	K_{ad}	0.2
K_{ep}	-15	K_{ed}	0.8
K_{rp}	-0.5	K_{rd}	0.5
$\delta_{e_{trim}} = 4.3602\alpha_{com}^2 - 3.1514\alpha_{com} - 0.0332$ [rad]			

5.3.1.2 シミュレーション結果

本節では, 飛行シミュレーションにより軌道プランナによるリアルタイム誘導の有効性を検証する. そこで, 軌道を最初の一回のみ探索した結果 (軌道を更新しない) を用いて飛行した場合と, 軌道を一定時間毎にリアルタイム更新した誘導飛行の 2 通りの結果を比較する.

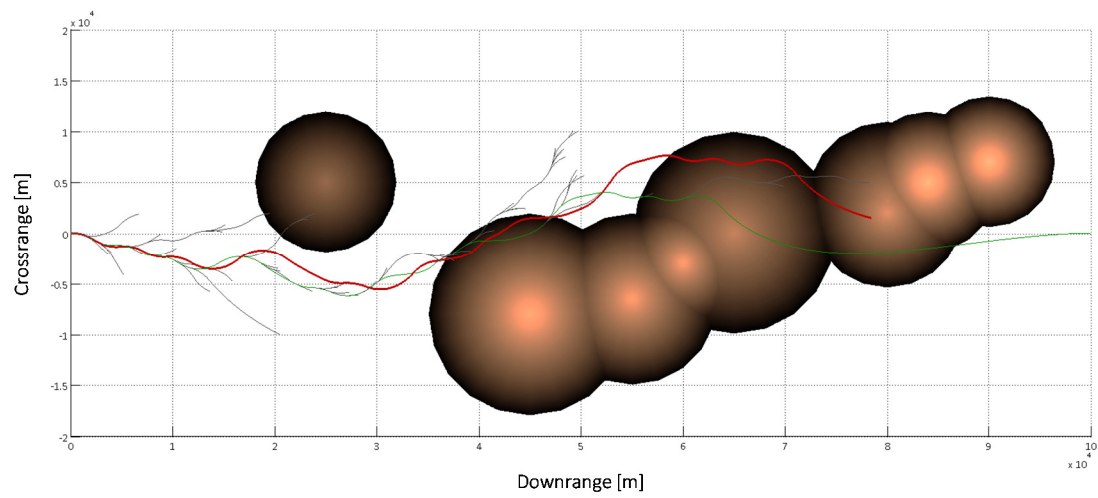
(1) 軌道を更新しない場合

スタート位置で一度のみ軌道プランナによる軌道生成を行い, そこから与えられる指令値の時系列データを用いて 6 自由度シミュレーションを実行した. シミュレーションにより得られた軌道プランナのツリーおよび, 飛行軌跡を Fig. 5.3.3 に示す. 図において, 黒色実線が軌道プランナにより探索された全てのツリーであり, 緑色実線は軌道プランナの解軌道を表す. また, 赤色実線は, 誘導飛行した機体の軌跡である. 結果より, 力学フィルタの 3 自由度モデルと本シミュレーションの 6 自由度モデルの差異や, 制御設計の不完全性により飛行軌跡が徐々に目標軌道から離れていく様子が分かる. そして機体が山に衝突 (Fig. 5.3.4) し, 火星探査航空機の 100km の探査飛行は失敗する結果となった.

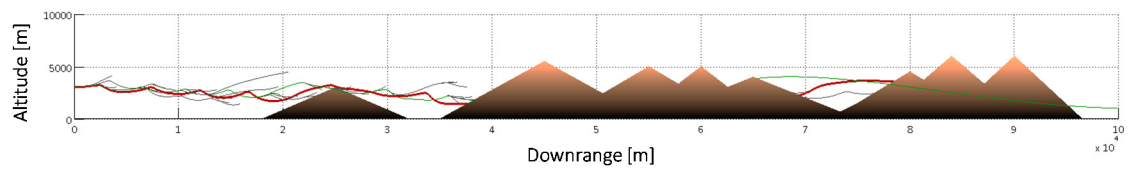


(a) Bird's-Eye View

Fig. 5.3.3 6 DOF Simulation Result without Trajectory Update
(to be continued)



(b) Top View



(c) Side View

Fig. 5.3.3 6 DOF Simulation Result without Trajectory Update
(to be concluded)

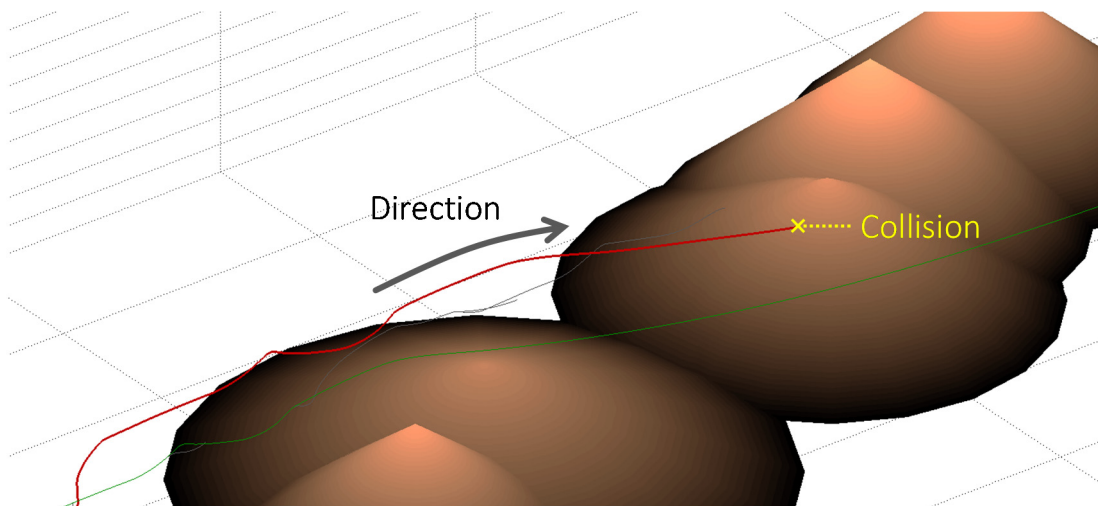
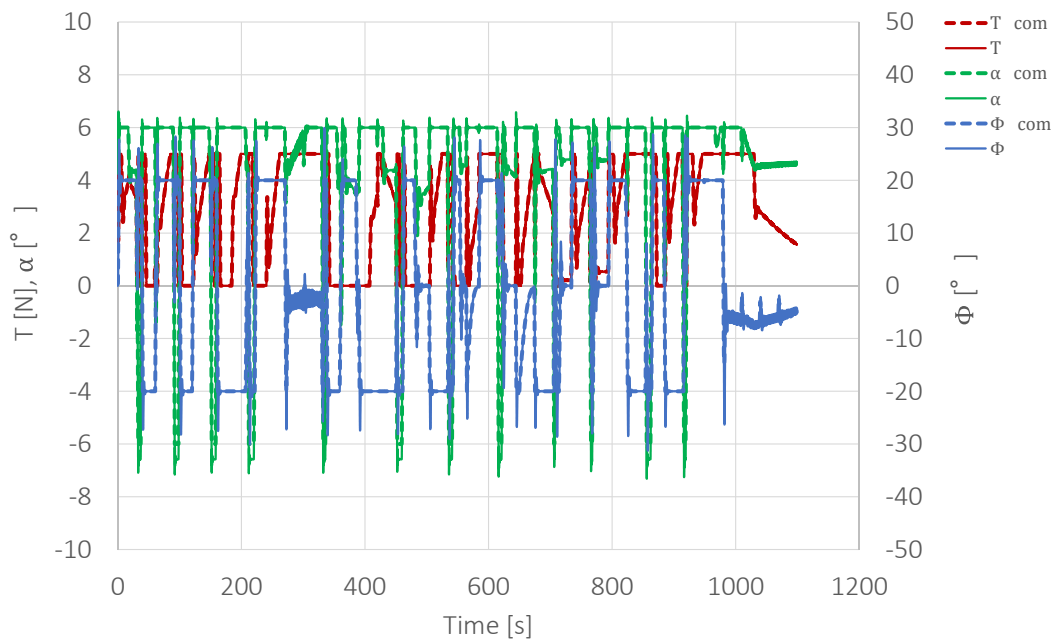


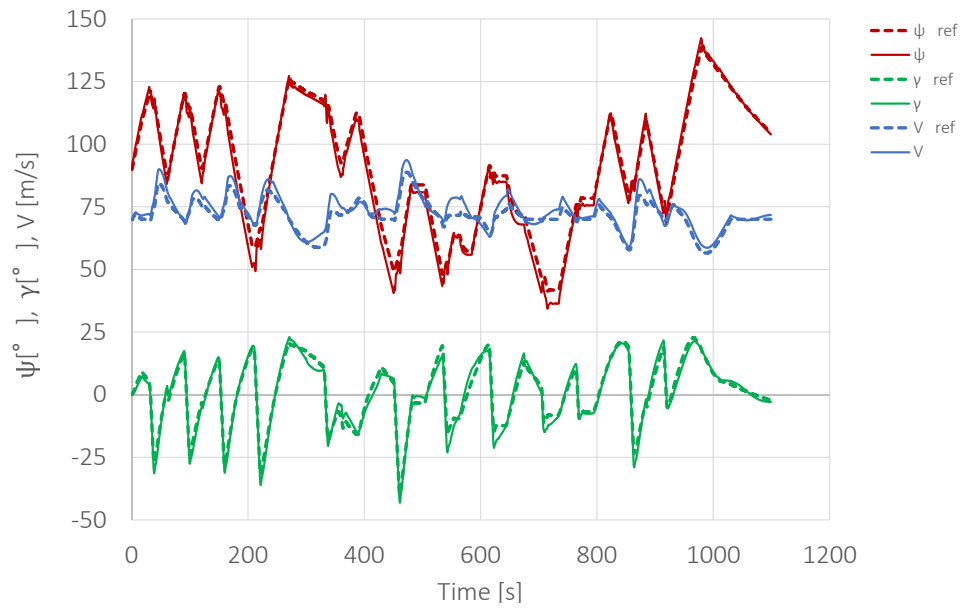
Fig. 5.3.4 Collision with Obstacle

次に、軌道プランナが与えた各指令値（破線）と実際の機体の挙動（実線）を比較したグラフを Fig. 5.3.5(a)に示す。推力は指令値をそのまま入力しているので完全に一致している。迎角は概ね良く追従しているが、低迎角時（ $\alpha_{com} = -6 [^\circ]$ ）に約 $0.5 [^\circ]$ の偏差が残っている。これは、トリム舵角をモデル化する際に二次曲線近似を用いていることによる誤差が影響していると考えられる。バンク角は小さな振動が見られる箇所がある（約 300 秒付近）が、概ね良好に追従している。さらに、方位角，経路角，速度に関して、軌道プランナの計算結果（添字 ref で表す）と機体の挙動を比較した結果を Fig. 5.3.5(b)に示す。いずれの結果も軌道プランナの値に概ね良好に追従しているが、多少の誤差を含んでいる。時々刻々の誤差は小さいが、軌道を計算する際には、これらの誤差が積分されて影響するため、次第に機体は目標軌道から外れていく。その誤差により障害物との衝突が発生する。



(a) Thrust, Angle of Attack and Bank Angle

Fig. 5.3.5 Reference and Response State Variables without Trajectory Update
(to be continued)



(b) Direction Angle , Flight Path Angle and Velocity

Fig. 5.3.5 Reference and Response State Variables without Trajectory Update
(to be concluded)

この他に、得られた舵角および機体の角速度を以下に示す。バンク角のコマンドが振動的なとき、同様にエルロン δ_a およびロールレート p の振動が見られる（300 秒付近）が、その後は概ね安定して飛行していることが分かる。

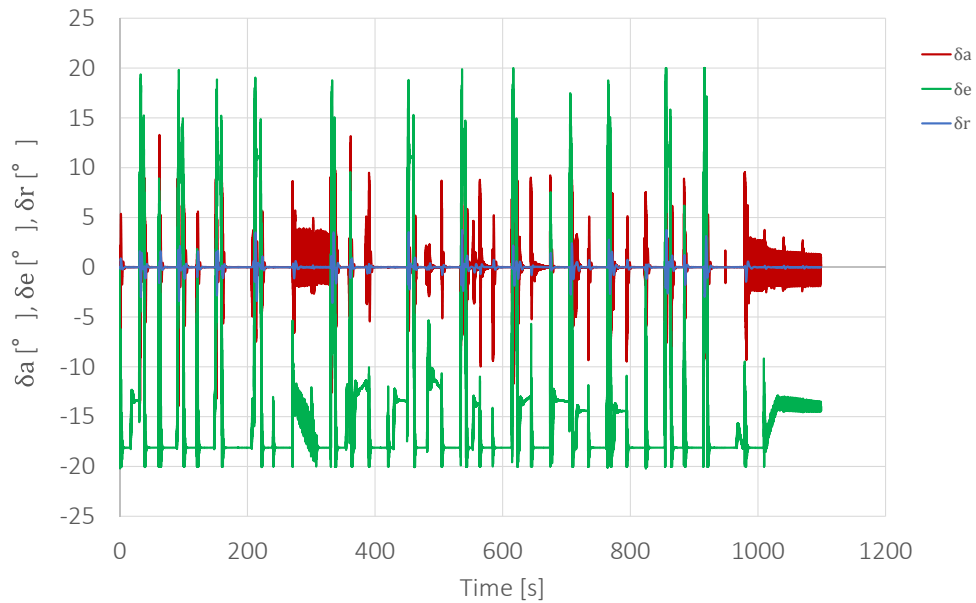


Fig. 5.3.6 Control Surface Angle Profile without Trajectory Update

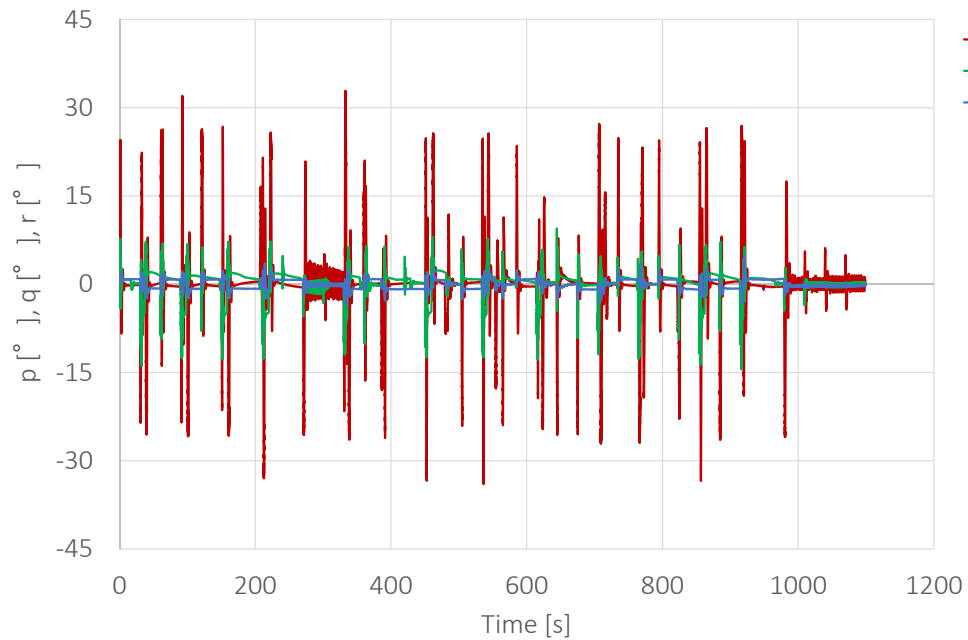
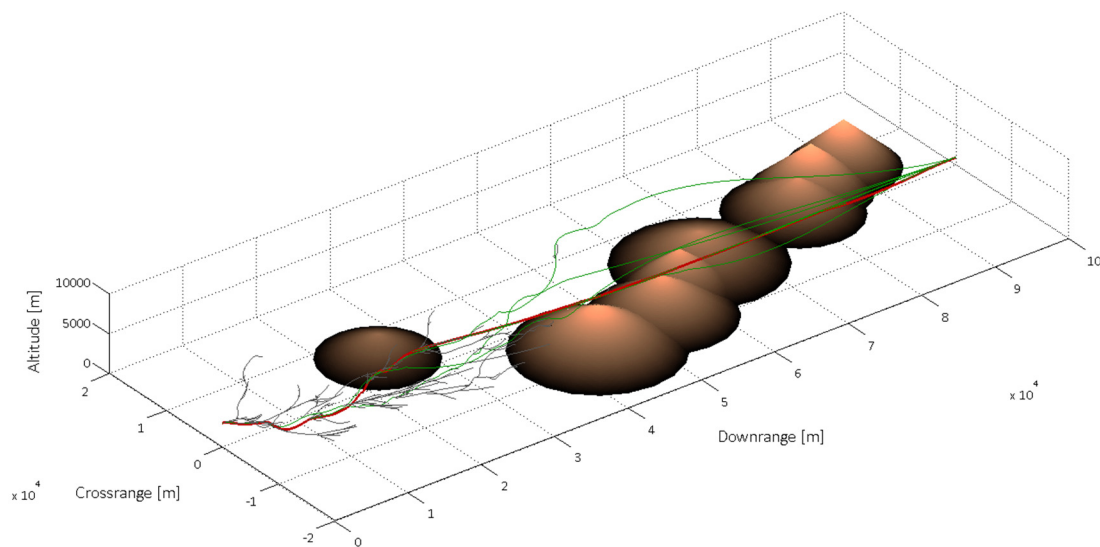


Fig. 5.3.7 Angular Velocity Profile without Trajectory Update

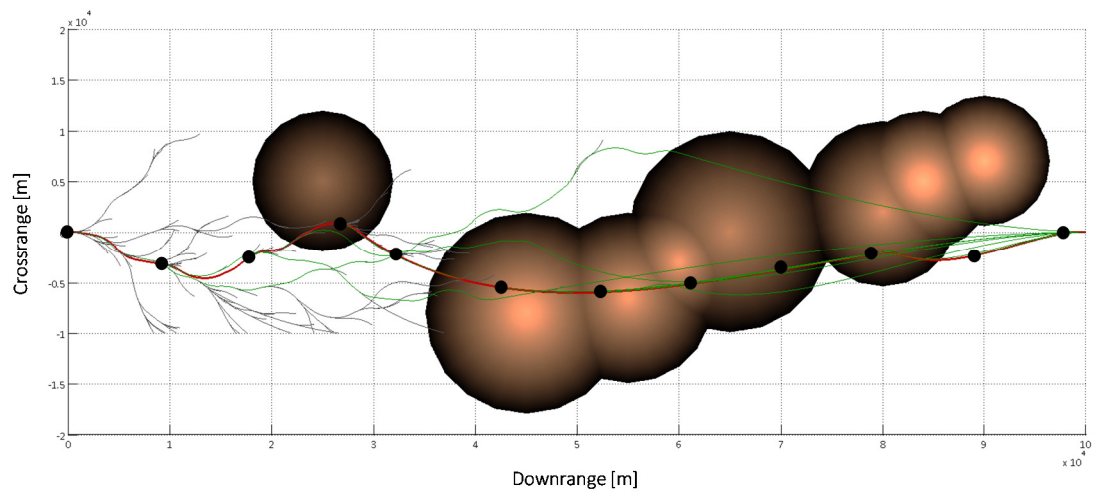
(2) 軌道を更新する場合

次に、軌道プランナを用いて 120 秒毎に軌道を更新した。シミュレーションの結果を Fig. 5.3.8 に示す。複数本の緑色実線は、軌道更新した結果を全て表示しており、黒色実線の探索ツリーも全て表示している。また Fig. 5.3.8(b)および(c)では、軌道更新がなされた地点を黒色丸プロットで示しており、合計 12 回の軌道生成が行われたことが分かる。シミュレーションの結果より、軌道修正をすることでゴールに到達して 100km の探査ミッションが達成できる誘導飛行が行えることが明らかとなった。さらに Fig. 5.3.9 では、誘導飛行の終盤において、山と機体のニアミスが発生した箇所の拡大図を示している。ここでは、軌道プランナの目標軌道に対して誤差が生じて機体は僅かに高度が下がっており、山肌に近づいている。ここで軌道の更新が行われて（図内黒色丸プロット箇所）衝突が回避された様子が分かる。

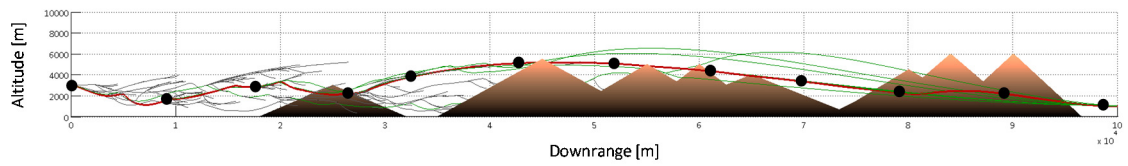


(a) Bird's-Eye View

Fig. 5.3.8 6 DOF Simulation Result with Trajectory Update
(to be continued)



(b) Top View



(c) Side View

Fig. 5.3.8 6 DOF Simulation Result with Trajectory Update
(to be concluded)

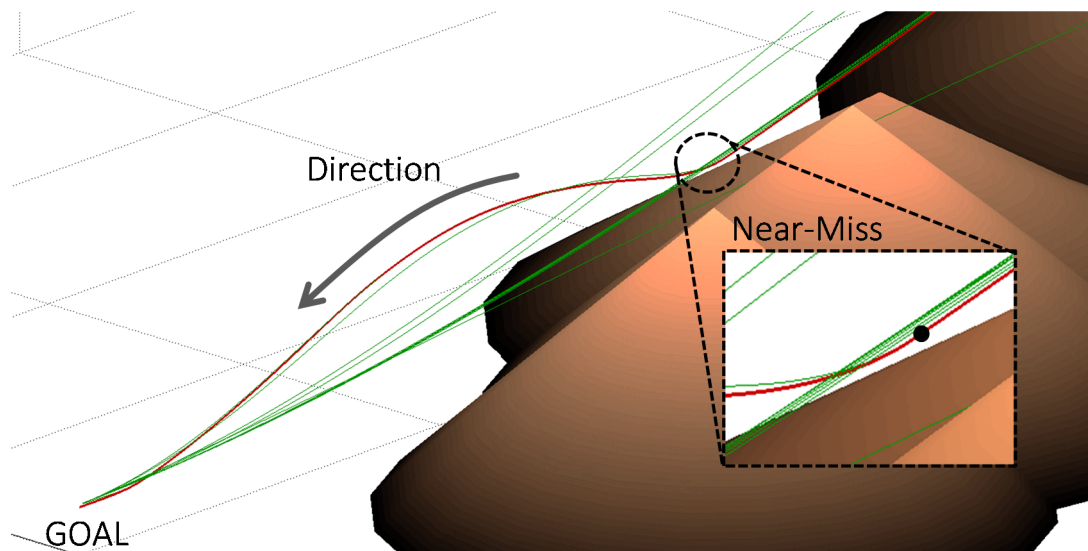
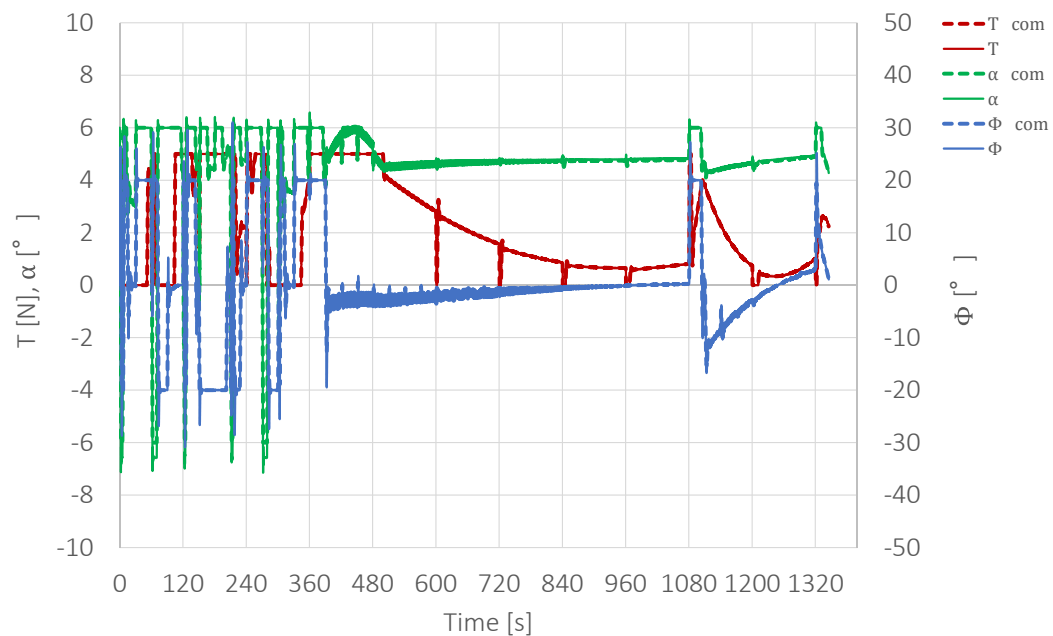


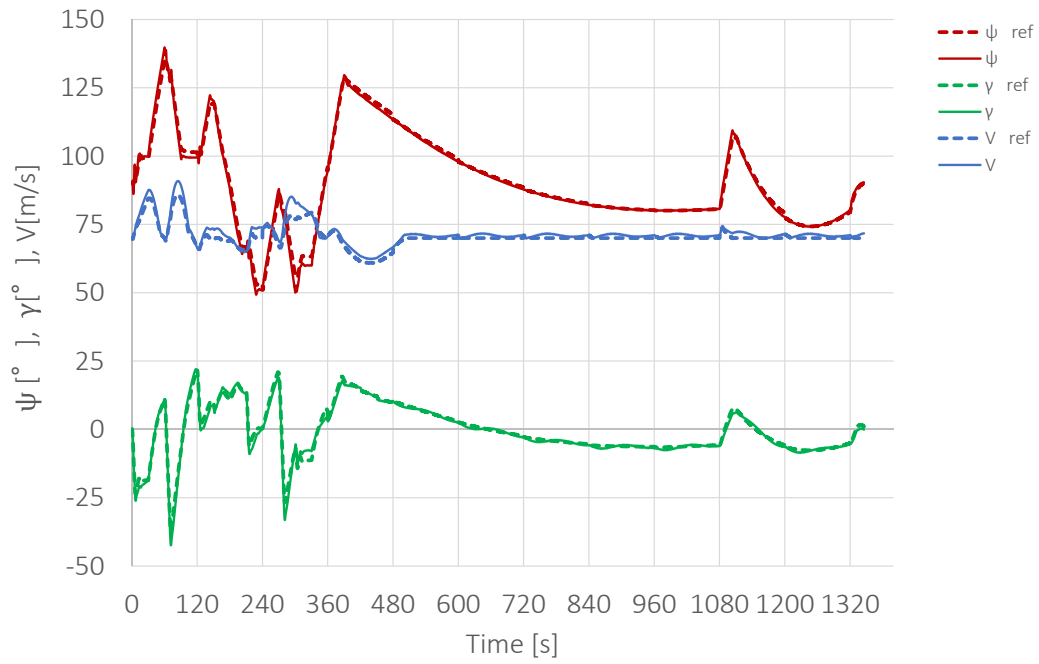
Fig. 5.3.9 Near-Miss Flight Situation

軌道プランナが与えた指令値と実際の機体の挙動を比較したグラフを Fig. 5.3.10(a) に示す. 120 秒毎の軌道更新時に, 軌道修正のため指令値が変動している. また, 方位角, 経路角, 速度に関して, 軌道プランナの計算結果 (添字 **ref** で表す) と機体の挙動を比較した結果を Fig. 5.3.10(b) に示す. 軌道更新が行われても, 軌道は滑らかに接続されているため, 急激な変動が見られることなく軌道の目標値に対して高い追従性を示した. なお, 1080 秒以降の大きな変化は, Fig. 5.3.9 に示したニアミスの地点における軌道更新を表しており, 再びランダム探索が行われている.



(a) Thrust, Angle of Attack and Bank Angle

Fig. 5.3.10 Reference and Response State Variables with Trajectory Update
(to be continued)



(b) Direction Angle , Flight Path Angle and Velocity

Fig. 5.3.10 Reference and Response State Variables with Trajectory Update
(to be concluded)

この他に、得られた舵角および機体の角速度を以下に示す。迎え角とバンク角のコマンドが振動的なとき（380 秒以降）、6 自由度シミュレーションのエレベータ δ_e 、エルロン δ_a が反応し、その結果、機体の迎え角とバンク角およびピッチレート q 、ロールレート p も振動している。また、120 秒毎の軌道更新時にコマンドが変動するのに対応してそれぞれの値が僅かに変動している。

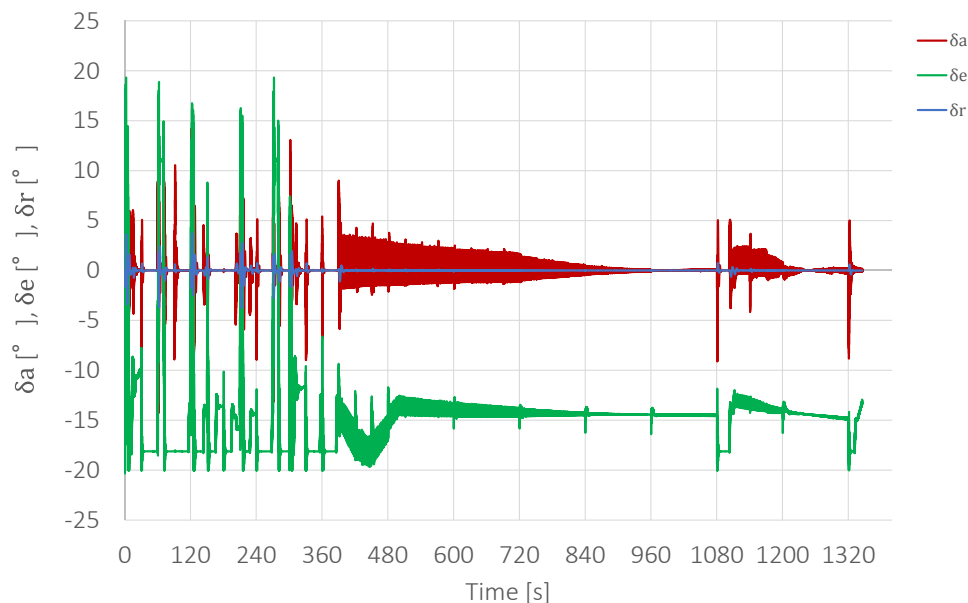


Fig. 5.3.11 Control Surface Angle Profile with Trajectory Update

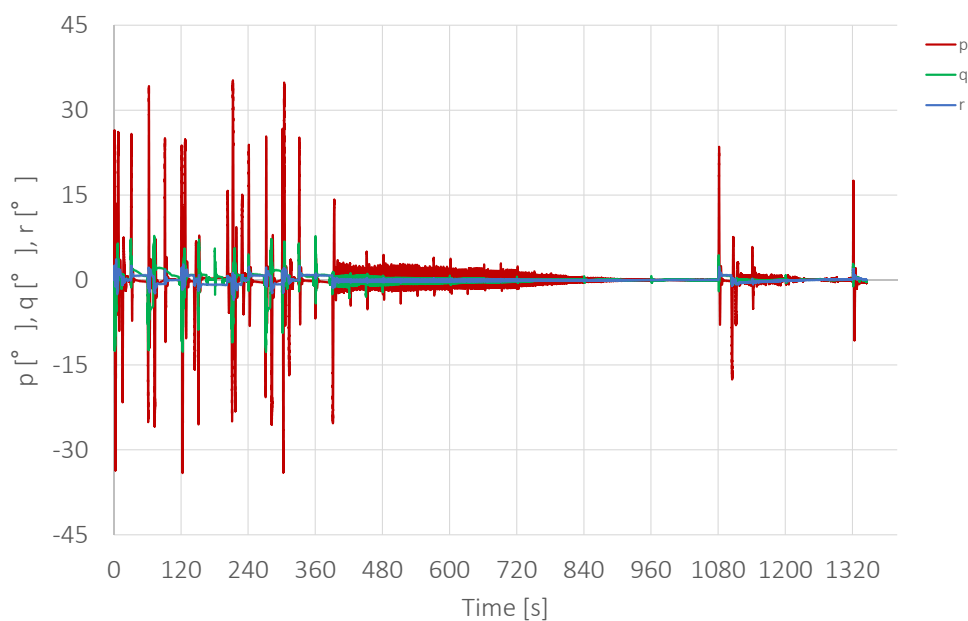


Fig. 5.3.12 Angular Velocity with Trajectory Update

5.3.2 有翼ロケット実験機

5.3.2.1 シミュレーション条件

有翼ロケット実験機 **WIRES#015** の打上げ実験候補地の一つとして検討されているアメリカ，テキサス州ヴァンホーンをシミュレーション場所として想定する．探索域の模式図を Fig. 5.3.13 に示す（地図は Google Earth より引用）．ヴァンホーン近郊に存在する施設の上空を飛行禁止区域（黒丸）とし，軌道頂点からの帰還飛行を行う．有翼ロケットは軌道頂点から滑空飛行を行い，目的地上空でパラシュートを展開して軟着陸する．図内の黄色の丸プロットは軌道頂点 (Apogee) とパラシュート展開地点 (Landing Site) を表し，それらの定める方位角を矢印で示している．

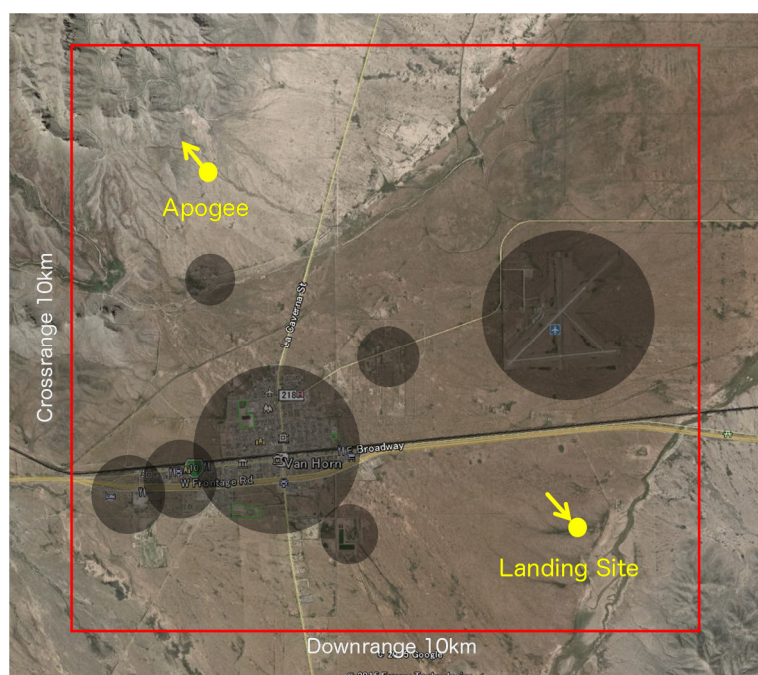


Fig. 5.3.13 Terrain Map at Van Horn / Texas

スタート (Apogee) およびゴール (Landing Site) の位置と軌道角度を下表のように定める．WIRES#015 の本シミュレーションでは最高到達高度 6[km]とし，高度 2[km]でパラシュートの展開を想定する．また，初期速度を 80[m/s]とし，その他機体の初期パラメータおよび軌道プランナ的设计パラメータは前章と同じとする．

Table 5.3.3 Initial and Terminal Trajectory Conditions

スタート位置 $[\xi, \eta, h]$	[m]	[2000, 3500, 6000]
スタート軌道角度 $[\psi, \gamma]$	[°]	[-45, 0]
ゴール位置 $[\xi, \eta, h]$	[m]	[8500, -3500, 2000]
ゴール軌道角度 $[\psi, \gamma]$	[°]	[135, -10]

ここで、6 自由度シミュレーションの制御ゲインとトリム舵角を Table 5.3.4 に示す.

Table 5.3.4 Control Gains and Trim Angles for 6 DOF Simulation

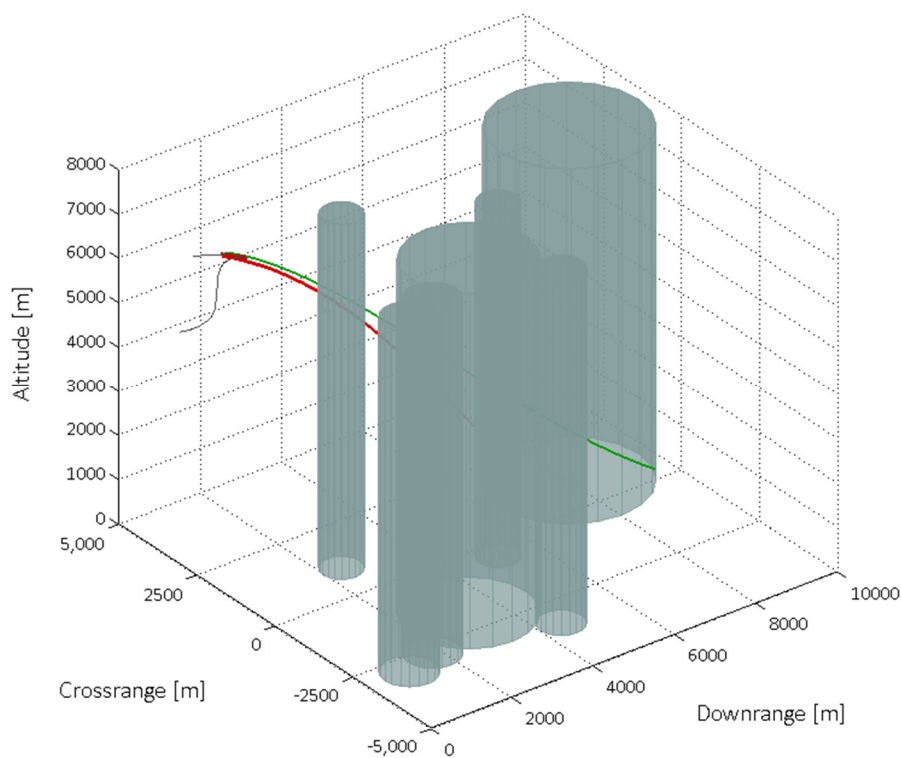
K_{ap}	0.4	K_{ad}	-0.01
K_{ep}	-1	K_{ed}	0.1
K_{rp}	1	K_{rd}	-0.1
$\delta_{e_{trim}} = -0.3998\alpha_{com}^2 - 0.7481\alpha_{com} + 0.0791 [rad]$			

5.3.2.2 シミュレーション結果

火星探査航空機同様に軌道を更新しない場合とリアルタイム更新する場合の 2 ケースでシミュレーションを行った。

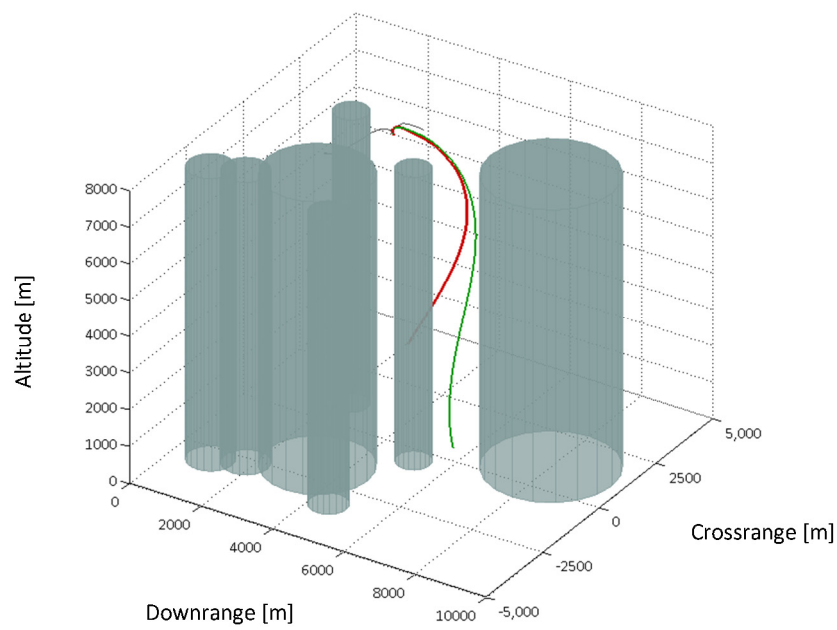
(1) を更新しない場合

シミュレーションにより得られた軌道プランナのツリーおよび、飛行軌跡を Fig. 5.3.14 に示す。力学フィルタの 3 自由度モデルと本シミュレーションの 6 自由度モデルの差異や、制御設計の不完全性により、飛行軌跡が徐々に目標軌道から離れていき、機体が飛行禁止区域内に入る結果となった (Fig. 5.3.15)。



(a) Bird's-Eye View 1

Fig. 5.3.14 6 DOF Simulation Result without Trajectory Update
(to be continued)



(b) Bird's-Eye View 2

Fig. 5.3.14 6 DOF Simulation Result without Trajectory Update
(to be concluded)

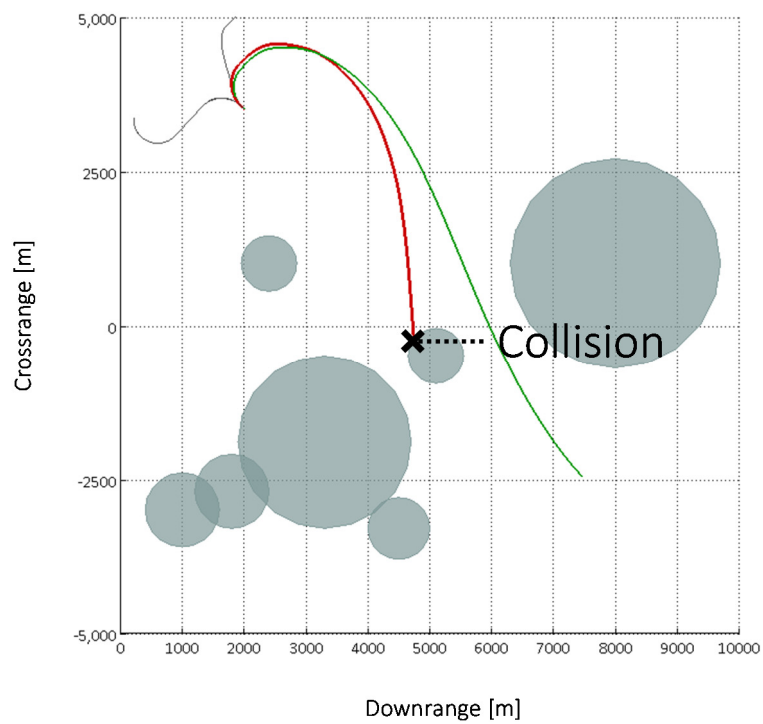
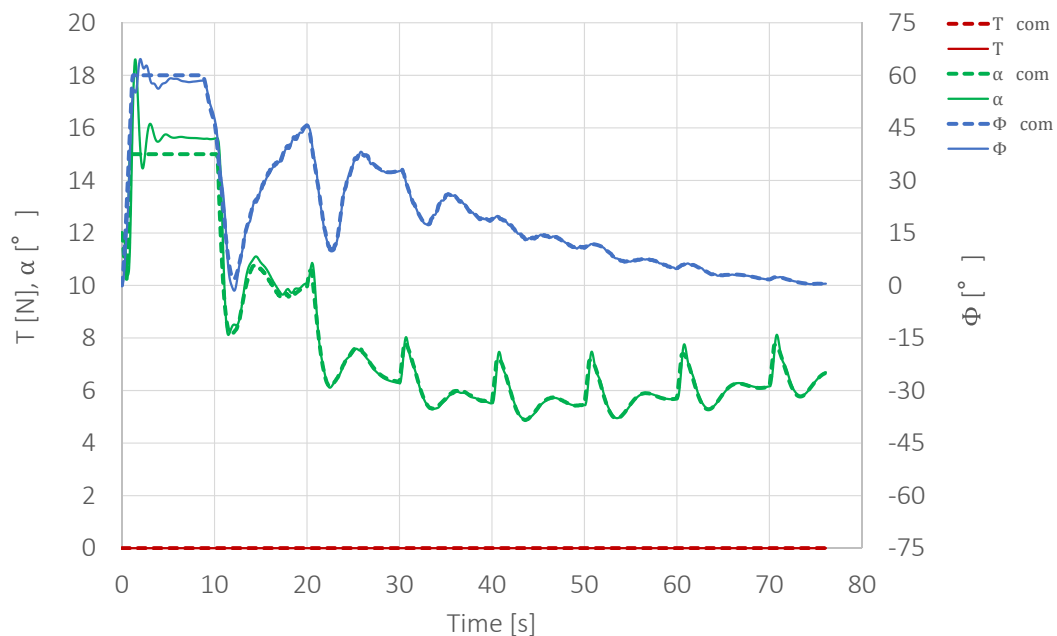


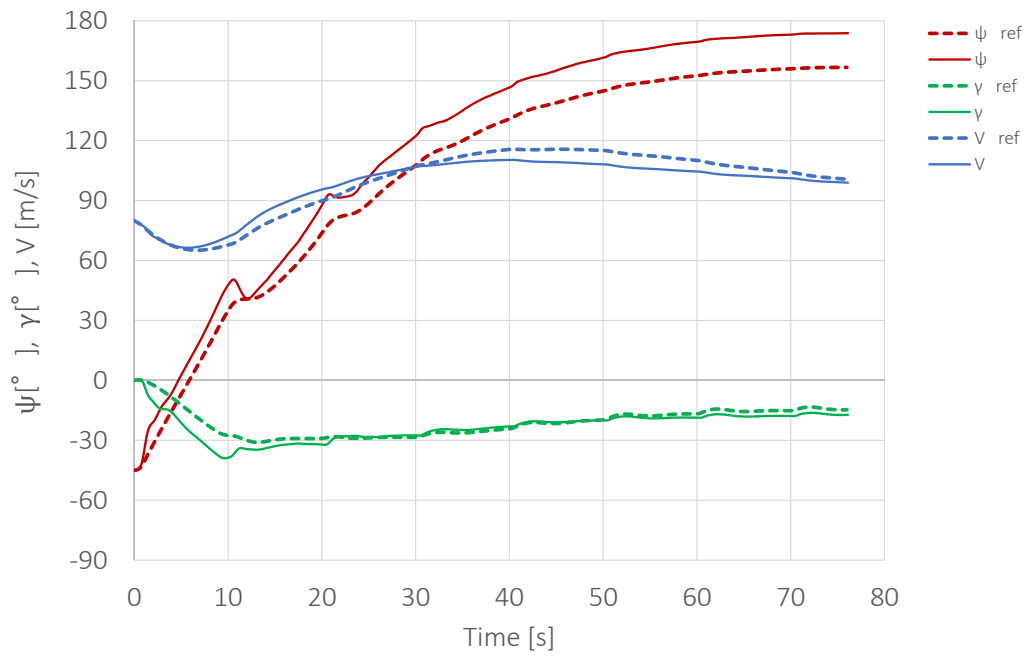
Fig. 5.3.15 No-Fly Zone Violation

次に、軌道プランナが与えた指令値と実際の機体の挙動を比較したグラフを Fig. 5.3.16(a)に示す。推力は無推力滑空飛行のため常に 0[N]である。迎角、バンク角ともに初期変動時に振動および偏差が見られるが、その他は概ね良好に追従している。高迎角時に偏差が残っているが、これは火星探査航空機同様にトリム舵角を 2 次曲線で近似していることによる影響と考えられる。また、方位角、経路角、速度に関して、軌道プランナの計算結果（添字 ref で表す）と機体の挙動を比較した結果を Fig. 5.3.16(b)に示す。いずれの結果も軌道プランナの値に追従傾向を示しているが、特に方位角に関して終端地点で約 20[°]の誤差が確認できる。



(a) Thrust, Angle of Attack and Bank Angle

Fig. 5.3.16 Reference and Response State Variables without Trajectory Update
(to be continued)



(b) Direction Angle , Flight Path Angle and Velocity

Fig. 5.3.16 Reference and Response State Variables without Trajectory Update
(to be concluded)

この他に, 得られた舵角および機体の角速度を Fig. 5.3.17 および Fig. 5.3.18 に示す. Fig. 5.3.17 よりエレベータ δ_e が周期的に変動しているが, これは軌道プランナのベジエ曲線を修正する際に α_{com} が周期的に変動するためである. その他は, 大きな振動などなく, リミッターにも掛からずに安定して飛行していることが分かる.

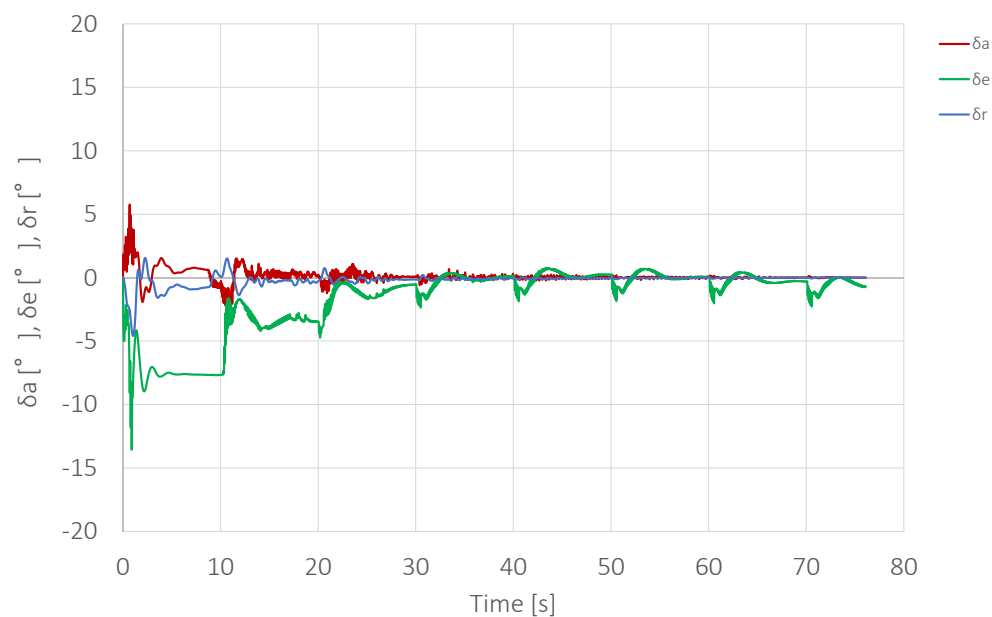


Fig. 5.3.17 Control Surface Angle Profile without Trajectory Update

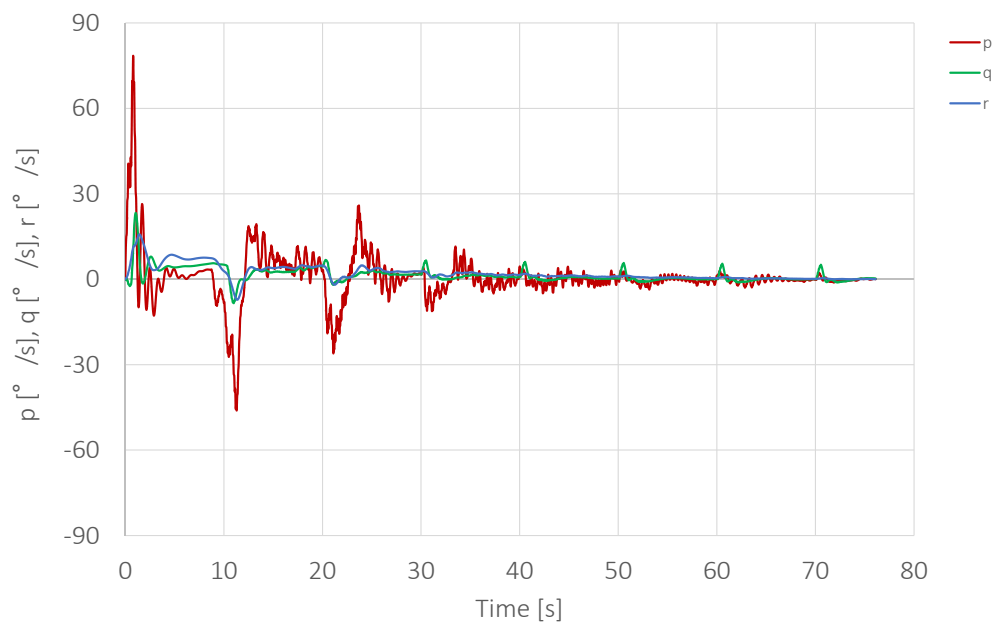
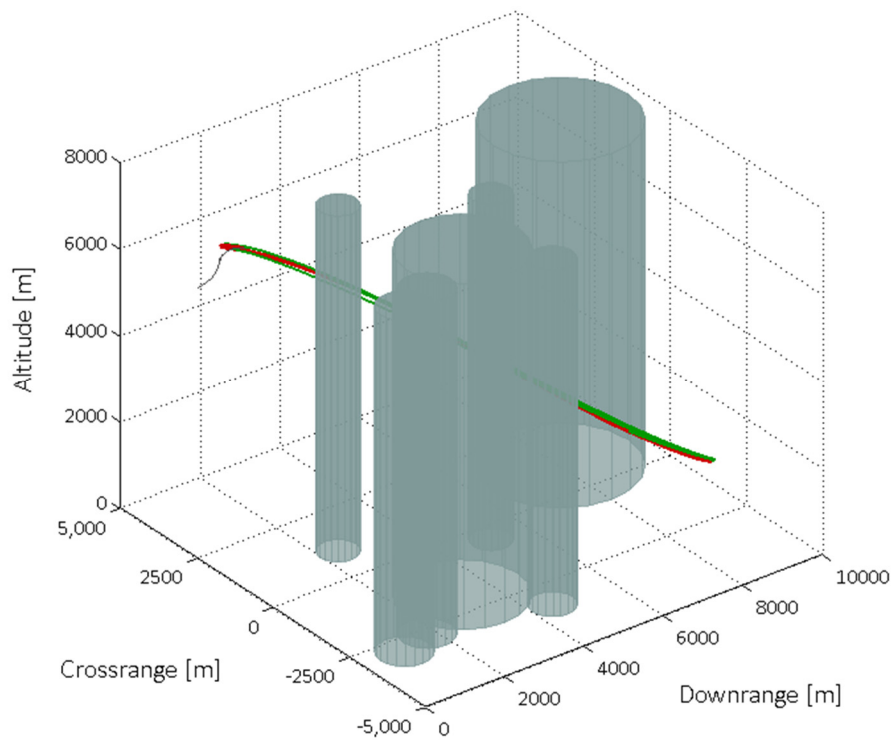


Fig. 5.3.18 Angular Velocity Profile without Trajectory Update

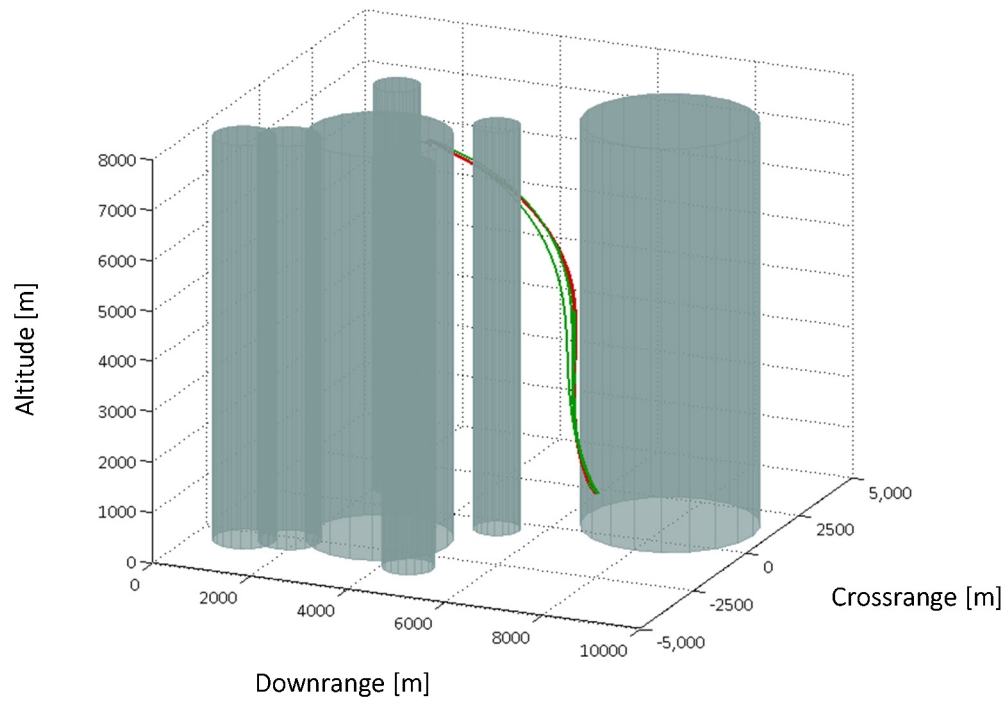
(2) 軌道を更新する場合

次に、軌道プランナを用いて 20 秒毎に目標軌道を更新した。シミュレーションの結果を Fig. 5.3.19 に示す。図では、軌道更新がなされた座標を黒色丸プロットで示しており、合計 7 回の軌道生成が行われた。シミュレーションの結果より、軌道修正をすることでゴールに到達できる誘導飛行が行えることが明らかとなった。なお、今回のケースでは、初回の探索時以外は、いずれもベジエ曲線の疑似軌道のみでゴールに到達できたためランダム探索フェーズが実行されず、ツリーの枝（黒色実線）の成長がほとんど見られない。

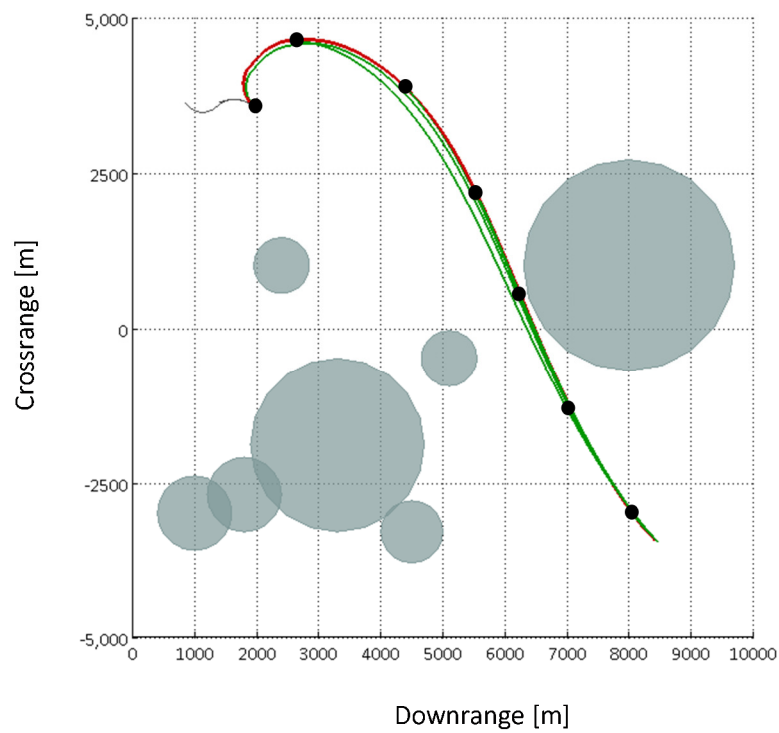


(a) Bird's-Eye View 1

Fig. 5.3.19 6 DOF Simulation Result with Trajectory Update
(to be continued)



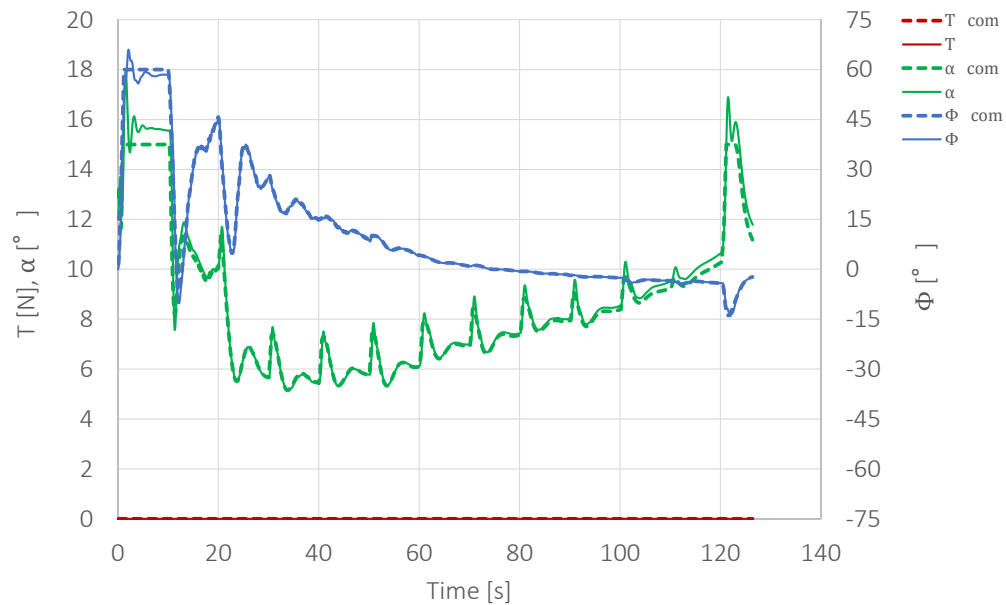
(b) Bird's-Eye View 2



(c) Top View

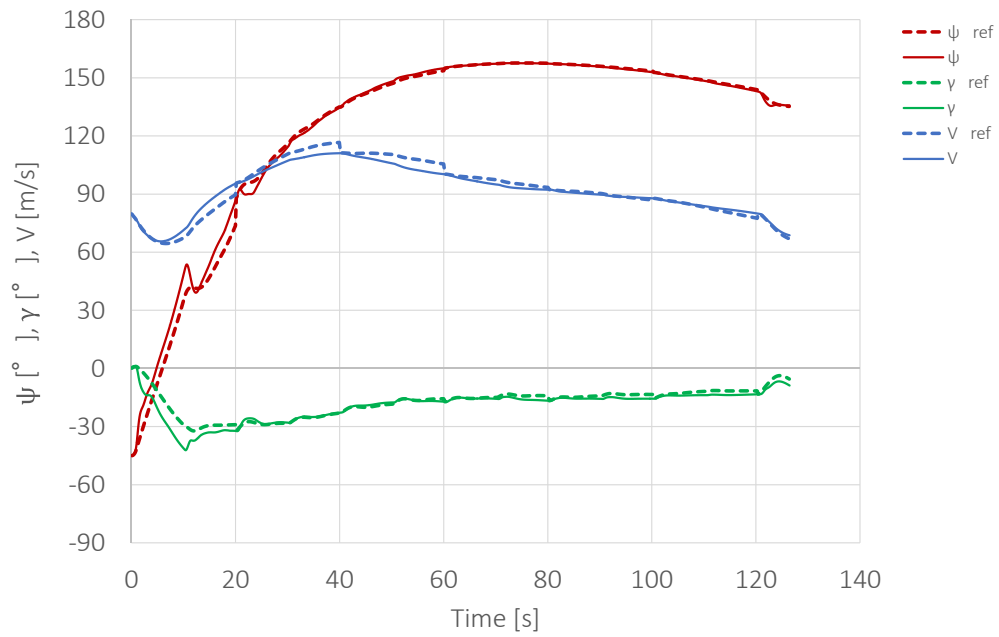
Fig. 5.3.19 6 DOF Simulation Result with Trajectory Update
(to be concluded)

軌道プランナが与えた指令値と実際の機体の挙動を比較したグラフを Fig. 5.3.20(a) に示す. 高迎角時に多少の偏差が見られるものの, 良好に追従している. また, 方位角, 経路角, 速度に関して, 軌道プランナの計算結果 (添字 **ref** で表す) と機体の挙動を比較した結果を Fig. 5.3.20(b) に示す. 特に軌道プランナから得られる速度 V_{ref} において 20 秒毎の軌道更新時に目標値が不連続となっているのは, 更新時点の機体の状態を初期値として軌道プランナが計算されるためである. この更新により, Fig. 5.3.16 で生じた軌道誤差が修正され, ゴール地点に到達できる軌道が描けている.



(a) Thrust, Angle of Attack and Bank Angle

Fig. 5.3.20 Reference and Response State Variables with Trajectory Update
(to be continued)



(b) Direction Angle , Flight Path Angle and Velocity

Fig. 5.3.20 Reference and Response State Variables with Trajectory Update
(to be concluded)

この他に、得られた舵角および機体の角速度を以下に示す。目標値が大きく変化する初期飛行の段階では、激しい変動が見られるが以降は概ね安定して飛行していることが分かる。エレベータ δ_e の周期的な変動は、軌道更新およびベジエ曲線による疑似軌道更新時（枝成長時間は 10 秒）に迎え角のコマンド α_{com} が変動するためである。

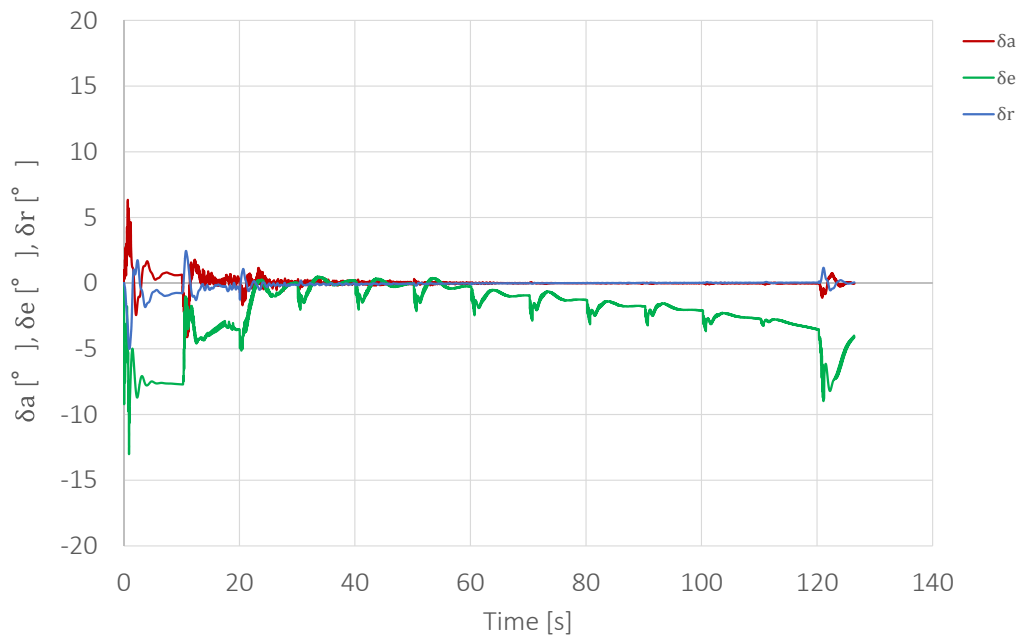


Fig. 5.3.21 Control Surface Angle Profile with Trajectory Update

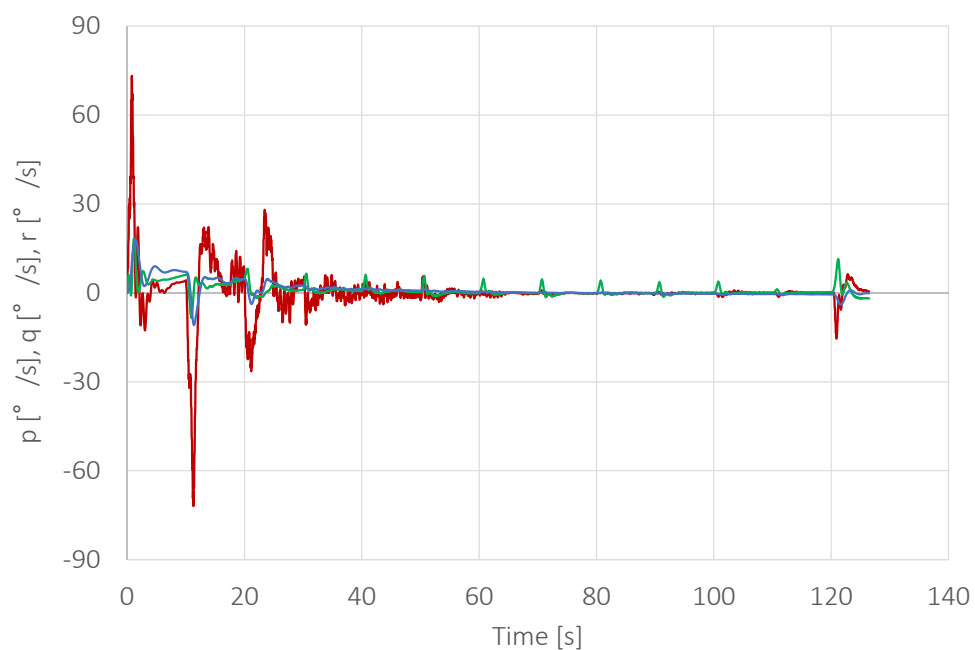


Fig. 5.3.22 Angular Velocity Profile with Trajectory Update

5.4 本章のまとめ

本章では, はじめに飛行シミュレーションに用いる 6 自由度運動方程式と制御則について記述した. そして, 火星探査航空機と有翼ロケット実験機 **WIRES#015** のモデルを用いて, 軌道更新を行わないケースと軌道更新をリアルタイムで行うケースを検証した. いずれの機体モデルにおいても, 初期の軌道プランナから得た指令値のみを用いて飛行する場合 (軌道を更新しない場合), モデル誤差や制御設計の不完全性などの影響により徐々に目標軌道から逸脱した. その結果, 障害物に衝突したり, 飛行禁止区域内を飛行する事態が発生し, ミッションの成立性に大きな問題があることが分かった. そこで, 軌道の更新を一定時間毎に行い軌道修正を行うことにより, 両モデルにおいて無事に目的地まで誘導飛行できることを確認した. よって, 自律飛行システムにおいて軌道プランナによるリアルタイム誘導が極めて重要であることが検証により明らかとなった.

第 6 章 UAV によるリアルタイム誘導計算飛行実験

6.1 はじめに

第 5 章では，リアルタイム誘導の必要性を明らかにした．そこで本章では，実際に小型 UAV に搭載する CPU に融合型軌道プランナを実装して，オンボードのリアルタイム誘導計算の飛行実験を行う．本実験により，飛行中の機体から取得されるデータを用いて，軌道生成が問題なく行えることを検証する．また，オンボード CPU による軌道生成時間も計測し，軌道プランナのリアルタイム性能を評価する．

6.2 実験システム

6.2.1 システム概要

飛行実験システムの構成図を Fig. 6.2.1 に示す．図の左側にはシステム全体の構成を示し，オンボード計算を実行するための機体システム，機体から送信されるデータを受信し監視するテレメトリシステム，および機体を操縦するコントローラの 3 つで構成される．また，図の右側は，機体に搭載されるオンボードシステムの構成を示す．将来的な自律飛行を見据えて，航法，誘導，制御それぞれの CPU を備える．航法 CPU には GPS (Global Positioning System)/ INS (Inertial Navigation System) 複合航法センサ，エアデータを取得する ADS (Air Data System)，およびそれらの計測データを記録する SD ロガーを備える．また，制御 CPU にはコントローラの受信機，テレメトリの送信機，推力用のブラシレス DC モータ，およびエルロン，エレベータ，ラダー用のサーボモータが接続されている．さらに，誘導 CPU では本研究の誘導型軌道プランナを実装して軌道計算を行い，その結果を SD ロガーで記録する．

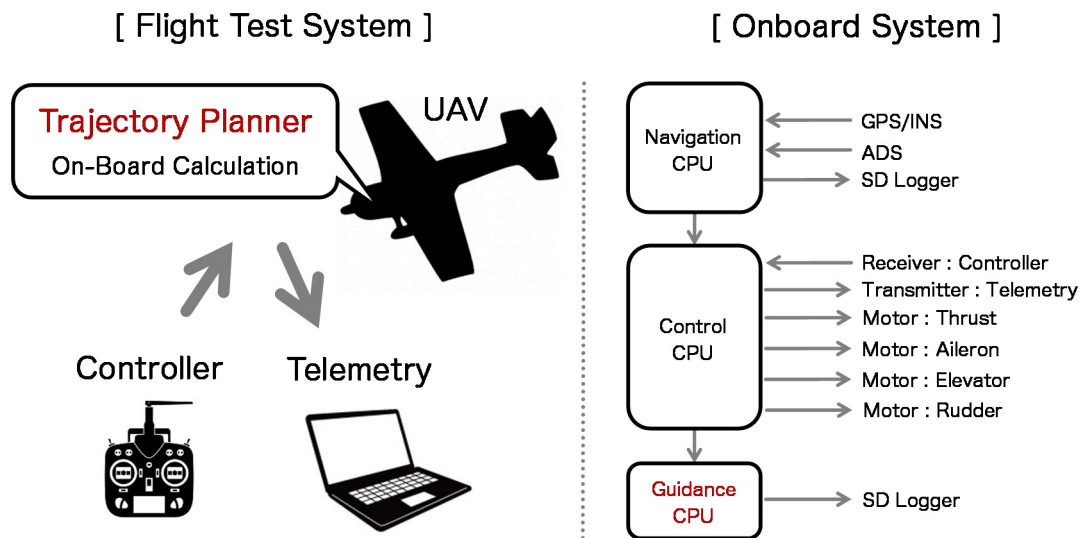


Fig. 6.2.1 Flight Test and Onboard System

6.2.2 機体システム

6.2.2.1 機体諸元

機体の外観および諸元を Fig. 6.2.2 と Table 6.2.1 にそれぞれ示す.



Fig. 6.2.2 UAV

Table 6.2.1 Specification of UAV

Body Length	[m]	1.18
Wing Span	[m]	1.90
Wing Area	[m ²]	0.335
Mass	[kg]	2.492

6.2.2.2 搭載機器

搭載機器の配置図を以下に示す.

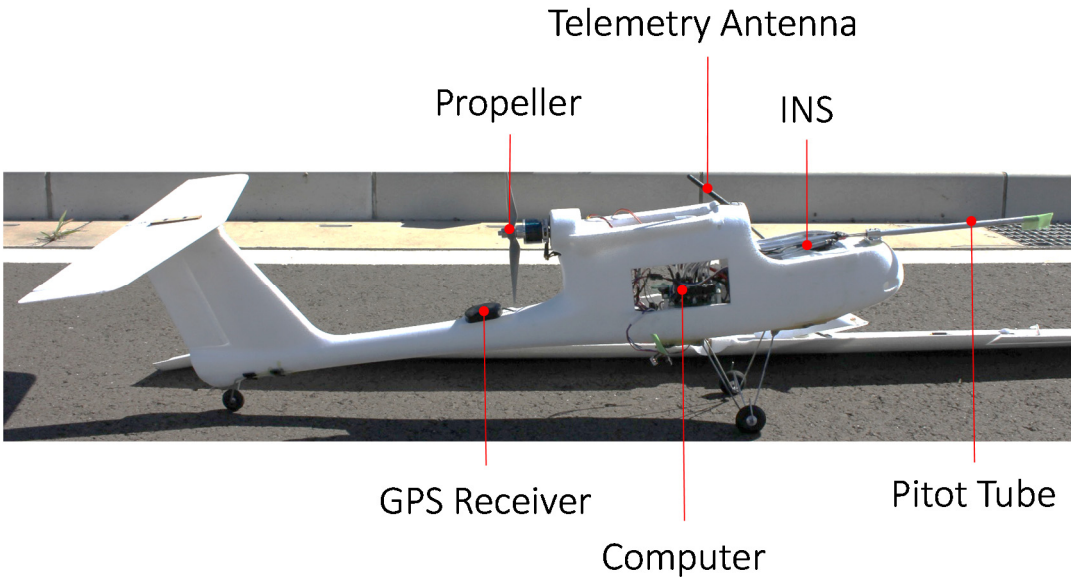


Fig. 6.2.3 System Arrangement

また, 搭載機器の各仕様を以下に記載する.

(1) コンピュータ

航法および制御基板は, ゼノクロス株式会社 UAV 事業部が設計製作した. 基板外観図および CPU の仕様を以下に示す.

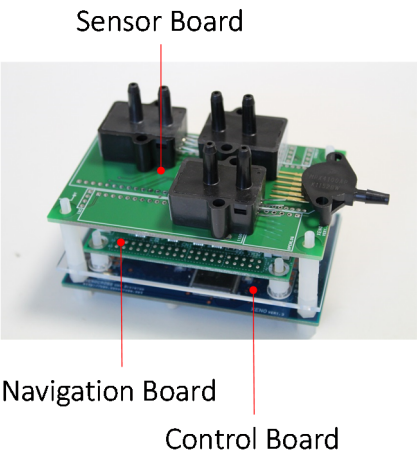


Table 6.2.2 Navigation and Control
CPU Specification

Name	[-]	H8S2638
Clock	[MHz]	20
Memory	[KB]	256
RAM	[KB]	16

Fig. 6.2.4 Navigation and Control Board

また誘導基板は、制御基板とシリアル通信で繋がっている．誘導基板の外観図と CPU の仕様を以下に示す．

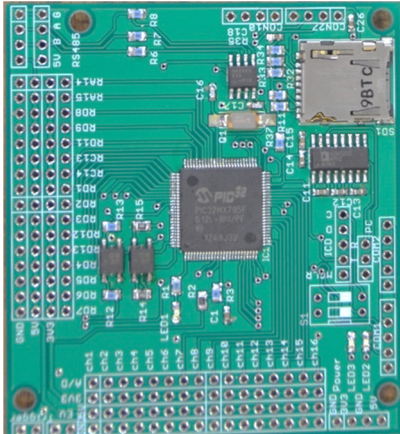


Fig. 6.2.5 Guidance Board

Table 6.2.3 Guidance CPU Specification

Name	[-]	PIC32MX795F512L
Clock	[MHz]	80
Memory	[KB]	512
RAM	[KB]	128

(2) ADS

気圧高度計測には絶対圧センサを用いて、動圧，迎角，横滑り角の計測には差圧センサを用いている．それぞれの外観と仕様を以下に示す．



Fig. 6.2.6 Absolute Pressure Sensor (Altitude)

Table 6.2.4 Specification of Altitude Pressure Sensor

Pressure Range	[kPa]	20~105
Supply Voltage	[V]	5
Accuracy	[%V _{FSS}]	1.8
Operating Temperature	[°C]	-40~125



Fig. 6.2.7 Differential Pressure (Velocity)

Table 6.2.5 Specification of Velocity Pressure

Sensor		
Pressure Range	[inH ₂ O]	±10
Supply Voltage	[V]	5
Operating Temperature	[°C]	-25~85



Fig. 6.2.8 Differential Pressure
(Angle of Attack and Side Slip Angle)

Table 6.2.6 Specification of Angle of Attack and Side Slip Angle Pressure Sensors

Pressure Range	[inH ₂ O]	±5
Supply Voltage	[V]	5
Operating Temperature	[°C]	-25~85

また、各センサの較正試験を行った。気圧高度計の較正結果を Fig. 6.2.9 に、ピトー管を用いた迎角、横滑り角の較正結果を Fig. 6.2.10 と Fig. 6.2.11 にそれぞれ示す。

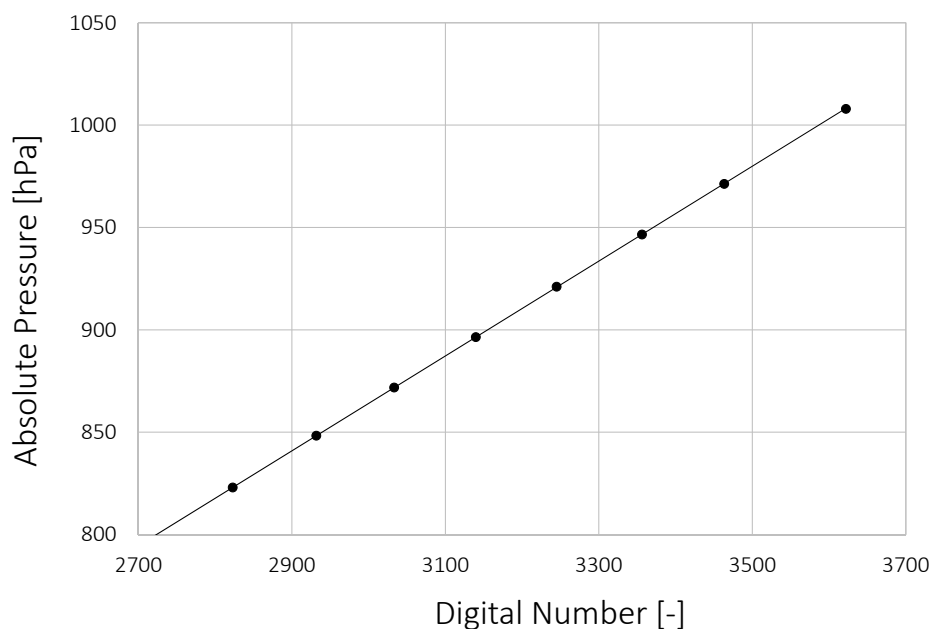


Fig. 6.2.9 Calibration of Pressure Altimeter

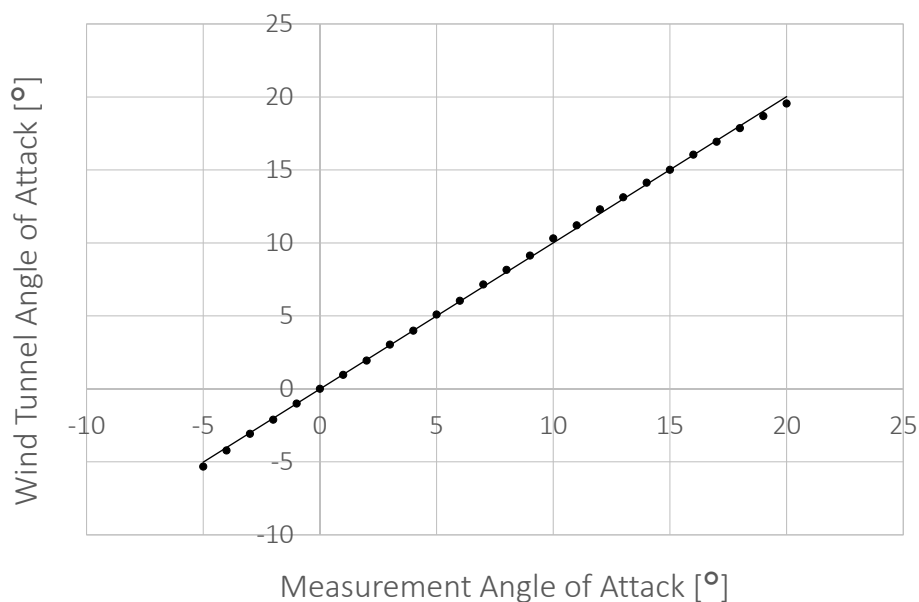


Fig. 6.2.10 Calibration of Angle of Attack Using Pitot Tube

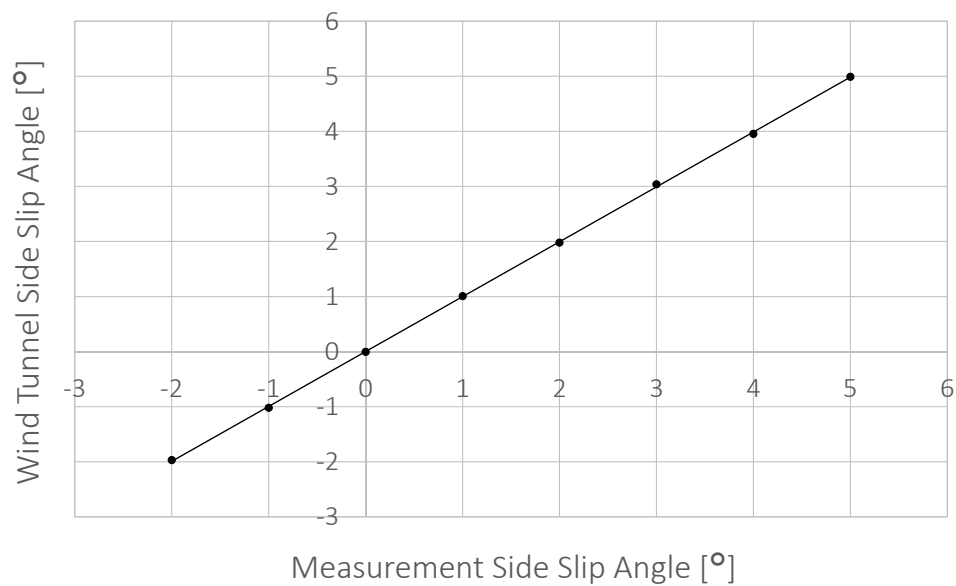


Fig. 6.2.11 Calibration of Side Slip Angle Using Pitot Tube

(3) GPS/INS

複合航法センサの外観及び主要諸元を以下にそれぞれ示す。



Fig. 6.2.12 GPS/INS Sensor

Table 6.2.7 Specification of GPS/INS Sensor

Position Accuracy	[m]	± 2.5
Velocity Accuracy	[m/s]	± 0.1
Attitude Accuracy	[°]	± 0.25
Acceleration Range	[g]	± 5
Gyro Range	[°/s]	± 300
Data Output Rate	[Hz]	500

(4) 推進器

機体に用いるプロペラは長さ 9 インチ、ピッチ 6 インチのものを用いて、モータはハイペリオン製のブラシレスモータを用いる。モータの外観および諸元を以下にそれぞれ示す。



Fig. 6.2.13 Brushless Motor

Table 6.2.8 Specification of Brushless Motor

Weight	[kg]	0.162
KV	[rpm/V]	922
Max. Current	[A]	56
Max. Power	[W]	850

6.2.2.3 空力係数

機体の揚力係数、抗力係数は理論式により算出した。航空機力学入門^[62]によると、翼厚比 20%程度までの二次元翼揚力傾斜の近似値を

$$\left(\frac{\partial C_l}{\partial \alpha}\right) = 2\pi + 4.7(t/c)(1 + 0.00375\phi_{TE}) \quad (6.1)$$

としている。ここで、 (t/c) は翼厚比、 ϕ_{TE} は後縁角を $^{\circ}$ で表したものである。このとき揚力係数は、以下のように推定できる。

$$C_l = C_{l0} + \left(\frac{\partial C_l}{\partial \alpha}\right)\alpha \quad (6.2)$$

C_{l0} は迎え角 0° の揚力係数であるが、これは XFLR5^[63]と呼ばれる翼型解析ソフトにより、機体の翼型に近く模型飛行機で一般的に用いられる翼型 CLARK-Y を解析した結果を用いる。

また，抗力係数の推算式は再び航空機力学入門^[62]を参考にすると

$$C_d = C_{dmin} + \frac{C_l^2}{\pi e AR} \quad (6.3)$$

とされる． e は飛行機効率と呼ばれ，高翼機では 0.8 程度の値である．また， AR はアスペクト比を表し， C_{dmin} は抗力の最小値であるが，この値は迎角ゼロのときの抵抗値と仮定し，風洞試験で機体胴体の抵抗係数を計測した (Fig. 6.2.14)．



Fig. 6.2.14 Wind Tunnel Test for Body Drag Measurement

以上の方法により，機体の揚力係数と抗力係数を推定した結果を Fig. 6.2.15 に示す．

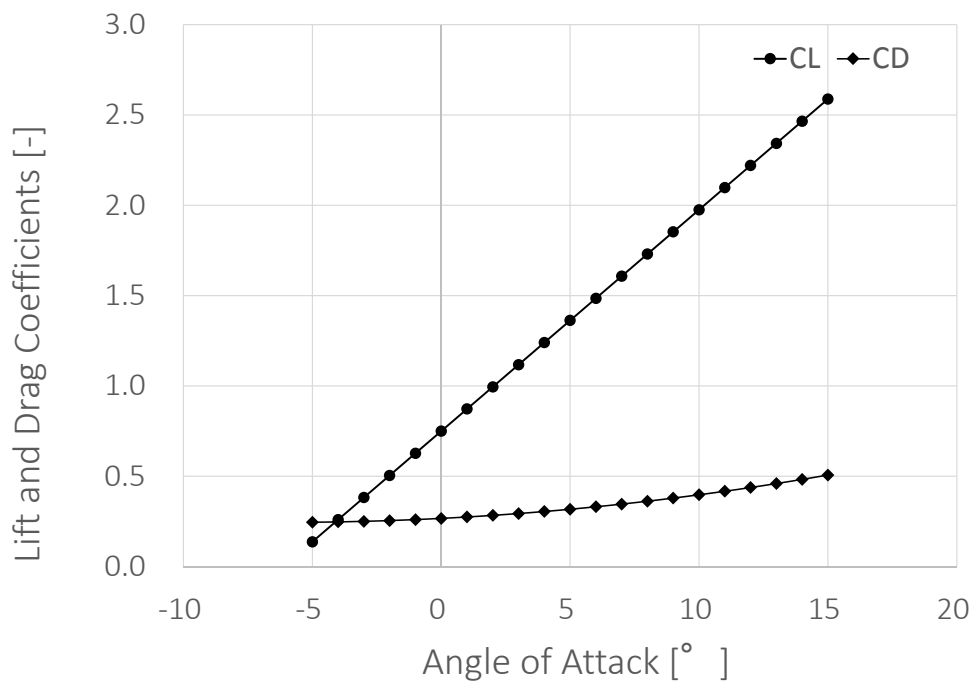


Fig. 6.2.15 Lift and Drag Coefficients of UAV

6.2.3 テレメトリシステム

本実験システムには地上で機体の状態をモニタリング可能なテレメトリシステムを備えている。機体データの監視には、株式会社ゼノクロスが作成したモニタリングソフトを使用する。テレメトリの通信モジュール及びモニタリングソフトを以下にそれぞれ示す。

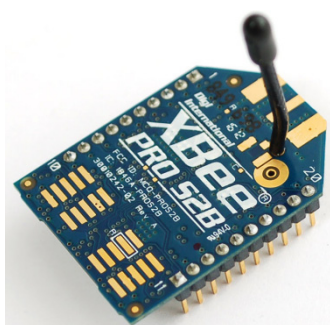


Fig. 6.2.16

Communication Module

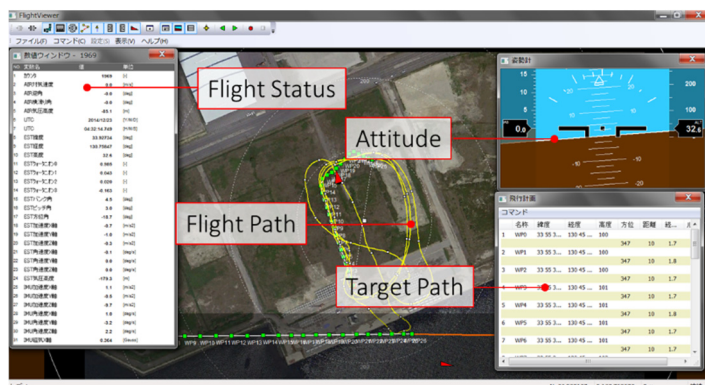


Fig. 6.2.17 Monitoring Software

6.2.4 コントローラ

本章のリアルタイム誘導飛行実験において機体は、自律飛行を行わず操作者によって操縦される。操縦に用いるコントローラを Fig. 6.2.18 に示す。コントローラでは、推力、エルロン、エレベータ、ラダーの操作が行え、将来の自律飛行に備えてオートパイロットへの切り替えも行える。



Fig. 6.2.18 Radio Controller

6.3 実験条件

6.3.1 実験場所

飛行実験は福岡県若松区の響灘で行う (Fig. 6.3.1). 探索領域はダウンレンジ, クロスレンジをそれぞれ 500m, 高度 (海拔) を 300m とした.

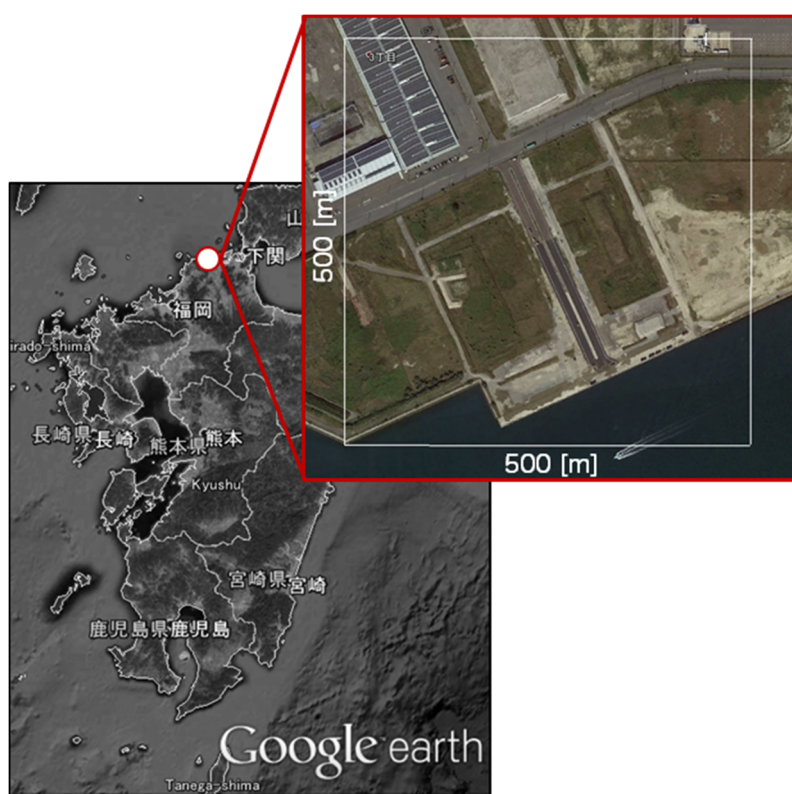


Fig. 6.3.1 Flight Test Area

6.3.2 障害物配置

本飛行実験では, 障害物を仮想的に配置する. すなわち, 誘導 CPU の内部に障害物の配置と大きさを予めプログラミングして誘導計算を行う. ここでは, 3つの障害物配置パターンによって実験を行う. 書くパターン (A,B,C) を Fig. 6.3.2 に示す. Pattern A, B, C の障害物数はそれぞれ 1, 3, 5 個である.

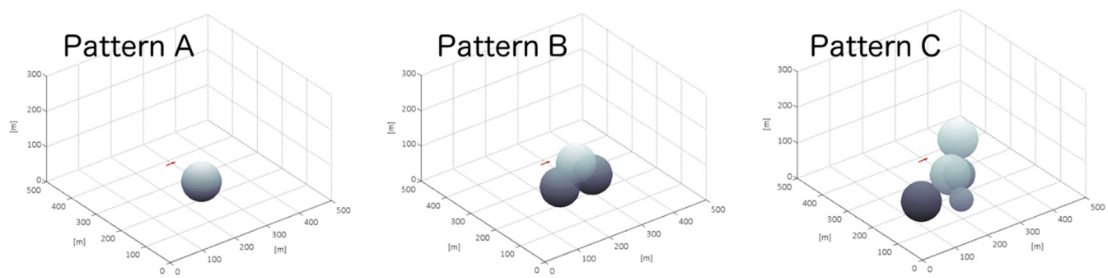


Fig. 6.3.2 Obstacle Pattern

6.3.3 軌道プランナのパラメータ

スタートの位置や初期値は誘導計算開始時の飛行データを用いる. 軌道プランナに用いるその他の各種パラメータを下表に示す.

Table 6.3.1 Trajectory Planner Setup

ランダム方位角指令値の範囲	[°]	$-180 \leq \psi_{rand} \leq 180$
ランダム経路角指令値の範囲	[°]	$-20 \leq \gamma_{rand} \leq 20$
枝成長時間 T	[s]	5
ベジエ制御点比例定数 λ	[-]	0.3
衝突回避マージン τ	[m]	5
探索終了条件：許容距離誤差	[m]	10 以下
探索終了条件：許容角度誤差	[°]	20 以下

また，力学フィルタの制御器において，本実験で使用する目標速度は $V_{ref} = 15$ [m/s]とし，各制御ゲインは下表を用いた．

Table 6.3.2 Control Gains

K_{vp}	[1/s ²]	0.05	K_{vd}	[1/s]	0.01
K_{gp}	[1/s ²]	10.0	K_{gd}	[1/s]	5.0
K_{pp}	[1/s ²]	10.0	K_d	[1/s]	5.0

さらに，力学フィルタには実用的な軌道を生成させるため，各コマンドに制限値を設けている．設定した制限値を以下に示す．

Table 6.3.3 Command Limiters

$0 \leq T_{com} \leq 10$	[N]	$-20 \leq \dot{T}_{com} \leq 20$	[N/s]
$-10 \leq \alpha_{com} \leq 10$	[°]	$-20 \leq \dot{\alpha}_{com} \leq 20$	[°/s]
$-30 \leq \phi_{com} \leq 30$	[°]	$-60 \leq \dot{\phi}_{com} \leq 60$	[°/s]

以上の実験条件において，リアルタイム軌道生成実験を行う．結果を次節で述べる．

6.4 実験結果

本節では，UAV を用いたオンボードリアルタイム誘導計算飛行実験の結果を述べる．それぞれの障害物パターンに対し，特徴的な 2 例の解軌道を示す．

はじめに，飛行実験の様子を Fig. 6.4.1 に示す．実験結果の概要として，UAV をコントローラで操縦することで飛行可能なことを確認し，またテレメトリによる地上モニタリングができた．さらに，オンボードの誘導計算も全ての障害物パターンで解軌道を得られた．ただし，通信不良により軌道計算が開始しないケースが確認された．



Fig. 6.4.1 Real Time Calculation Flight Test

6.4.1 パターン A

障害物パターン A における軌道生成結果を 2 例示す (Fig. 6.4.2(a), Fig. 6.4.2(b))．黒線は軌道プランナにより探索された全ツリーを示し，緑線は得られた解軌道を表す．また，ゴールの位置から目標軌道角度を表す赤色のベクトルを図示しており，さらに得られた計算結果を用いて機体姿勢を簡易的な紙飛行機モデルで可視化し，モデルの付近には軌道プランナで得られた解軌道のスタート地点からの飛行経過時間を示している．

結果より，いずれもゴールに到達する軌道が得られた．Fig. 6.4.2(a)においては，初期位置と軌道角度が障害物に衝突する方向であったため，軌道プランナは障害物を回避するランダム探索を実行した様子 (0 秒～6 秒付近) が分かる．また，Fig. 6.4.2(b)では，スタートの時点でベジエ曲線の疑似軌道がゴールまで到達できたため，ランダム探索は実行されていない．しかし，疑似軌道を用いた軌道生成において，探索終了条件 (Table 6.3.1) を一度で満たすことができず，条件を満たせるまで複数回にわたり疑似軌道によりゴール付近を旋回している (約 20 秒以降)．

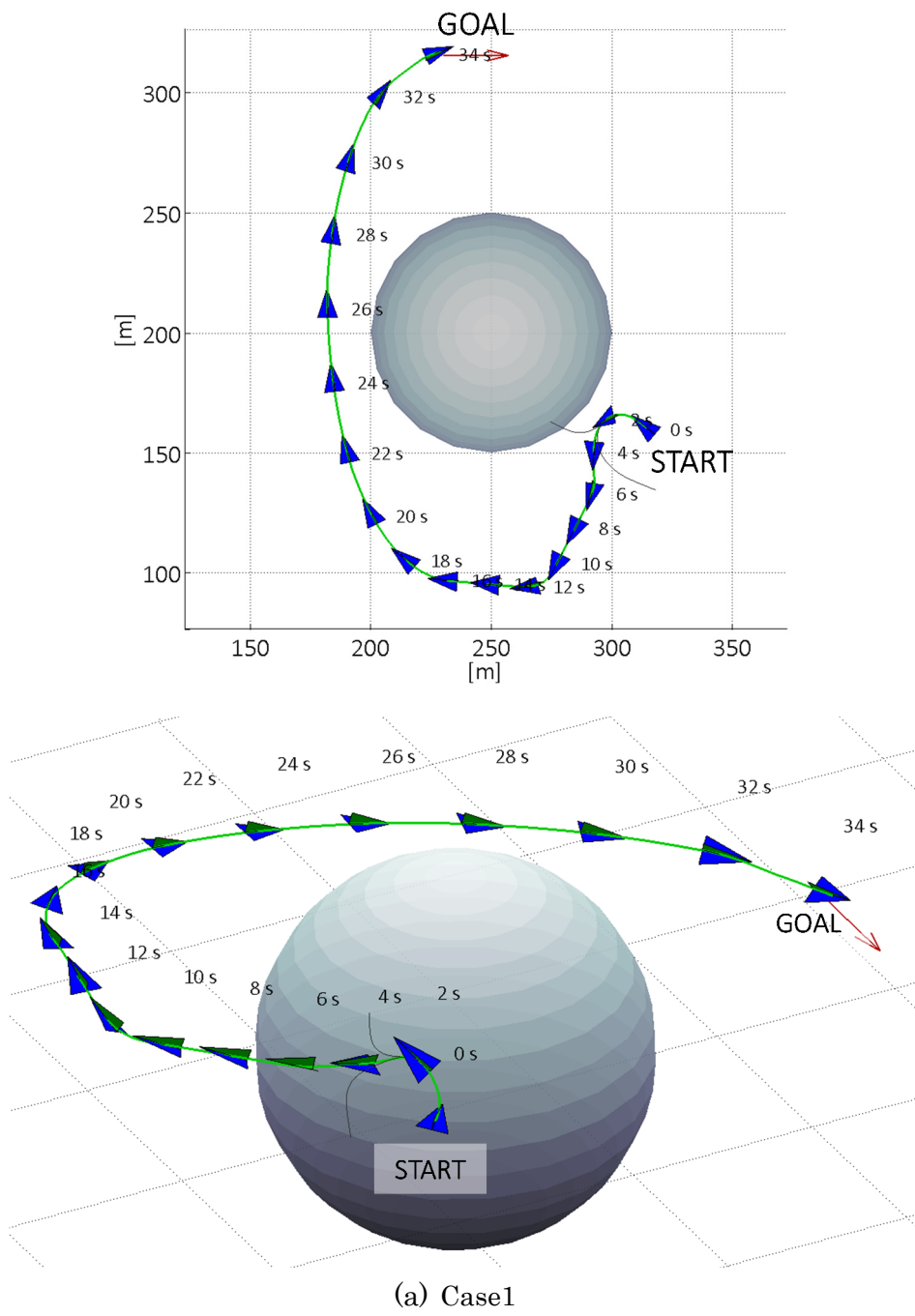


Fig. 6.4.2 Trajectory Generation Calculation of Pattern A
(to be continued)

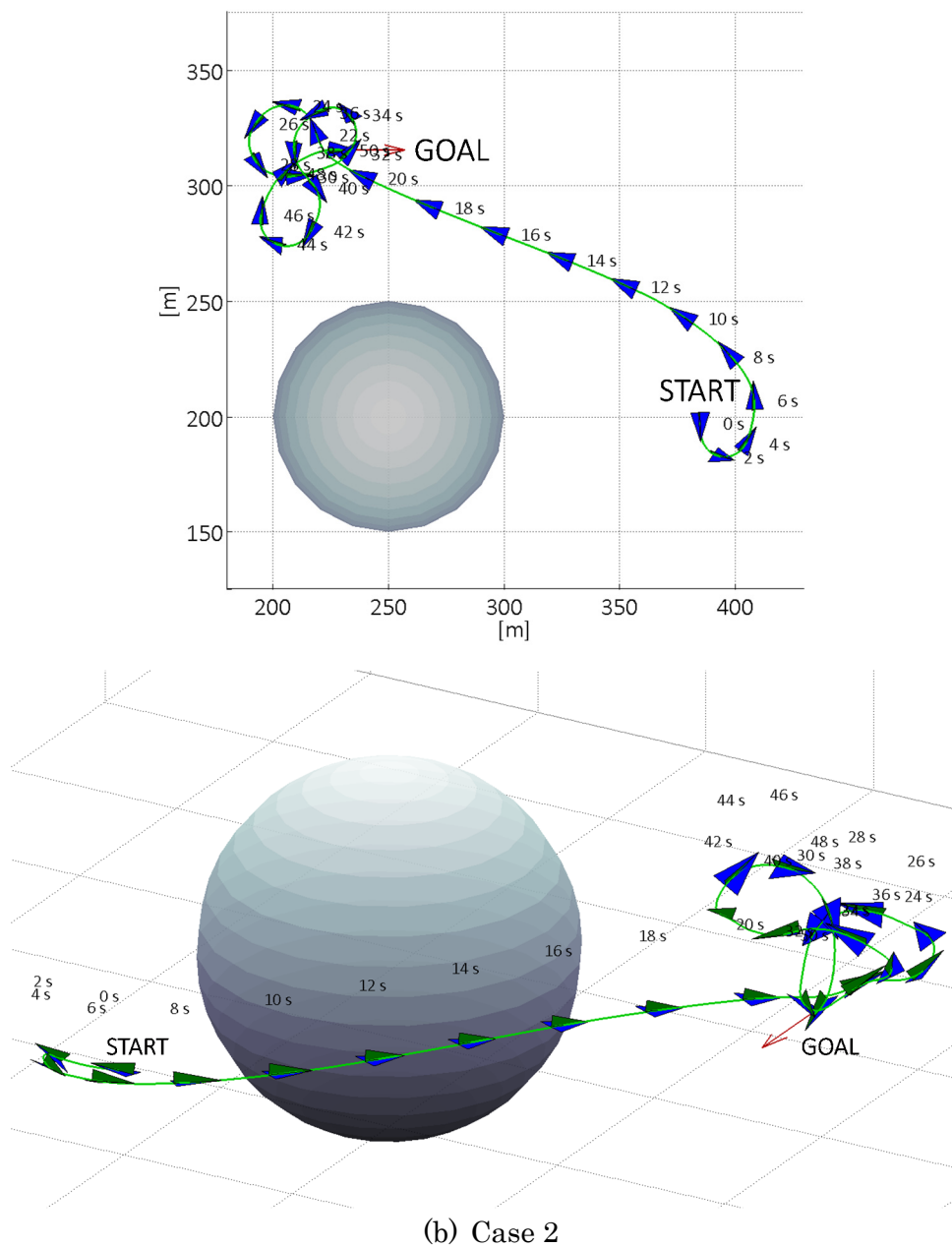


Fig. 6.4.2 Trajectory Generation Calculation of Pattern A
(to be concluded)

6.4.2 パターン B

障害物数が 3 個のパターン B における軌道生成結果を 2 例示す (Fig. 6.4.3(a), Fig. 6.4.3(b)). Fig. 6.4.3(a)では, 軌道プランナの初期段階でランダム探索が行われた. その後, 疑似軌道によりゴール付近まで向かっているが, 一度, 疑似軌道为目标とする運動計算に失敗して著しく降下する軌道を描き失敗 (44 秒付近) した. その後, 再度疑似軌道を用いてゴールへ到達している. Fig. 6.4.3(b)の結果では, ランダム探索により障害物の左側からゴールへアプローチしており, 約 30 秒までランダム探索が行われたため, 何度も旋回しながら探索する軌道を描いている.

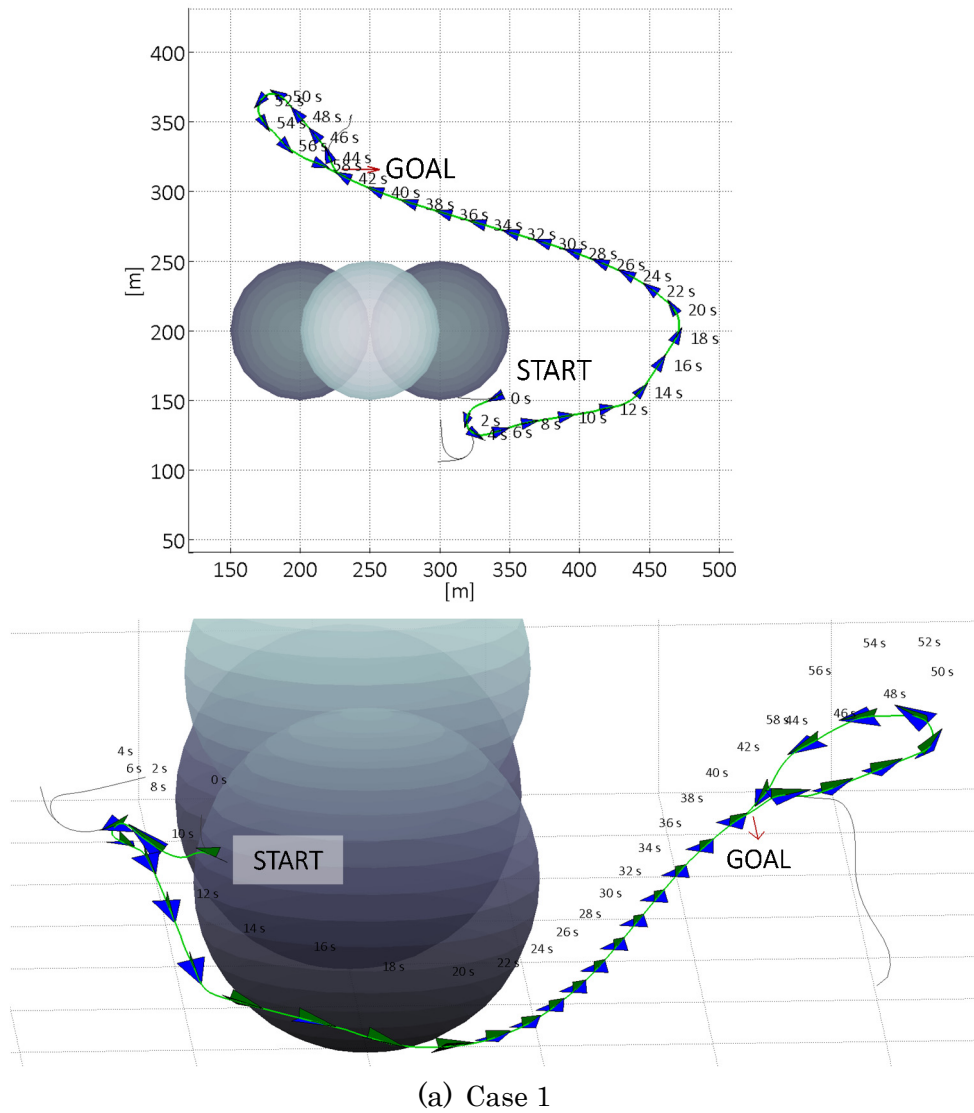
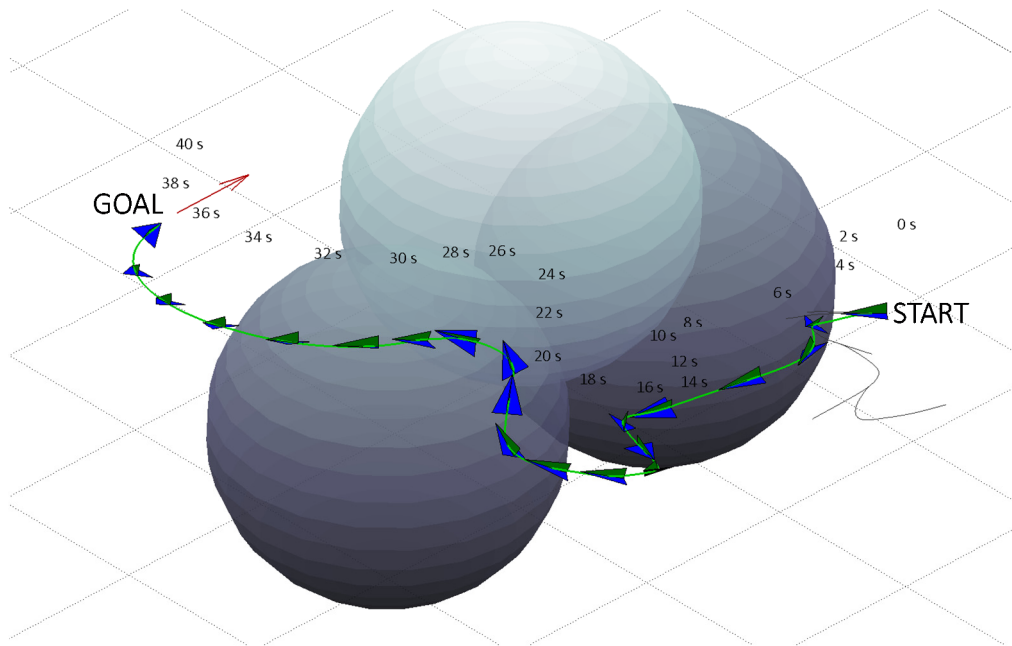
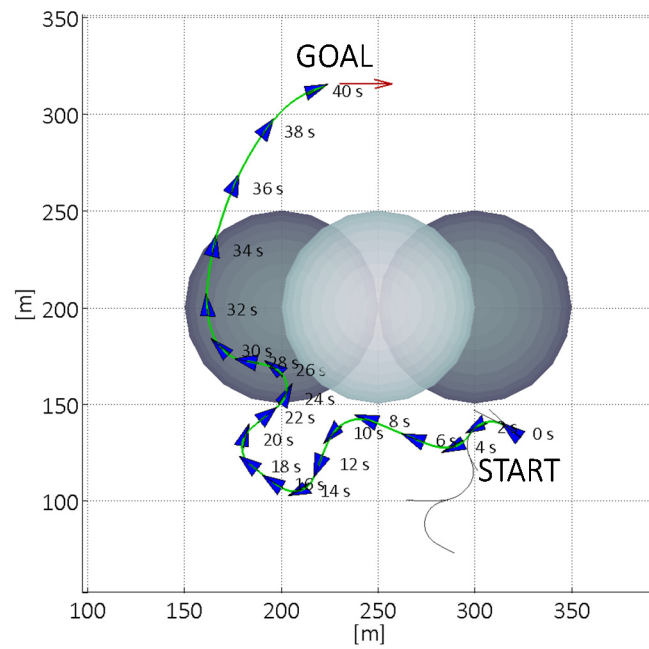


Fig. 6.4.3 Trajectory Generation Calculation of Pattern B
(to be continued)



(b) Case 2

Fig. 6.4.3 Trajectory Generation Calculation of Pattern B
(to be concluded)

6.4.3 パターン C

障害物数を 5 個まで増加させたパターン C における軌道生成結果を 2 例示す (Fig. 6.4.4(a), Fig. 6.4.4(b)). Fig. 6.4.4(a)の結果において, 初期段階は, 障害物を避けるようにランダム探索が実行され, ゴール付近ではパターン A のケース 2 と同様に複数回の疑似軌道による軌道生成を試みている. また, Fig. 6.4.4(b)の結果では, 初期位置がゴールから遠く, 複雑な障害物環境を回避する必要があったため, 多くのランダム探索が実行されたが, ゴールまで到達する軌道を描くことに成功した.

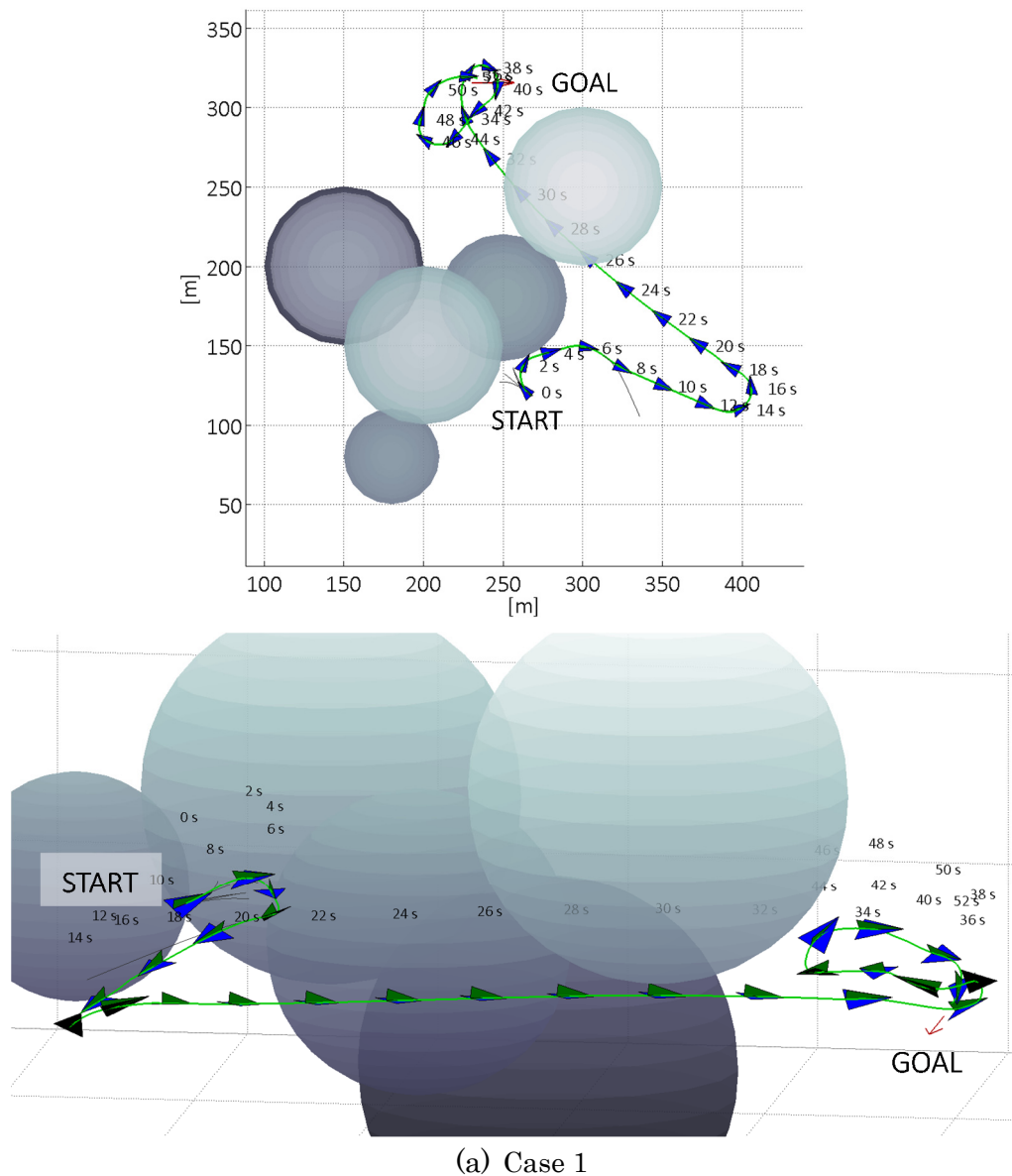
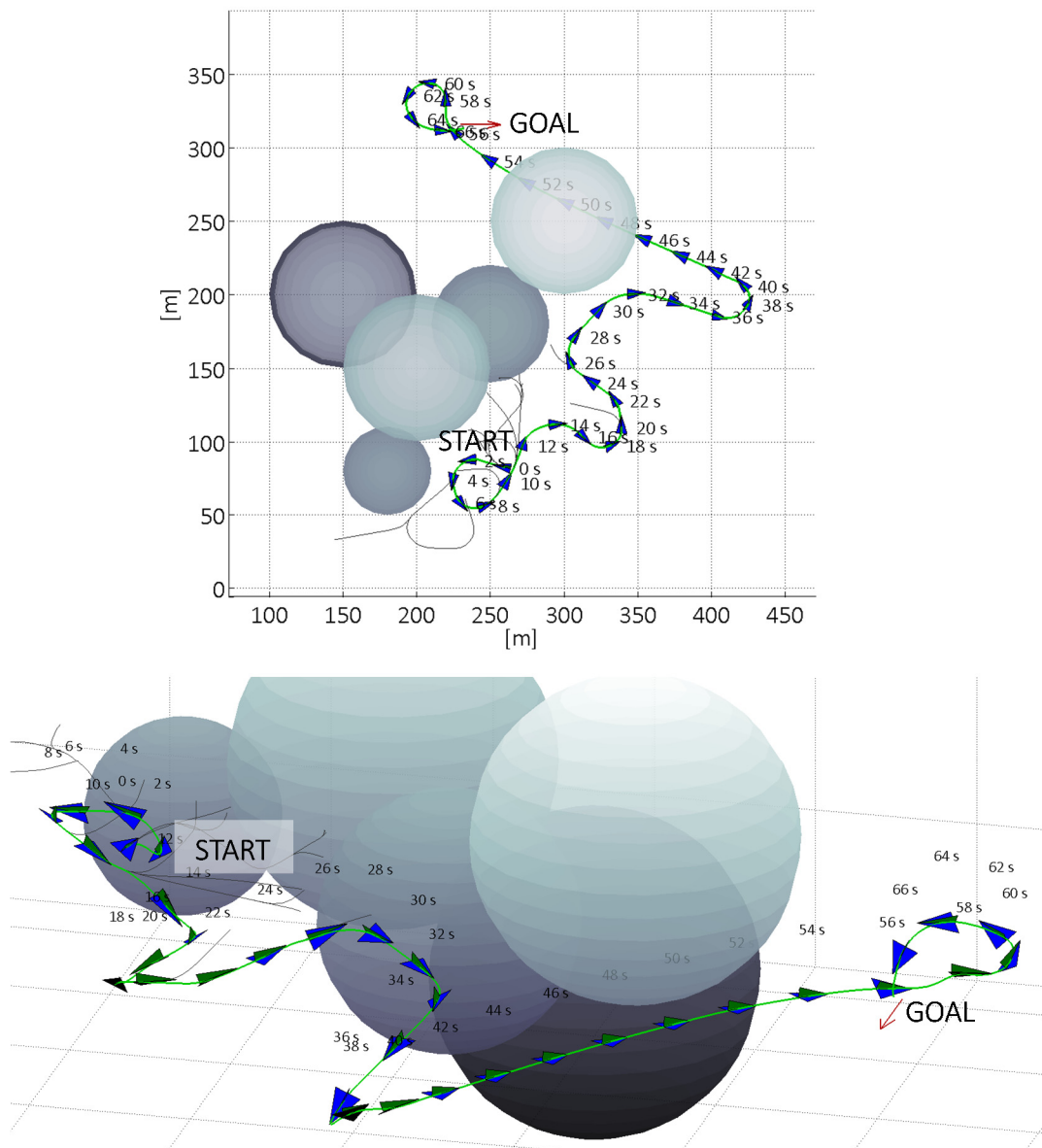


Fig. 6.4.4 Trajectory Generation Calculation of Pattern C
(to be continued)



(b) Case 2

Fig. 6.4.4 Trajectory Generation Calculation of Pattern C
(to be concluded)

ここで、パターン C の Case 2 で得られた結果の指令値時系列データを Fig. 6.4.5 に示す。図より、バンク角はリミッターに多くかかり急旋回しており、迎え角と推力は概ね滑らかに遷移している。初期のバンク角がリミッターより大きな値（約 $36[^\circ]$ ）となっているのは、計算開始時に機体から取得したバンク角がリミッターより大きく、その値を初期値として計算しているためである。

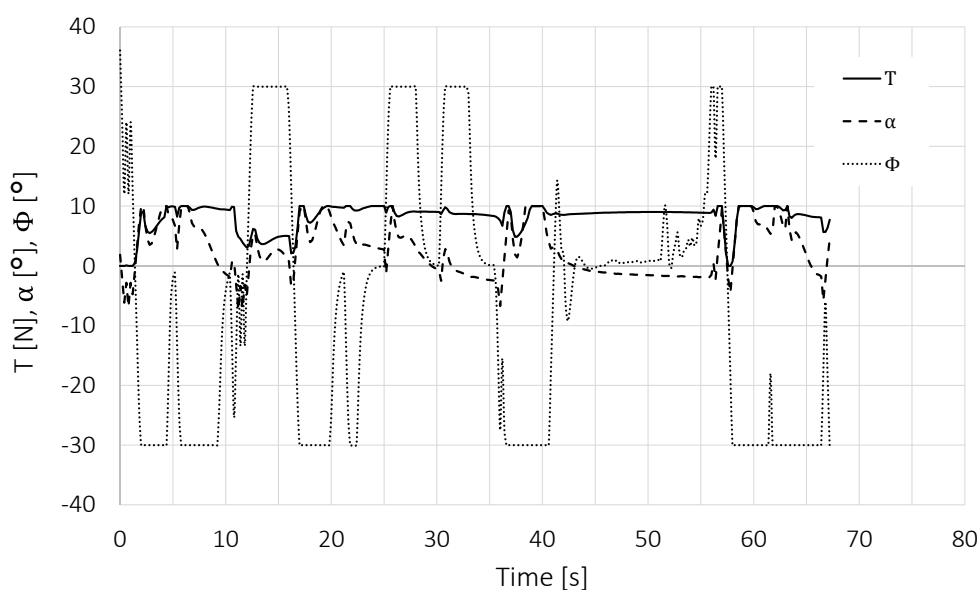


Fig. 6.4.5 Command Time Histories for Case 2 of Pattern C

また、軌道プランナにより得られた機体の状態量遷移を Fig. 6.4.6 に示す。方位角はランダム探索で与えられたランダム指令値に追従傾向があるが、経路角は追従性は高くない。これは大きくバンク角を取ると、迎え角等で経路角をコントロールすることが困難になるためと考えられる。速度に関しても、急旋回する箇所では速度が大きく低下しており指令値を維持していない。なお、36 秒以降は疑似軌道によりゴールへ向かうフェーズであるが、そこでは指令値を記録しておらず方位角、経路角は 0 を示している。

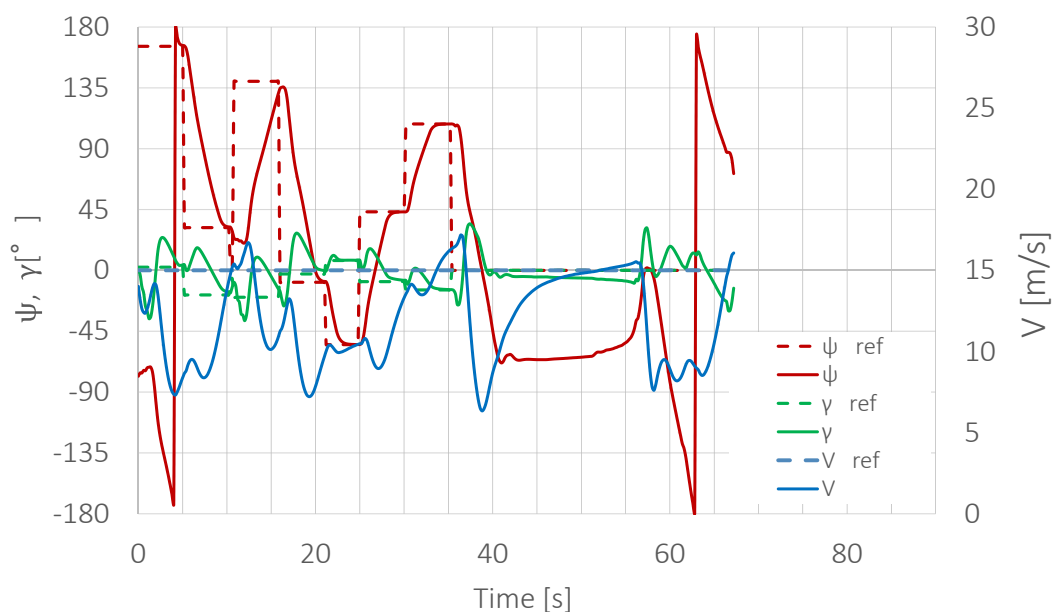


Fig. 6.4.6 Reference and Response Histories for Case 2 of Pattern C

さらに、パターン C において解軌道が得られなかった例を挙げる。通信不良により軌道生成が開始しない場合の他に、メモリ不足により途中で探索が終了するケースがあった。このときの結果を Fig. 6.4.7 に示す。探索開始時に多くのランダム探索が行われ、ゴール付近でもベジエ曲線の疑似軌道により複数回の反復が行われたが、到達できなかった。探索ツリーのデータを格納する領域を CPU のメモリ限界まで事前に確保していたが、ゴールへ到達する前に配列が一杯になり探索を終了した。このように、疑似軌道を用いた力学フィルタで目的地に到達できない場合が確認され、アルゴリズムの改良が必要であることが分かった。また、解軌道の生成に失敗した場合には、新たに得られる現在の飛行情報を用いて即座に再度軌道生成を行うなどの対策も有効と考えられる。

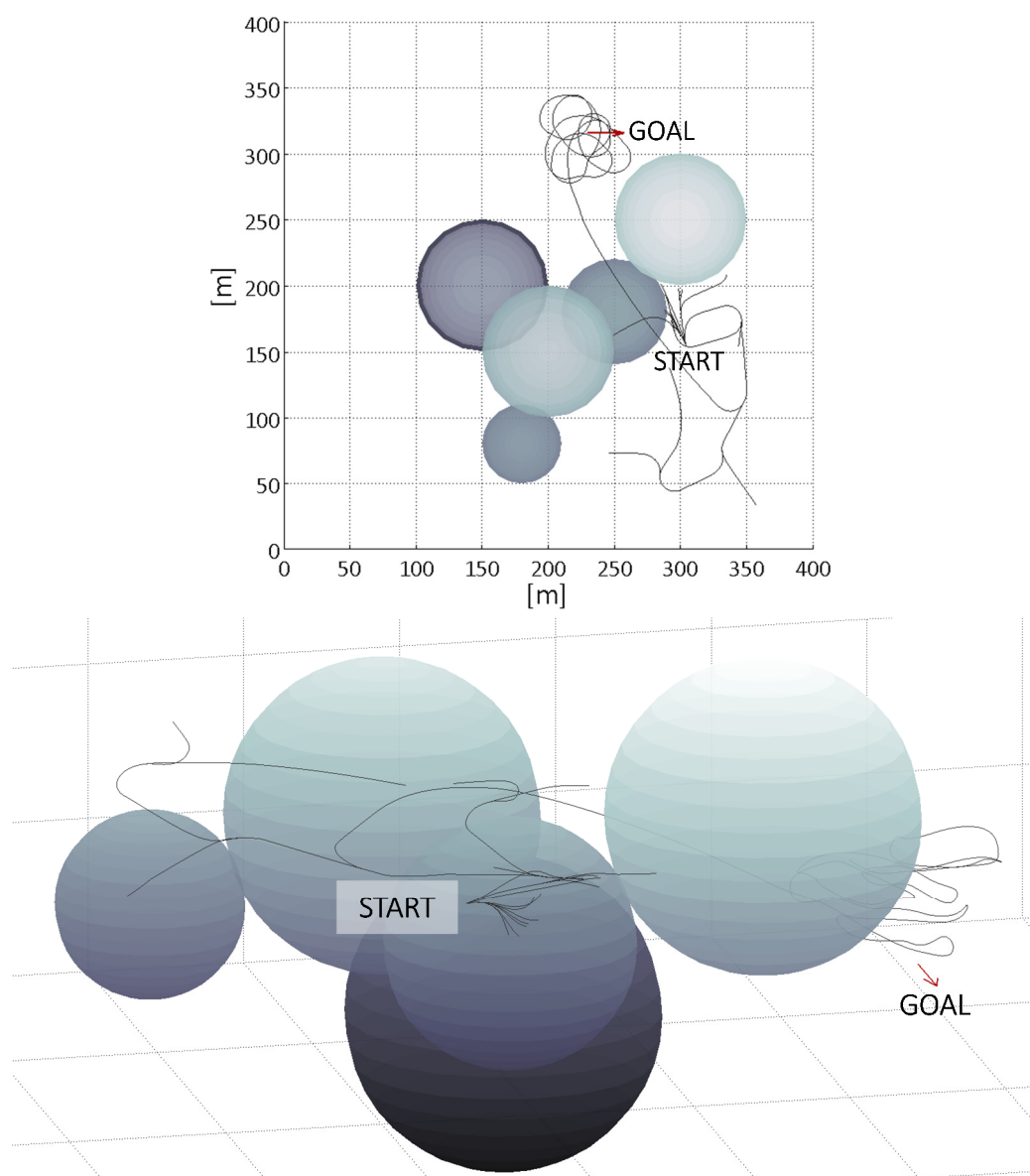


Fig. 6.4.7 Incomplete Trajectory Generation

6.4.4 リアルタイム性能の評価

本実験では、軌道プランナのリアルタイム性の評価を目的として、計算時間を記録した。パターン A は 24 ケース、パターン B は 17 ケース、パターン C は 16 ケースの軌道生成に成功しており、それらの時間計測結果の平均値を取って比較する。ただし、この時間計測は通信時間やデータロガーへの保存時間は含まず、軌道プランナの計算時間のみとしている。結果を Fig. 6.4.8 に示す。結果より、パターン A, B, C それぞれの計算時間の平均値は 0.22 秒, 0.38 秒, 0.32 秒となった。また、全ケースの平均計算時間は 0.31 秒である。これはつまり、0.31 秒毎に軌道更新が行えることを示しており、この結果から本研究の軌道プランナは、誘導の更新周期として十分高速に解が得られるアルゴリズムであることが明らかとなった。

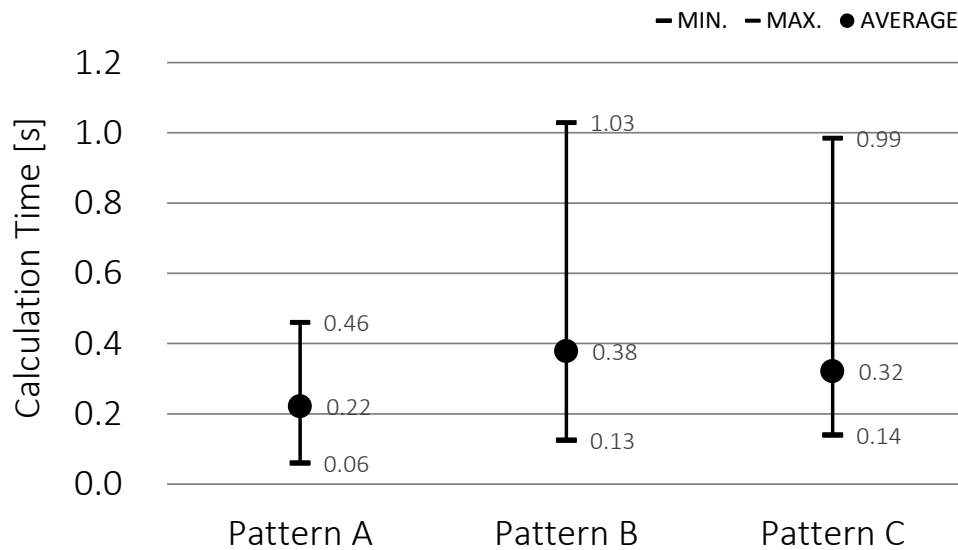


Fig. 6.4.8 Average Calculation Time for Three Obstacle Patterns

6.5 本章のまとめ

本章では、はじめに UAV を用いたリアルタイム誘導計算飛行実験に向けたシステム構成や実験条件について述べた。そして本実験では、障害物を 1 個、3 個、5 個とした 3 種類の障害物環境を想定し、実際に UAV を飛行させてデータを取得しながらリアルタイムに軌道プランナの計算を行った。結果は、全ての障害物パターンについて良好な解軌道が得られることが示された。一方で、通信不良やメモリ不足といった原因で解軌道が得られないケースも確認できた。メモリ不足に陥る要因の一つに、ゴール付近での疑似軌道による複数回の試行があり、今後アルゴリズムの改善が必要と思われる。また、本実験では軌道プランナの計算時間も計測した。結果より、全平均で 0.31 秒で解軌道の計算が行えており、リアルタイム誘導アルゴリズムとして十分に高速な手法であることが証明された。

第7章 結言

本論文では、無人航空機の衝突回避を可能とする自律誘導アルゴリズムの構築を目指した。本研究の独自性と新規性は、多数の障害物が存在する複雑な飛行環境下において、厳密な障害物回避軌道を高速に生成することである。研究手法では、従来研究の多くで用いられた軌道の最適性を追わないことにより、高速に解を得る手法に着想した。また、機体モデルに基づく数値計算により軌道を生成することで、飛行可能な滑らかな軌道を獲得することができ、さらに制御コマンドを同時に生成可能な手法を提案した。以下に各章のまとめを記載する。

第1章では、研究背景と目的について述べた。本研究は、九州工業大学が参画する火星探査航空機と有翼ロケット実験機の2つのプロジェクトにおいて実用的なアルゴリズムの構築を目指した。火星探査航空機は、火星環境において100kmの滑空飛行探査を行う機体であり、有翼ロケット実験機は、将来的に高度100kmまで打上げられて弾道飛行の後、滑空飛行により着地点に帰還するシステムである。

第2章では、障害物回避経路探索法として知られる様々な手法の特徴をシミュレーションにより確認し、それらの特徴を比較検討した。検討結果より、ランダム探索アルゴリズムの一種であるRRTは、障害物環境に依らず探索領域を均一に探索できることが分かり、他の手法と比較して探索に失敗しにくい特徴があることが分かった。そこでRRTを本研究の経路探索法として採用し、航空宇宙機の誘導に適したアルゴリズムへと改修した。また、障害物を段階的に増加させた場合の探索シミュレーションを行い、障害物の増加にも対応できるアルゴリズムであることを実証した。

第3章は、力学フィルタの手法を研究した。ランダム探索で与えられる指令値へ機体の運動を追従させるため、逆運動学を用いた手法を構築した。そして、力学フィルタのシミュレーションにより滑らかな軌道が得られることを確認した。また、ランダム探索と力学フィルタを組み合わせた基礎的な軌道プランナを構築した。さらに、軌道生成シミュレーションを行った。結果として障害物を増加させても衝突を回避する軌道が描けるケースもあったが、衝突する事象や目的地までに誤誘導が生じるなど、基礎的な軌道プランナは多くの課題が明らかとなった。

第4章では、前章で明らかとなった課題を克服することができる新たな手法を提案した。同手法では、ランダム探索に力学フィルタを組み込み、軌道を描きながらランダムに探索を実行する融合型軌道プランナを構築した。また、融合型軌道プランナを用いた

軌道生成シミュレーションを火星探査航空機と有翼ロケット実験機の両モデルにおいて行い、障害物が多数存在する複雑な飛行環境下でも良好な探索結果が得られることが確認できた。

第 5 章では、融合型軌道プランナで生成された軌道を用いた 6 自由度シミュレーションによりリアルタイム誘導の必要性について検討した。軌道更新を行わない場合には、モデル誤差や制御の不完全性などの影響により目標軌道から逸脱し、障害物に衝突する軌道となった。一方で、軌道更新を一定周期で行うことにより目的地まで良好に誘導飛行することに成功し、自律飛行システムにおける軌道プランナのリアルタイム誘導の有効性が実証された。

第 6 章は、UAV を用いたリアルタイム誘導計算飛行実験を行った。UAV へ実装した誘導用コンピュータにより軌道プランナをオンボードで計算した。結果として、障害物数を変化させた 3 つの障害物配置の全てにおいて解軌道を得ることに成功した。また、軌道プランナの計算時間も計測し、平均 0.31 秒で探索が完了することを確認でき、リアルタイム誘導アルゴリズムとして十分に高速な手法であることが証明された。

今後は UAV の完全自律飛行システムを構築し、実用化を目指した軌道プランナによるリアルタイム誘導飛行実験を行うことが必要である。また、障害物を感知するセンシングシステムとの統合も本研究の重要な課題の一つである。

謝辞

本研究を遂行し論文を執筆するにあたり、熱心なご指導、ご鞭撻を頂いた九州工業大学 米本浩一教授に心より感謝を申し上げます。米本教授には、火星探査航空機や有翼ロケット実験機といった世界でも類を見ない壮大なプロジェクトに関わらせて頂き、将来それらに適用可能な夢のある研究テーマを与えて頂きました。また、大変お忙しいなかでも昼夜を問わずご相談に乗って頂き、膨大な知識と経験に裏付けされた多くの的確なご助言をいただきました。ここに謹んで感謝の意を表します。

論文をまとめるにあたり、貴重なご意見、ご指導を頂きました九州工業大学 赤星保浩教授、相良慎一准教授、浅海賢一准教授ならびに宇宙航空研究開発機構（JAXA）石本真二様に心より感謝を申し上げます。

2014 年度まで、九州工業大学宇宙システム研究室の助教としてご指導をいただきました松本剛明様には、研究テーマの着想に至るきっかけや研究指針など、多岐にわたる有益なご指導を頂きました。心より感謝を申し上げます。

株式会社ゼノクロス 満武勝嗣様には、小型無人飛行機の開発において、制御装置や監視ソフトの開発などを担って頂き、また本研究においても、ご自身の研究経験に基づき様々なアドバイスを頂きました。心より感謝を申し上げます。

宇宙システム研究室の先輩、同輩、後輩には、ゼミや普段の研究生活でたくさんのご意見を頂き、ご相談に乗って頂きました。心より感謝いたします。

本研究は、日本学術振興会 特別研究員奨励費 267733 の助成を頂きました。謹んでお礼申し上げます。

最後に、研究生生活を陰ながら支えてくれた家族、友人に心より感謝いたします。

参考文献

- [1] http://www.iadf.or.jp/8361/LIBRARY/MEDIA/H24_dokojyoho/24-3.pdf
- [2] http://www.cradleofaviation.org/history/permanent_exhibits/world_war_i/curtiss_sperry_aerial_torpedo.html
- [3] 平成 17 年度無人機（UAV）の汎用化に伴う防衛機器産業への影響調査報告書，社団法人 日本機械工業連合会 日本戦略研究フォーラム，2006.
- [4] <http://www.wmof.com/firebee.html>
- [5] <http://airandspace.si.edu/collections/artifact.cfm?id=A20000794000>
- [6] <http://www.northropgrumman.com/capabilities/globalhawk/Pages/default.aspx>
- [7] <http://www.airforce-technology.com/projects/predator-uav/>
- [8] <http://n-seikei.jp/2015/02/post-26879.html>
- [9] <http://www.aero.jaxa.jp/research/frontier/sst/d-send.html>
- [10] <http://www.news2u.net/releases/107520>
- [11] <http://itpro.nikkeibp.co.jp/article/NEWS/20131203/522103/?ST=management&rt=ocnt>
- [12] http://mitsui.mgssi.com/issues/report/r1410t_yano.pdf
- [13] <http://www.cnn.co.jp/usa/35064567.html>
- [14] http://www.nikkei.com/article/DGXLASDG22H5G_S5A420C1CC0000/
- [15] http://www.tokiorisk.co.jp/risk_info/up_file/201504241.pdf
- [16] http://www.nikkei.com/article/DGXLASFS02H48_S5A600C1PP8000/
- [17] <http://nssdc.gsfc.nasa.gov/nmc/spacecraftDisplay.do?id=1971-049A>
- [18] <http://nssdc.gsfc.nasa.gov/planetary/viking.html>
- [19] <http://nssdc.gsfc.nasa.gov/nmc/spacecraftDisplay.do?id=1996-062A>
- [20] <http://nssdc.gsfc.nasa.gov/planetary/mesur.html>
- [21] <http://www.stp.isas.jaxa.jp/nozomi/>
- [22] <http://flab.eng.isas.jaxa.jp/meav/>
- [23] http://www.um.u-tokyo.ac.jp/web_museum/ouroboros/v15n1/v15n1_sato.html
- [24] <http://marsairplane.larc.nasa.gov/>
- [25] 大山聖，米本浩一，竹内伸介，得竹浩，永井大樹，砂田茂，大槻真嗣：火星探査飛行機の多目的設計探査，第 55 回宇宙科学技術連合講演会，愛媛，2011.
- [26] 竹内伸介，野坂拓矢，佐々木岳，岩田稔，松本剛明，米本浩一：火星探査航空機用超軽量翼構造の開発，第 57 回宇宙科学技術連合講演会，鳥取，2013.

- [27] 玉井至, 齋藤秀全, 佐々木岳, 松本剛明, 米本浩一: 極低レイノルズ数における2次元翼周りの可視化と翼特性の相関, 日本機械学会九州支部講演論文集, 第2014巻, 803号, 2014.
- [28] 佐々木岳: 火星探査航空機主翼翼型の多目的最適設計探索とクラスタ分析を用いたデータマイニング, 博士論文, 九州工業大学, 2015.
- [29] <http://natgeo.nikkeibp.co.jp/nng/article/news/14/5910/>
- [30] http://www.metsoc.jp/tenki/pdf/2013/2013_05_0063.pdf
- [31] 井上博夏, 高村英雅, 新井健太郎, 小野雅裕, 足立修一: 画像処理と測位信号による火星飛行機の自己位置推定, 第58回宇宙科学技術連合講演会, 長崎, 2014.
- [32] 濱田吉郎, 佐藤昌之, 齊藤健一, 中山空星: ドップラーライダーを用いた突風応答軽減制御系のシミュレーションソフトウェア, 第52回飛行機シンポジウム, 長崎, 2014.
- [33] <http://www.technologyreview.com/article/424586/was-the-space-shuttle-a-mistake/page/2/>
- [34] <http://www.nasa.gov/exploration/systems/orion/index.html>
- [35] <http://www.scaled.com/projects/tierone/>
- [36] 浅田正一郎, 森健, 山尾裕之, 若松逸雄: Hope-X 開発から将来宇宙輸送系に向けて, 三菱重工技報, Vol. 39, No.1, pp. 16-19, 2012.
- [37] http://www.jaxa.jp/library/space_law/chapter_4/4-1-1-4/4-1-1-42_j.html
- [38] http://www.jaxa.jp/projects/rockets/orex/index_j.html
- [39] http://www.jaxa.jp/projects/rockets/hyflex/index_j.html
- [40] http://www.jaxa.jp/projects/rockets/alflex/index_j.html
- [41] http://www.soranokai.jp/pages/HOPE_history.html
- [42] 長友信人, 稲谷芳文, 成尾芳博, 川口淳一郎: HIMES 飛翔体構想, 日本航空宇宙学会誌, 第39巻, 第454号, pp. 7-11, 1991.
- [43] ISAS ニュース, 宇宙科学研究所, No. 91, pp. 4, 1988.
- [44] ISAS ニュース, 宇宙科学研究所, No. 132, pp. 4, 1992.
- [45] <http://www.hastic.jp/index130812.htm>
- [46] Arthur, R. and Jonathan P. H. "Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Programming," American Control Conference, Vol. 3, pp. 1936-1941, 2002.
- [47] Miyamoto, S., Narumi, T., Matsumoto, T. and Yonemoto, K. "Real-time Guidance of Reusable Winged Space Plane Using Genetic Algorithm Implemented on Field Programmable Gate Array," IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 349-354, 2012.

- [48] 宮本信太郎：母集団を分割統合する遺伝的アルゴリズムの提案とその応用によるリアルタイム最適軌道生成，博士論文，九州工業大学，2014.
- [49] Doebbler, J., Gesting, P. and Valasek, J. “Real-Time Path Planning and Terrain Obstacle Avoidance for General Aviation Aircraft,” AIAA Guidance, Navigation, and Control Conference and Exhibit, California, 2005.
- [50] Hrabec, S. “3D Path Planning and Stereo-based Obstacle Avoidance for Rotorcraft UAVs,” IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, 2008.
- [51] 満武勝嗣：無人航空機のための障害物回避を考慮したリアルタイム経路生成法に関する研究，博士論文，九州大学，2012.
- [52] Dijkstra, E. W. “A Note on Two Problems in Connexion with Graph,” *Numerische Mathematik* 1, pp. 269-271, 1959.
- [53] Hart, P. E. “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, pp. 100-107, 1968.
- [54] 太田順，新井民夫，倉林大輔：知能ロボット入門—動作計画問題の解法，コロナ社，2000.
- [55] Kavraki, L. E., Svestka, P., Latombe, J. C. and Overmars, M. H., “Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces,” *IEEE Transactions on Robotics and Automation*, Vol. 12, pp. 566-580, 1996.
- [56] LaValle, S. M., “Rapidly-Exploring Random Trees: A new tool for path planning,” Department of Computer Science, Iowa State University, TR 98-11, 1998.
- [57] Matsumoto M., and Nishimura T., “Mersenne twister: a 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator,” *ACM Transactions on Modeling and Computer Simulation - Special issue on uniform random number generation*, Vol.8, pp. 3-30, 1998.
- [58] 山根克，中村 仁彦：力学フィルタによるヒューマノイドの運動生成-HRP 仮想プラットフォームへの適用-，第 17 回日本ロボット学会学術講演会予稿集，vol. 3, pp. 1213-1214, 1999.
- [59] Nakamura, Y. and Hanafusa, H., “Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control,” *Journal of Dynamic Systems, Measurement, and Control*, Vol. 108, pp. 163-171, 1986.
- [60] Farin, G., Hoschek, J. and Kim, M. S., “Handbook of Computer Aided Geometric Design,” *Elsevier Science B. V.*, 2002.
- [61] https://www.encyclopediaofmath.org/index.php/Bernstein_polynomials
- [62] 加藤寛一郎，大屋昭男，柄沢研治：航空機力学入門，東京大学出版会，2012.
- [63] <http://d.hatena.ne.jp/ina111/20100828/1282977773>

付録

以下に，本論文の 3.4 節で述べた力学フィルタを改善する式の構成を記載する．

まず，状態ベクトル \mathbf{x} ，入力ベクトル \mathbf{u} をそれぞれ以下のように定義する．

$$\mathbf{x} = \begin{Bmatrix} V \\ \gamma \\ \psi \\ T \\ \alpha \\ \phi \end{Bmatrix} \quad (\text{a.1})$$

$$\mathbf{u} = \begin{Bmatrix} \dot{T} \\ \dot{\alpha} \\ \dot{\phi} \end{Bmatrix} \quad (\text{a.2})$$

また，関数ベクトル $\mathbf{f}(\mathbf{x})$ は以下のように定める．

$$\mathbf{f}(\mathbf{x}) = \begin{Bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ f_3(\mathbf{x}) \\ 0 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} \frac{1}{m} \left(T \cos \alpha - \frac{1}{2} \rho V^2 S C_D - mg \sin \gamma \right) \\ \frac{1}{mV} \left(T \sin \alpha + \frac{1}{2} \rho V^2 S C_L \cos \phi - mg \cos \gamma \right) \\ \frac{1}{mV \cos \gamma} \left(\frac{1}{2} \rho V^2 S C_L \sin \phi \right) \\ 0 \\ 0 \\ 0 \end{Bmatrix} \quad (\text{a.3})$$

このとき，質点系の運動方程式は，

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{A}\mathbf{u} \quad (\text{a.4})$$

と表せる．ただし

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{a.5})$$

である．

ここで、観測ベクトル \mathbf{y} を

$$\mathbf{y} = \begin{Bmatrix} V \\ \gamma \\ \psi \end{Bmatrix} \quad (\text{a.6})$$

と定めると、 \mathbf{x} を用いて以下のように表せる.

$$\mathbf{y} = \mathbf{B}\mathbf{x} \quad (\text{a.7})$$

ただし,

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (\text{a.8})$$

である. このとき, Eq. (a.7)を微分し, Eq. (a.4)を代入すると

$$\begin{aligned} \dot{\mathbf{y}} &= \mathbf{B}\dot{\mathbf{x}} \\ &= \mathbf{B}\{\mathbf{f}(\mathbf{x}) + \mathbf{A}\mathbf{u}\} \\ &= \mathbf{B}\mathbf{f}(\mathbf{x}) + \mathbf{B}\mathbf{A}\mathbf{u} \end{aligned} \quad (\text{a.9})$$

となる. ここでさらに微分すると

$$\ddot{\mathbf{y}} = \mathbf{B}\dot{\mathbf{f}}(\mathbf{x}) + \mathbf{B}\mathbf{A}\dot{\mathbf{u}} \quad (\text{a.10})$$

であるが, $\mathbf{B}\mathbf{A} = \mathbf{0}$ より

$$\ddot{\mathbf{y}} = \mathbf{B}\dot{\mathbf{f}}(\mathbf{x}) \quad (\text{a.11})$$

と書ける. ここで $\dot{\mathbf{f}}(\mathbf{x})$ は

$$\begin{aligned} \dot{\mathbf{f}}(\mathbf{x}) &= \frac{d\mathbf{f}(\mathbf{x})}{dt} \\ &= \frac{d\mathbf{f}(\mathbf{x})}{d\mathbf{x}} \frac{d\mathbf{x}}{dt} \\ &= \mathbf{J}_f \dot{\mathbf{x}} \end{aligned} \quad (\text{a.12})$$

である. ただし, ヤコビアン \mathbf{J}_f は

$$J_f = \begin{bmatrix} \frac{\partial f_1}{\partial V} & \cdots & \frac{\partial f_1}{\partial \phi} \\ \frac{\partial f_2}{\partial V} & \cdots & \frac{\partial f_2}{\partial \phi} \\ \frac{\partial f_3}{\partial V} & \cdots & \frac{\partial f_3}{\partial \phi} \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{bmatrix} \quad (\text{a.13})$$

よって Eq. (a.11)は

$$\begin{aligned} \ddot{\mathbf{y}} &= \mathbf{B} \frac{d\mathbf{f}(\mathbf{x})}{d\mathbf{x}} \dot{\mathbf{x}} \\ &= \mathbf{B} \mathbf{J}_f \dot{\mathbf{x}} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial f_1}{\partial V} & \cdots & \frac{\partial f_1}{\partial \phi} \\ \frac{\partial f_2}{\partial V} & \cdots & \frac{\partial f_2}{\partial \phi} \\ \frac{\partial f_3}{\partial V} & \cdots & \frac{\partial f_3}{\partial \phi} \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\gamma} \\ \dot{\psi} \\ \dot{T} \\ \dot{\alpha} \\ \dot{\phi} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial V} & \frac{\partial f_1(\mathbf{x})}{\partial \gamma} & \frac{\partial f_1(\mathbf{x})}{\partial \psi} & \frac{\partial f_1(\mathbf{x})}{\partial T} & \frac{\partial f_1(\mathbf{x})}{\partial \alpha} & \frac{\partial f_1(\mathbf{x})}{\partial \phi} \\ \frac{\partial f_2(\mathbf{x})}{\partial V} & \frac{\partial f_2(\mathbf{x})}{\partial \gamma} & \frac{\partial f_2(\mathbf{x})}{\partial \psi} & \frac{\partial f_2(\mathbf{x})}{\partial T} & \frac{\partial f_2(\mathbf{x})}{\partial \alpha} & \frac{\partial f_2(\mathbf{x})}{\partial \phi} \\ \frac{\partial f_3(\mathbf{x})}{\partial V} & \frac{\partial f_3(\mathbf{x})}{\partial \gamma} & \frac{\partial f_3(\mathbf{x})}{\partial \psi} & \frac{\partial f_3(\mathbf{x})}{\partial T} & \frac{\partial f_3(\mathbf{x})}{\partial \alpha} & \frac{\partial f_3(\mathbf{x})}{\partial \phi} \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\gamma} \\ \dot{\psi} \\ \dot{T} \\ \dot{\alpha} \\ \dot{\phi} \end{bmatrix} \\ &= \mathbf{J}_1 \dot{\mathbf{y}} + \mathbf{J}_2 \mathbf{u} \end{aligned} \quad (\text{a.14})$$

ただし、ヤコビアン \mathbf{J}_1 , \mathbf{J}_2 は

$$\mathbf{J}_1 = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial V} & \frac{\partial f_1(\mathbf{x})}{\partial \gamma} & \frac{\partial f_1(\mathbf{x})}{\partial \psi} \\ \frac{\partial f_2(\mathbf{x})}{\partial V} & \frac{\partial f_2(\mathbf{x})}{\partial \gamma} & \frac{\partial f_2(\mathbf{x})}{\partial \psi} \\ \frac{\partial f_3(\mathbf{x})}{\partial V} & \frac{\partial f_3(\mathbf{x})}{\partial \gamma} & \frac{\partial f_3(\mathbf{x})}{\partial \psi} \end{bmatrix} \quad (\text{a.15})$$

$$J_2 = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial T} & \frac{\partial f_1(\mathbf{x})}{\partial \alpha} & \frac{\partial f_1(\mathbf{x})}{\partial \phi} \\ \frac{\partial f_2(\mathbf{x})}{\partial T} & \frac{\partial f_2(\mathbf{x})}{\partial \alpha} & \frac{\partial f_2(\mathbf{x})}{\partial \phi} \\ \frac{\partial f_3(\mathbf{x})}{\partial T} & \frac{\partial f_3(\mathbf{x})}{\partial \alpha} & \frac{\partial f_3(\mathbf{x})}{\partial \phi} \end{bmatrix} \quad (\text{a.16})$$

となる．ただし， J_1 ， J_2 を計算すると以下のとおりである．

$$J_1 = \begin{bmatrix} \frac{\rho V S C_D}{m} & -g \cos \gamma & 0 \\ \frac{\rho S C_L \cos \phi}{2m} & \frac{g \sin \gamma}{V} & 0 \\ \frac{\rho S C_L \sin \phi}{2m \cos \gamma} & -\frac{\rho V S C_L \sin \phi}{2m \sqrt{1 - \gamma^2}} & 0 \end{bmatrix}$$

$$J_2 = \begin{bmatrix} \frac{\cos \alpha}{m} & -\frac{1}{m} \left\{ T \sin \alpha + \frac{1}{2} \rho V^2 S_w (c_{D2} + 2c_{D3} \alpha) \right\} \\ \frac{\sin \alpha}{mV} & \frac{1}{m} \left\{ T \cos \alpha + \frac{1}{2} \rho V S_w c_{L2} \cos \phi \right\} \\ 0 & \frac{\rho V S_w c_{L2} \sin \phi}{2m \cos \gamma} \\ & 0 \\ & -\frac{1}{mV} \left\{ T \sin \alpha + \frac{1}{2} \rho V^2 S_w (c_{L1} + c_{L2} \alpha) \sin \phi \right\} \\ & \frac{\rho V S_w (c_{L1} + c_{L2} \alpha) \cos \phi}{2m \cos \gamma} \end{bmatrix} \quad (\text{a.17})$$

以上より Eq. (a.14)で入力 \mathbf{u} が現れるので， \mathbf{u} を逆算できるようになる．

$$\mathbf{u} = J_2^{-1}(\ddot{\mathbf{y}} - J_1 \dot{\mathbf{y}}) \quad (\text{a.18})$$

ただし，Eq. (a.18)の $\ddot{\mathbf{y}}$ には，設計された制御器が導入される．この構成では，3.4節で示した力学フィルタのように省略した項は無く，入力 \mathbf{u} を積分する必要もない．今後は，この構成を新たな力学フィルタとして検討する必要がある．