

情報特別演習 I

最終レポート

「エージェントシミュレーションによる旅客機運用のモデル化」

提出日：2019 年-月-日

提出期限：2019 年 2 月 8 日

学籍番号：201711388

所属：情報学群情報科学類

氏名：西山 大輝

クラス：3 組

1. 演習の目的と方法

1.1. 目的

近年、世間の国際化や観光産業の発達などにより公共交通機関の需要が大きく高まっている。中でも早く目的地に行くことができる飛行機は極めて重宝されている。そのため旅客機の就航数は年々増加傾向にあり、例えば羽田空港の年間飛行機発着数は 224,707 回¹、つまり 1 日あたり約 616 回と、極めて膨大な数となっている。これらすべての飛行機を運用・管理するのは管制塔にいる管制官である。特に夕方の時間帯は着陸機が、昼前の時間帯は離陸機が多くなるため空港周辺が過密になり、管制官は大変多忙になってしまう。また、2001 年には日本航空機駿河湾上空ニアミス事故という管制に起因する事故も発生している。

そこで私は、すべての飛行機が自律的に高度や角度、速度等を調整し、最終的に目的地へ着陸するシステムによって、「管制塔が無い」状態で飛行機を運用できるのではないかと考えた。そのようにすることによって、管制側の人為的なミスを 0 に近づけることができ、前述のような事故の防止や航空機の混雑時における効率的な運用を実現できると考えられ、これを本演習の目的とする。

1.2. 方法

方法として、自律的に航空機が動作し着陸に至ることができるアルゴリズムを考え、モデル化し、マルチエージェントシミュレーションで着陸機の振る舞いを確認することにした。

1.2.1. マルチエージェントシミュレーション

マルチエージェントシミュレーションとは、複数のエージェントに同時進行的に各々のルールの下、互いに作用を受けながら実行させるシミュレーションのことを言う。またエージェントは、自分の周囲の状況を認識しそれに基づいて一定のルールのもとで自律的に行動する主体のことであると定義される。ⁱ 本演習の場合、エージェントが航空機、ルールが航空路や航空機同士の間隔などに相当する。図 1 にマルチエージェントシミュレーションの概要図を示す。

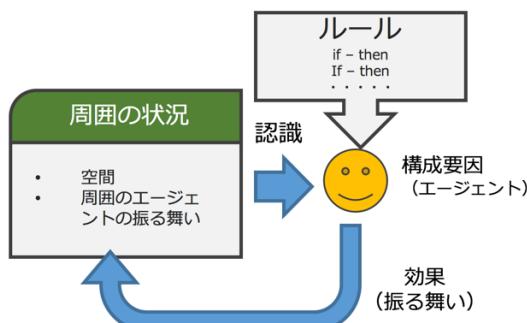


図 1 マルチエージェントシミュレーションの概要図²

¹ 2016 年の年間データ、国土交通省平成 28 年空港管理状況調書より

² 画像：構造計画研究所ホームページ(<http://mas.kke.co.jp/modules/tinyd4/index.php?id=16>)より

1.2.2. エージェントとしての航空機

航空機をエージェントとして捉えるために、周囲から認識する情報と決定する情報を次の表1のようにまとめた。

周囲から認識する情報	決定する情報
<ul style="list-style-type: none">自分の座標空港の風向き上空待機(HOLD)命令次に目指すチェックポイントの位置制限速度制限高度周囲の飛行機の位置	<ul style="list-style-type: none">チェックポイントの更新新たな速度新たな高度新たな角度新たな座標

表1 エージェントとしての航空機を取り巻く情報

表1の左側の情報から右側の情報を決定する際には、先に述べた通りルールが必要である。このルールは現実に即したものを利用することとし、これは2章で述べる航空路や風向きによる影響、航空機間で必要な間隔などに相当する。

2. 調査

第1章2節2項で取り上げた情報やルールなどは、実際の航空機運用においてどのようなものであるか、もしくはどのようなものに相当するかを調べた。この章ではこれについて述べる。

2.1. 航空路

航空機は無作為に飛行しているわけではない。航空機が飛行していく方向や飛行する高度を決定し、安全に航行できるようにした空の道、つまり航空路に従う。この航空路には次の4種類がある。

- ENROUTE：空港間を飛行する航空機の既定ルート
- DEPARTURE：離陸から ENROUTEまでのルート
- ARRIVAL：ENROUTEを飛行してきた航空機が着陸滑走路進入地点まで誘導されるルート
- APPROACH：着陸滑走路進入地点まで誘導された航空機が高度を下げつつ滑走路に着陸するルート

離陸した飛行機は DEPARTURE、ENROUTE、ARRIVAL、APPROACH の順にルートを飛行していくことで目的の空港に着陸する。

本演習では自律的な動きで着陸させることが目標であるので、以下の項では DEPARTURE 以外のルートについて解説する。

2.1.1. ENROUTE

ENROUTEは、空港間を飛行する航空機の規定のルートのことである。図2に日本周辺のENROUTEを表したENROUTE CHARTの一例を示す。

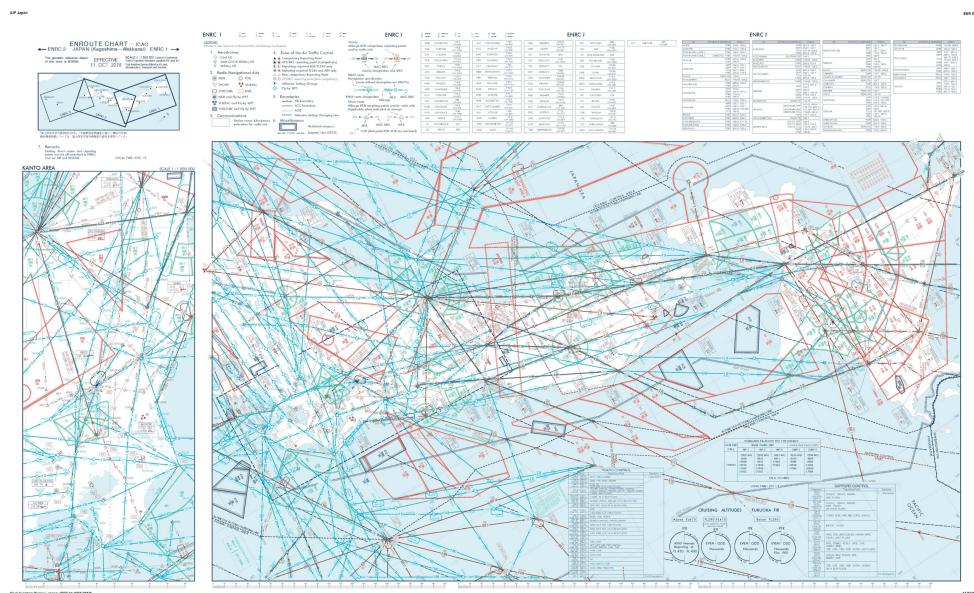


図 2 ENROUTE CHART - ICAO ENRC 1³

³ 国土交通省 AIS JAPAN, <https://aisjapan.mlit.go.jp> より

図2で水色や青色で示された線が航空機の通るルートである。ENROUTEは私たちの生活でいうところの国道や高速道路に相当すると言える。航空機は離陸してからDEPARTUREに従って航行し、次にこのENROUTEに合流し、目的の空港近くでENROUTEから離脱し、次はARRIVALに従う。

2.1.2. ARRIVAL

ARRIVALはENROUTEを飛行してきた航空機が着陸滑走路進入地点まで誘導されるルートのことである。図3に羽田空港へのARRIVALを表したARRIVAL CHARTの例を二つ示す。なお、RJTTは羽田空港を意味するICAO空港コードで、運行事業者が一般的に利用する表記である。

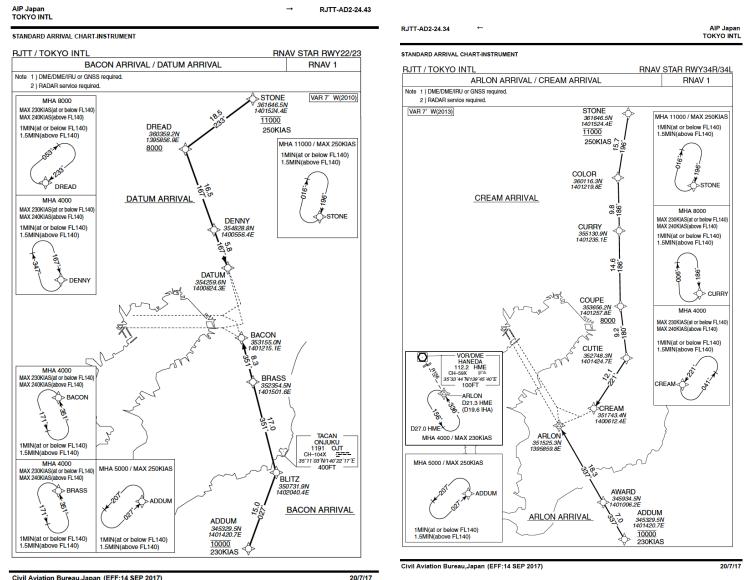


図3 STANDARD ARRIVAL CHART RJTT BACON/DATUM(左), ARLON/CREAM(右)⁴

図3には二つの例を示したが、羽田空港のARRIVALはヘリコプター専用ルートも含めて17ルート存在する。複数あるARRIVALの中でどのARRIVALを利用するかは風向きや時間帯、雲の位置などの様々な要因により、管制官によって決定される。例えば、飛行機は構造上向かい風状態で着陸するのが好ましいため、図3(左)は南風時に、図3(右)は北風時に利用される。また、図3(左)のARRIVALの南からくる航空機(BACON ARRIVAL)は人口が比較的多い千葉県千葉市上空付近を通過しているが、南風で夜間帯は騒音対策として千葉市上空を避けて東京湾を内房に沿うように飛行するBALAN ARRIVALを採択することがある。

先ほど述べた通り、ARRIVALは着陸滑走路進入地点までのルートであるが、これは図3におけるBACONやDATUM、ARLON、CREAMに相当する。この着陸滑走路進入地点から滑走路までのルートはAPPROACHに従う。

2.1.3. APPROACH

APPROACHは着陸滑走路進入地点まで誘導された航空機が高度を下げつつ滑走路に着陸するルートのことである。図4にAPPROACHを表したAPPROACH CHARTの一例を示す。

⁴ 国土交通省 AIS JAPAN, <https://aisjapan.mlit.go.jp> より

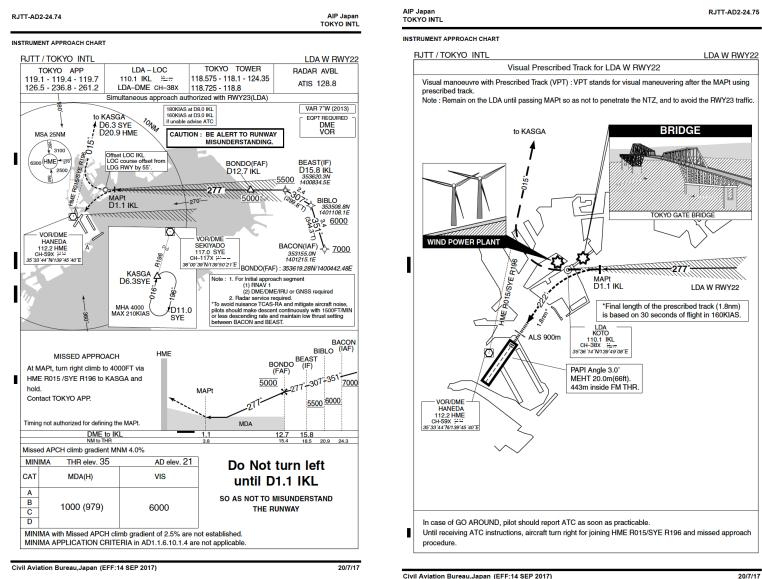


図 4 APPROACH CHART RJTT 表面(左), 裏面(右)⁵

図4のAPPROACH CHARTでは、着陸滑走路進入地点から実線に従って航行し、MAPtのポイントで右手に東京ゲートブリッジや風力発電機を目視した上で滑走路に進入することを表している。このAPPROACH CHARTの視点となる着陸滑走路進入地点はBACONであり、つまりBACON ARRIVALに続くAPPROACHであるとわかる。

2.2. HOLD

HOLDとは上空待機で航空機が旋回することである。HOLDは目的地とする空港においてゲリラ豪雨や事故などの障害が発生した時や、上空での航空機の整理が必要となった時などにされる。

HOLDにはどこでできるか、高度は何フィート以上出ないといけないか、速度は何ノット以下でなければいけないか、などといった制限がある。これはARRIVAL CHART及びAPPROACH CHARTに記されている。図3(右)で示したARLON ARRIVALのHOLDの表記部分を拡大したものを図5に示す。

⁵ 国土交通省 AIS JAPAN, <https://aisjapan.mlit.go.jp> より

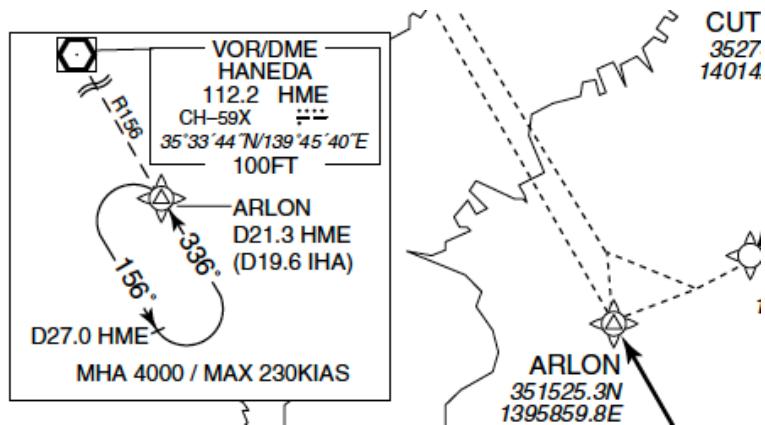


図 5 ARRON ARRIVAL の ARRON ポイントにおける HOLD 表記⁶

このように、特定のポイントで機首を向ける方向や、高度と速度の制限の下で航行すれば良いかなどの HOLD に必要な情報が記されている。HOLD 命令が出た際にはこれに従って航行する。

2.3. セパレーション

航空機同士の間には最低限確保しなければいけない間隔が存在する。これを管制間隔、セパレーションという。セパレーションには高度間隔、縦間隔、横間隔がある。それぞれで確保しなければ距離・航空時間を図 6 に示す。

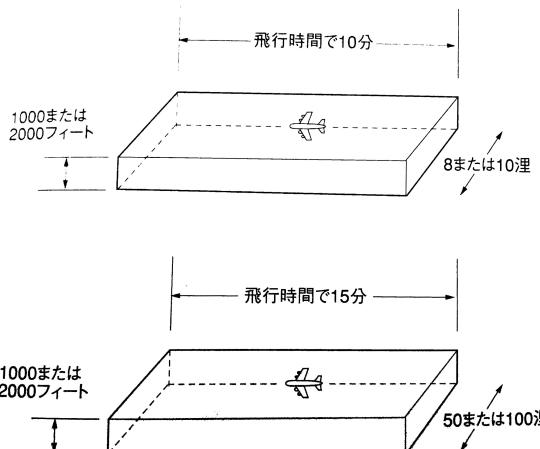


図 6 セパレーション（上）空港付近（下）それ以外⁷

航空機は最低限このセパレーションを確保しつつ航行することで安全な間隔を維持している。仮にセパレーションを過去保できなくなることが予測されたときは減速指示や下降指示などが管制官から出される。

⁶ 国土交通省 AIS JAPAN, <https://aisjapan.mlit.go.jp> より

⁷ 園山耕司（2003）「航空管制の科学」, 講談社より

2.4. 羽田空港

図7に三つのターミナルの位置と着陸機の進入方向を示した羽田空港の平面図を示す。なお方角は図の上がおよそ北東に相当する。

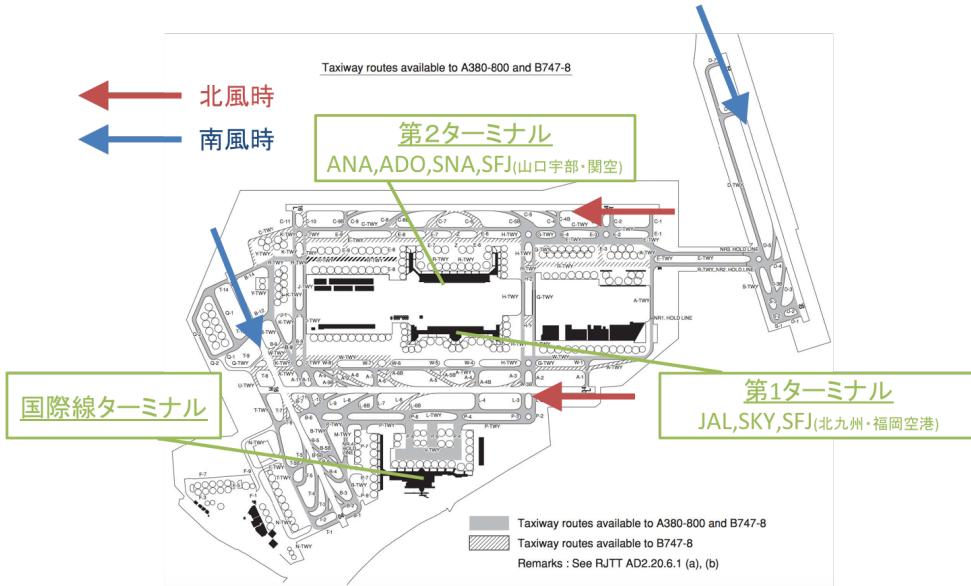


図7 羽田空港の平面図と着陸機の進入方向とターミナルの関係⁸

図7に示してある通り、羽田空港には井の字のように4本の滑走路がある。着陸機は、南風の時は青色矢印で示した方向で、北風の時は赤色矢印で示した方向でそれぞれ進入する。風向きによって使用する滑走路が異なるのは、2.1.2で述べた通り、飛行機は構造上向かい風状態で着陸するのが好ましいからである。また、羽田空港には図7にある通り国際線ターミナル、第1ターミナル、第2ターミナルの3つのターミナルがある。第1ターミナルには全日空、エア・ドゥ、ソラシドエア、山口宇部・関西空港便のスターフライヤーの各航空会社が、第2ターミナルには日本航空、スカイマーク、北九州・福岡空港便のスターフライヤーの各航空会社が、国際線ターミナルには国際線が主に搭乗口を持つ。そのため例として北風の時、全日空便がきた場合は第2ターミナルに近い図7の上の滑走路を利用することが多い。ただしこれは厳格なルールではなく、運用などの都合によって柔軟に変わる。

⁸ 国土交通省 AIS JAPAN, <https://aisjapan.mlit.go.jp> より

3. 実装

3.1. 環境・方針

2で調査した事柄をエージェントのルールとして組み込み、実装をした。

言語は C++17 で記述し、グラフィックスは OpenGL を利用して作成した。環境は MacBook Pro (13-inch, 2016, Four Thunderbolt 3 Ports)、OS は macOS Sierra ver10.12.6 である。また統合開発環境として JET BARINS の CLion 2018.2.4 を利用した。

実装の方針としては、まずプログラムのメインとなる main.cpp には状態更新のアルゴリズム(3.2 参照)を記述した。次に、main.cpp 内で必要となる関数は function.cpp に記し、メソッドに関する記述は method.cpp にそれぞれ記した。最後に、OpenGL によるグラフィックは display.cpp に記述した。また、それぞれの.cpp ファイルに対してヘッダーファイルを用意し、各.cpp ファイルで include した。

display.cpp 内では、次のような機能を実装した。

- エンターキー：一時停止、実行
- スペースキー：HOLD 命令 ON/OFF
- N キー：北風（初期で設定可能）
- S キー：南風（初期で設定可能）
- 0 キー：視点を初期に戻す
- 右クリック + マウス上下移動：拡大縮小
- 左クリック + マウス移動：視点移動

地図の画像データはフライトアウェア (<https://flightaware.com>) のものを利用し、これをテクスチャマッピングする。また飛行機の形をしたポリゴンを作成し、これを航空機のあるべき場所に表示させ、速度や便名、時刻といった視覚だけでは認知できない情報は別に表示する。

3.2. アルゴリズム

次の図 8 のようなアルゴリズムを考えた

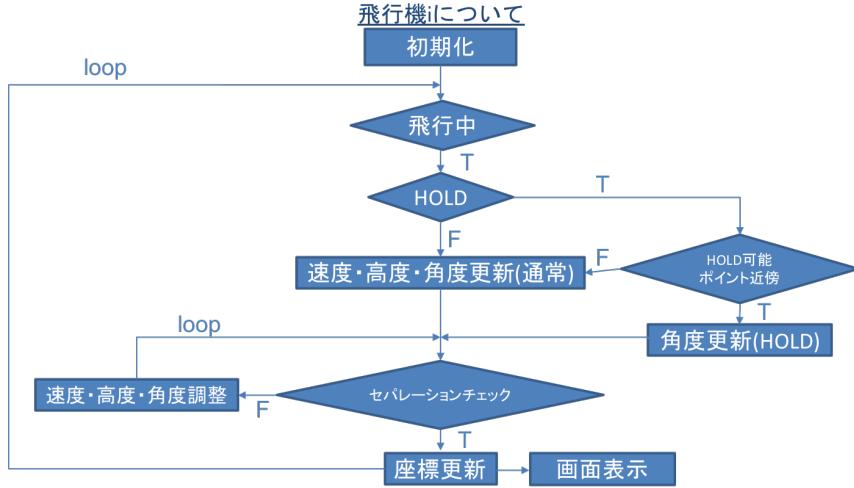


図 8 航空機の状態更新アルゴリズム

ある航空機 i について考える。まず速度などの属性を初期化する。次に飛行中で HOLD 命令がなければ、次のポイントへ向かって属性を更新する。このポイントとは ENROUTE、ARRIVAL、APPROACH のそれぞれの CHART に記されている ARRON や BACON などといったポイントのことである。そして更新された属性を元に新たな座標を決定するタイミングでセパレーションをチェックする。セパレーションが確保されていれば新たな座標を採用し、画面に表示する。一方でセパレーションが過去干されない時は、過去干されるように各属性の調整を繰り返す。また HOLD 命令が出ている時は、HOLD できるポイントの近傍にいれば速度と高度はそのまま角度を更新していき旋回を行う。HOLD できるポイントの近傍にいない場合は HOLD が可能なポイントの近傍に着くまで通常通り航行を行う。以上のことを行なう。そしてこのアルゴリズムを全ての航空機に適用する。

このアルゴリズムを main.cpp 内に記述した。

4. 結論・考察

4.1. 実行結果

4.1.1. シーン 1

2019年1月15日16:25(JST)の、羽田空港(南風)に向かう関東・東海・南東北を飛行中の飛行機をモデルにマルチエージェントシミュレーションで振る舞いを確認した。なお、その時間時点で航空機がどの位置でどういった速度・高度で航行しているかの情報はフライトレーダー24(<http://flightradar24.com>)というサイトの情報を参考とした。

まず、実行した最初の状態を拡大したものと画面中央に位置するANA680便を拡大したものを図9に示す。

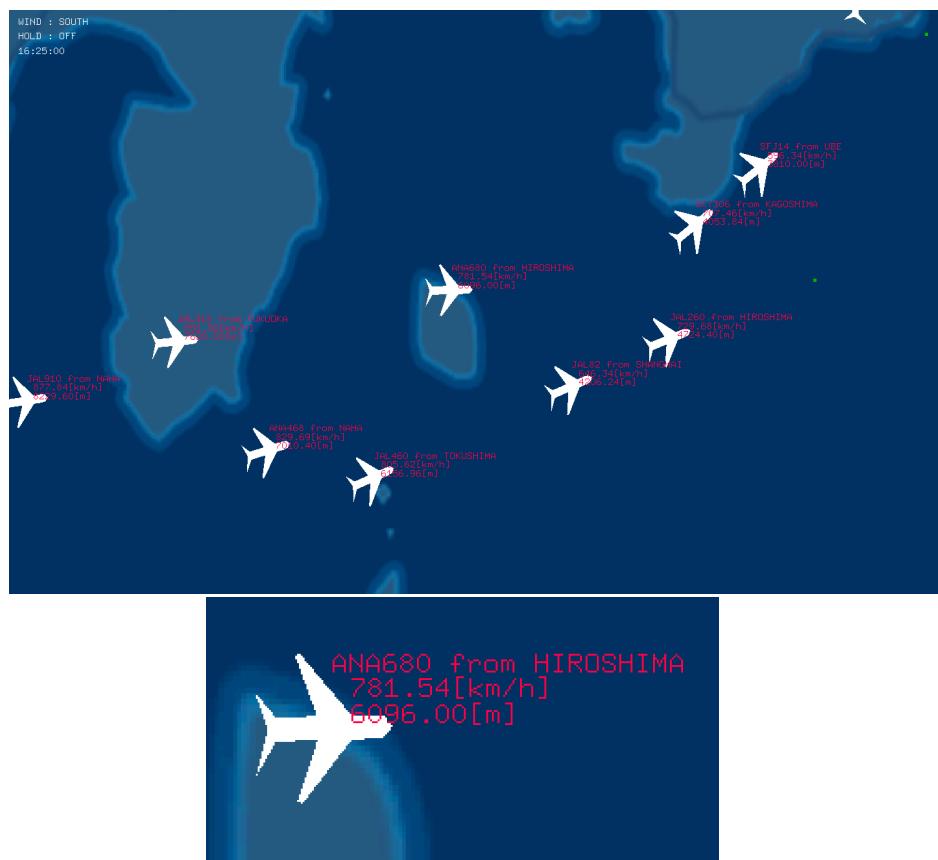


図9 実行初期状態

図9を見ると、各航空機とその右上あたりに赤文字で情報が、そして画面左上に風の状態と HOLD の状態と時間が表示されている。赤文字にはどこの航空会社の何便か、どこから来たか、時速何キロ及び高度何メートルで航行中なのが示されている。フライトレーダー24 から得た情報とはこのこととそれぞれの航空機の初期位置である。

図9の初期状態から時間を進めると図10のような結果となった。なお 16:33:13 から画面左側に表示されている文字はすでに羽田空港に到着した航空機の便名と到着時間であり、白い線は航跡である。

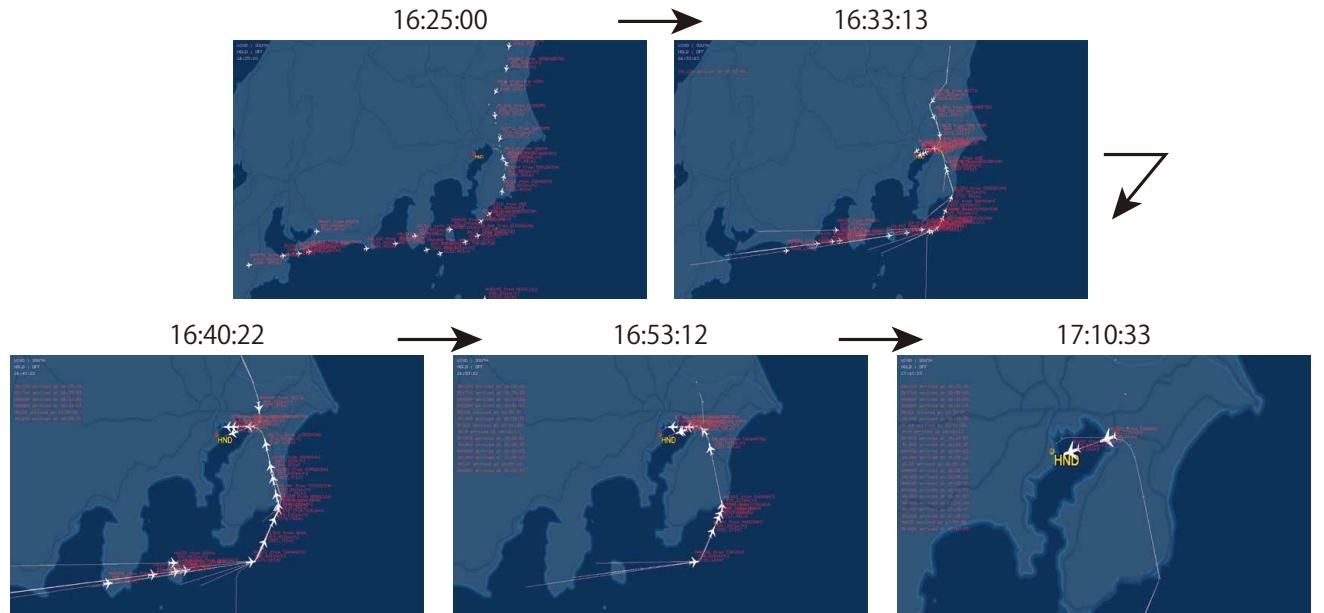


図 10 シーン 1 実行結果

この結果を見ると、時間が経つにつれて南から来た航空機が BACON ARRIVAL に従って、北から来た航空機が DATUM ARRIVAL に従ってそれぞれ羽田空港に至っていることがわかる。また、すべての航空機が羽田に至ったことからセパレーションを確保して安全な間隔を保ったまま航行していることがわかる。

4.1.2. シーン 2

次に 2019 年 1 月 13 日 17:40(JST) の、羽田空港(北風)に向かう関東・東海・南東北を飛行中の飛行機をモデルにマルチエージェントシミュレーションで振る舞いを確認した。この実行結果を図 11 に示す。

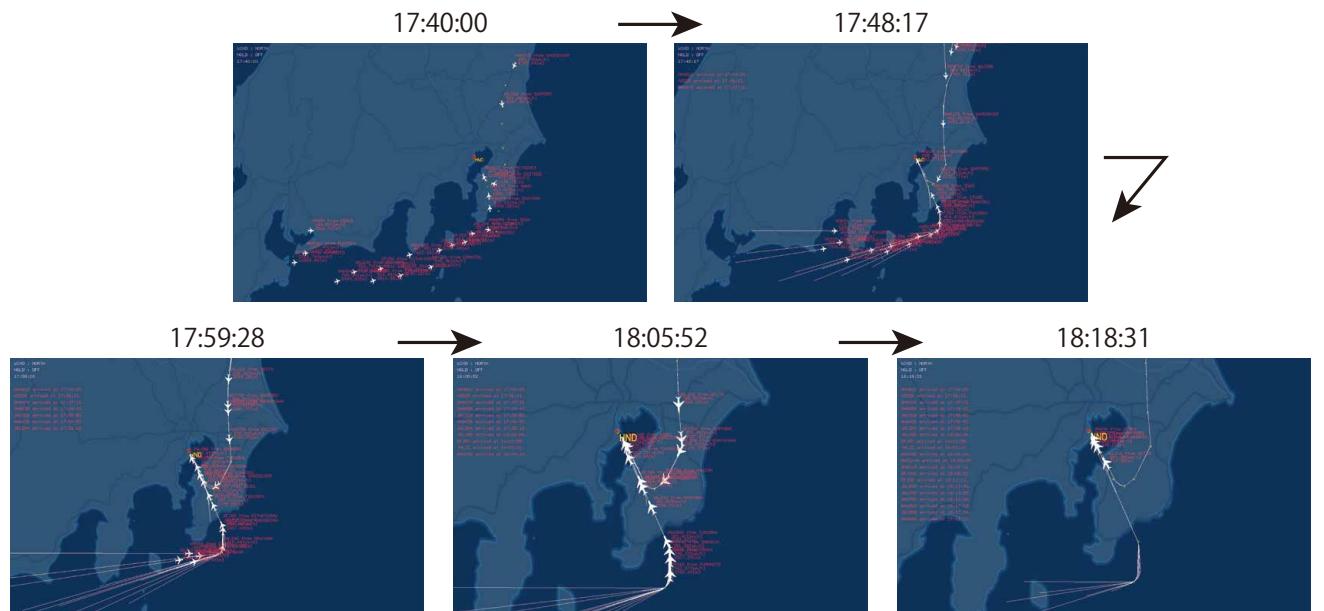


図 11 シーン 2 実行結果

この結果からも、北風の場合も同様にセパレーションを確保して航行し、羽田空港に至っていることがわかる。

4.1.3. シーン 3

最後に 2019 年 1 月 13 日 17:40(JST) の、羽田空港(北風)に向かう関東・東海・南東北を飛行中の飛行機をモデルにおいて、17:55 頃に HOLD 命令をだし、18:10 頃に HOLD 命令を解除した場合の振る舞いを確認する。実行結果を図 12 に示す。

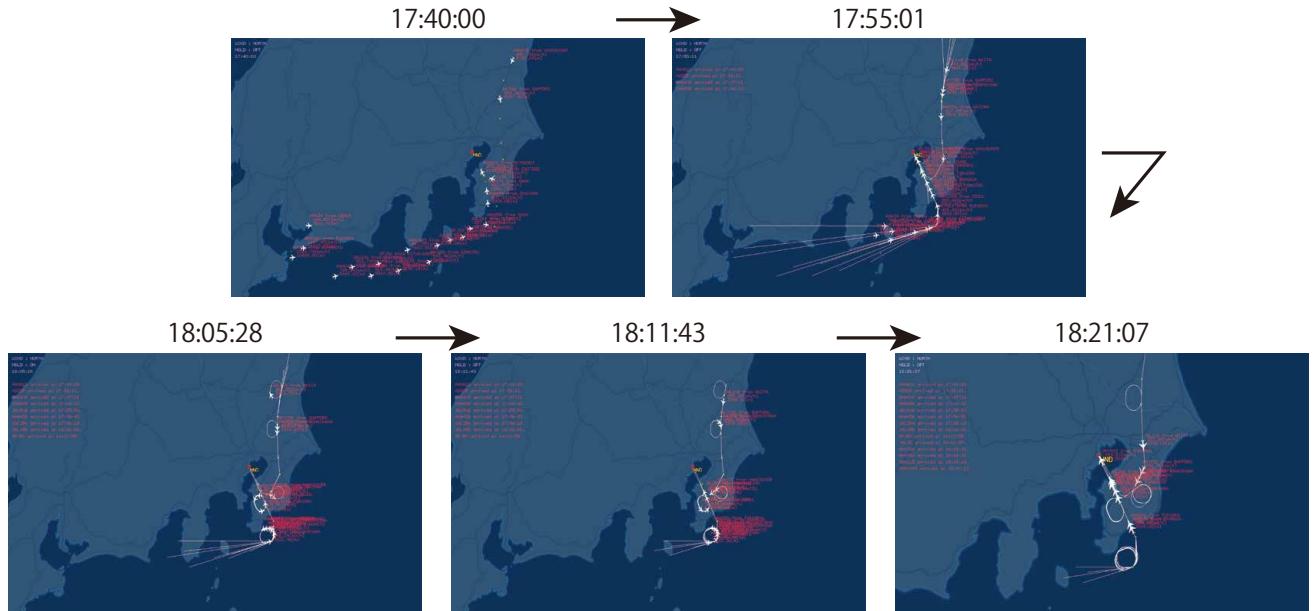


図 12 シーン 3 実行結果

この結果から、HOLD 命令が出た時、最寄りの HOLD 可能なポイントへ航行してそこで HOLD 命令がなくなるまで旋回を続け、HOLD 命令がなくなった後は最終的に羽田空港まで至っていることが確認できる。

4.2. 考察

以上の結果、マルチエージェントシミュレーションによって振る舞いを確認したところ、HOLD 命令が有無どちらの場合でも、セパレーションを確保しつつ、羽田空港に着陸できていることが確認できた。セパレーションを確保しているということは航空における安全上の問題はないため、あくまで空域では再現可能な振る舞いをシミュレーション上でできていると言える。一方、滑走路に進入する時間の間隔が極めて短く、空港側のキャパシティが考えられていない結果となっているのも事実である。

以上のことから実装したアルゴリズムは、空港に進入する時間間隔には無理があるものの、空域での振る舞いはおおよそ適切なものになっているため、目標は達成できたと考えられる。

4.3. 今後の課題

まず考察で述べた空港に進入する間隔を広げることで、空港側にとって不都合なく運用できるようにすることが挙げられる。これはアルゴリズムに修正を加えることによって実現が可能ではないかと考える。

また航空機の運用において、気象状況や航空機の遅延状況、残りの燃料などの様々な考慮していない要素がある。さらに、遅延時間や航行時間、航行距離を最適化することも可能ではないかと考える。これらを組み込んだアルゴリズムを考えることで、よりもっともらしいシミュレーション結果を望めると考えられる。

この演習では自分でアルゴリズムを考えて実装をした。そこで新たなアプローチとしてマルチエージェント強化学習ⁱⁱを取り込むことで、エージェントに目標を与え、エージェントが学習していくという方法が考えられる。これによってこの演習の結果とは別の、より良い結果が出る可能性がある。

以上のことと今後の課題としたい。

ⁱ 山影進（2007）「人工社会構築指南 artisoc によるマルチエージェント・シミュレーション入門」，書籍工房早山

ⁱⁱ 荒井 幸代, 宮崎 和光, 小林 重信 (1998) 「マルチエージェント強化学習の方法論 : Q-Learning と Profit Sharing による接近」人工知能学会誌,p609-618,4,13