Abstract

Distributed pilot decision-making plays a critical part in high volume operation in non-controlled/non-radar airports, a concept proposed by the NASA Small Aircraft Transportation System program. Realization of the concept relies on advanced cockpit systems that assist pilots in both information-processing and decision-making. In this paper, we presented the design of an onboard pilot decision aid system, called the Small Aircraft Pilot Assistant, which is dedicated to help pilots perform the high volume operation flight tasks. It increases the cockpit decision-making capacity and pilot situation awareness by automating part of the pilot decision-making process, especially in its early stages of information acquisition and analysis. The pilot tests of the system are conducted using a real-time, multi-aircraft, pilot-in-the-loop simulation system that is presently capable of middle-fidelity HVO simulation.

1 Introduction

The nation needs a new transportation system to relieve safety and congestion problems currently on highways and in the air; the nation’s highway systems are frequently plagued with delays and accidents, and the nation's hub-and-spoke airports are overwhelmed with increased air traffic, leading to frequent delays and flight cancellations [1]. With over 5,000 small airports already in place across the country, in almost every locality, a small aircraft transportation system that is both a safe and affordable alternative to current transportation systems could provide an effective solution [1].

The Small Aircraft Transportation System (SATS) Project is nation-wide project conducted through a public-private partnership including NASA, the FAA, and the National Consortium for Aviation Mobility (NCAM) SATSLabs [2]. The objective of the project is to enable expanded use of small airports and small aircraft for public transportation. Within the five year period (2001 – 2005), the project will demonstrate SATS operational capability in four major operating capability areas [2]:

1. Higher volume operations (HVO) in non-radar airspace and at non-towered airports.
2. Lower landing minimums at minimally equipped landing facilities.
3. Increase single-pilot crew safety and mission reliability.
4. En Route procedures and systems for integrated fleet operations.

The SATS HVO operational concept concerns with small airports that lack radar coverage and controlling tower facility. Its objective is to provide this type of small airports, with a minimum of infrastructure and at a low cost, the capability of higher volume operations under Instrumental Meteorological Conditions (IMC) [3]. The concept is significantly different from today’s Instrument Flight Rules (IFR) operations where the Air Traffic Control (ATC) provides separation assurance services. In the SATS HVO, pilots sustain the responsibility of safety separation between aircraft after their aircraft enter a designated aerospace volume surrounding the airport, called Self-Controlled Area (SCA): the ATC only provides separation assurance outside the SCA. This new responsibility for pilots is supported by new rules of entry, new flight procedures, and advanced onboard navigation tools [3].

Realization of the HVO concept requires advanced cockpit systems that assist pilots in managing information and decision-making. The HVO concept is base on a distributed decision-making environment within which most of the decision-making is left to pilots. Pilots are required to handle large amount of data fed from various sources. Increased information processing, as well as augmented aircraft safety responsibility, leads to demanding flight tasks and increased pilot workload. Today’s gauge-based avionics in most GA aircraft cannot match the cockpit system requirements of
the SATS. The advanced cockpit systems that currently widely used in commercial transporters and business jets, for example, PFD, MFD, and FMS, are not suitable for small aircraft due to their high cost. In this paper, we present the design of a new onboard pilot decision aid system, namely Small Aircraft Pilot Assistant (SAPA), which is a cost-effective system dedicated to the SATS aircraft conducting the HVO in an SCA. The SAPA aims at increasing the cockpit decision-making capacity by automating part of the pilot decision-making process, especially in its early stages of information acquisition and analysis.

Designing the SAPA is an extension of the research work conducted by the Texas A&M University Flight Simulation Laboratory (FSL) on the design and development of intelligent cockpit systems and pilot decision-aiding tools for GA aircraft in the last ten years. Examples of our previous pilot decision-aid tools are the General Aviation Pilot Advisor and Training System (GAPATS) [4] and the Hierarchical Agent Based System for GA Conflict Detection and Resolution (CD&R) [5]. The GAPATS is a computerized airborne advisory system. It assesses the pilot’s flying performance and issues recommendation for pilot actions in all flight phases from takeoff to landing. The Hierarchical Agent Based system is for GA aircraft in Free Flight environment. It provides pilots conflict free flight path guidance in situations where weather conflicts (thunderstorm hazards) and traffic conflicts (mid-air collisions) exist concurrently. The design experiences and the evaluation results of both systems are of great importance to the development of SAPA.

The paper is organized as follows. Section two introduces the SATS HVO operational concept. Section three describes in detail the design of the SAPA. In section four, we describe the SATS HVO simulation and system evaluation of the SAPA. Finally, section five concludes the paper.

2 THE SATS HVO Operational Concept

The SATS HVO concept is a collection of rules and procedures which, when followed, will assure safety separation between aircraft during the transition to the SCA, approach, missed-approach, landing, takeoff, departure, and transition out of the SCA[3]. The concept utilizes four main components: a designated airspace volume surrounding the airport called the Self Controlled Area (SCA); a centralized automated system called the Airport Management Module (AMM); aircraft to aircraft and aircraft to AMM data communication; and distributed, on-board navigation tools[3].

2.1 The Self Controlled Area (SCA)

The SCA (Figure 1) is normally 20 nm miles in diameter and 3000 ft above ground level (AGL) in height. The design of the SCA is similar to a GPS T approach [7]. The top view of the SCA (Figure 2) shows the fixes and segments of the arrival and departure paths of a nominal SCA design. The right and left initial arrival fixes (IAF-R, IAF-L) are also right and left missed-approach holding fixes (MAHF-R, MAHF-L). Other fixes in an SCA are intermediate fix (IF), final approach fix (FAF), missed-approach point (MAP), and right and left departure fixes (DF-R, DF-L).

Figure 1. Self Controlled Area airspace volume

Figure 2. Top view of the SCA
2.2 The Airport Management Module (AMM)

The AMM is the arbiter and sequencer of the SCA. It is an automated software system residing at the airport ground. It receives state information from aircraft in the vicinity of the airport via ADS-B and communicates with aircraft via digital data link. Nevertheless, it is not designed to replace the traditional ATC services. It supports the SCA flight operations by implementing the SCA entry rules, providing follow notifications, assigning missed-approach holding fixes, and broadcasting SCA status periodically [3].

2.3 Flight Procedures and Rules

Figure 3 illustrates the normal flight procedures in the SCA. An aircraft that plans to land at the airport sends the AMM a landing request when it is within 5 minutes of a 5 nm radius around the planned IAF. The aircraft enters the SCA either by vertical entry or lateral entry, which is decided by the AMM. In a vertical entry, the aircraft flies to the IAF at a lowest available altitude above the SCA (Point 1 in Figure 3), issued by the ATC. The aircraft holds at the fix until it is cleared by the ATC to descend to the next altitude level. Then it descends over the IAF flying a race track trajectory (from Point 1 to Point 2). The aircraft repeats this procedure until it reaches the lowest altitude level above the SCA, which is 4000 ft AGL (Point 2). It holds at the fix until it receives a message from the AMM granting a vertical entry into the SCA. Then it requests the ATC for clearance to enter the SCA. After it gets the clearance, it descends over the IAF flying a race track trajectory through 4000 to the 3000 feet AGL that is inside the SCA (from Point 2 to Point 3). The aircraft then holds at 3000 feet (Point 3) until it considers safe to descend to 2000 feet AGL. Then it descends to 2000 feet (from Point 3 to Point 4) and holds at the fix, waiting for approach. A lateral entry is possible when there are no other aircraft at or assigned to the aircraft’s planned IAF. In this case, the aircraft receives a message from the AMM granting a lateral entry into the SCA and it directly proceeds to the IAF at or above 2000 feet (Point 2 or Point 3) where it holds and waits for approach. Besides the type of entry, the message from the AMM also contains a follow notification and a missed-approach holding fix assignment. The follow notification is either none, if the aircraft is the first one in the landing sequence, or the identification of a leading aircraft. The aircraft holds at the IAF at 2000 feet until it considers the 3 nautical miles spacing criteria with the leading aircraft is satisfied. It then proceeds from the IAF to the IF (from Point 4 to Point 5), and from there to the FAF (from Point 5 to Point 6) and, finally, to the runway threshold. In case of a missed-approach, the aircraft flies to its assigned missed-approach holding fix at the lowest available altitude (e.g., from Point 7 to Point 3). Then, it either re-initiates the approach following a normal landing procedure or leaves the SCA. As the departure fixes are outside the SCA, the aircraft needs to request clearance from the ATC prior to a departure. After clearance is granted and the aircraft is ready for departure, the departing aircraft monitors the arrival stream for a departure slot.

If an aircraft follows the normal HVO procedure, it will fly along a nominal approach path (NAP) [8]. A NAP consists of a sequence of rectilinear segments leading the aircraft through the approach fixes (IAF, IF, FAF) to the runway threshold. These segments are called as Lateral Entry, Vertical Entry, Holding Pattern, Base Segment, Intermediate Segment, and Final Segment [8]. For example, if an aircraft flies an approach with a lateral entry into the SCA, its NAP is Lateral Entry → Holding Pattern → Base Segment → Intermediate Segment → Final Segment. There are two additional flight segments, Missed-approach and Departure, defined for the nominal missed-approach path (NMAP) that is from MAP to MAHF, and the nominal departure path (NDP) that is from runway to DP.
2.4 Data Communication Systems

Each aircraft in the SCA requires an ADS-B communication system for exchanging state information with other aircraft in the SCA. It also needs a data link system that supports 2-way text messaging from the aircraft to the AMM.

The ground communication system of the airport ground station receives ADS-B messages directly from every participating aircraft and provides the information to the computer executing the AMM software. It supports periodic broadcasting of pertinent airport information and 2-way messaging from the AMM to each participating aircraft using a unique aircraft flight ID.

2.5 Navigation Tools

On board navigation tools provide advisories to aid pilots in following the flight procedures and rules. Navigation tools are of necessity to pilots in lots of occasions, for example, when aircraft hold at 3000 ft, deciding the availability of the 2000 feet altitude; determining the appropriate time to initiate approach; determining the lowest available altitude at missed-approach hold fixes; identifying a departure slot. Pilots also use navigation tools, such as Cockpit Display of Traffic Information (CDIT), for safe separation with other aircraft in the SCA.

3 The Small Aircraft Pilot Assistant (SAPA)

3.1 System Overview

The HVO concept imposes high pilot workload without precedent in the terminal area. Of all the flight phases, approach and landing always has the highest task requirements. In the HVO concept, additional tasks and responsibilities are assigned to pilots who are already under demanding workload and excessive pilot workload eventually leads to deterioration of flight safety.

To resolve the high pilot workload problem, increasing cockpit automation is applied to share more and more pilot tasks. In modern cockpits of commercial transporters and business jets, automation systems like autopilots and FMS are widely employed. For general aviation aircraft, increasing cockpit automation is as well an inevitable trend. In the near future, however, in GA cockpit, automation can not entirely substitute for pilots. For safety purpose, human pilots always remain in the decision-making loop and automation only serves pilots as pilot decision aid.

Design of a pilot decision-aiding tool emphasizes on the interaction between pilots and machines, and how machines can assist pilots in performing a task. What kind of automated tools are more acceptable to pilots? To answer this question, first we need to examine the strengths and weaknesses of both humans and automations. Automations have advantages in monitoring and repeating simple tasks. However, automations are not good at managing information from various sources and making good decisions. On the other hand, according to human factors research done by the FAA, “People are notoriously poor monitors [9],” but “People are flexible information processors who are sensitive to changing conditions and situations. They are resourceful in using both quantitative and qualitative information and in integrating information received from various sources.[9]” It is the exceptional information-managing capacities, complemented with training and experience, that make the pilot an irreplaceable part of the cockpit. In light of the different characteristics of humans and automations, a successful real-time pilot decision-aiding tool is able to decrease pilots’ monitoring tasks and simple repetitive tasks. With the assist of this kind of decision-aid tool, pilots can perform flight tasks more efficiently and safely under time pressures.

There are lots of monitoring and repetitive tasks for pilots in the SCA. For example, when aircraft holding at 3000 feet, they need to check from time to time if the 2000 feet level is available; During the whole approach, they need to identify the current flight segments and if their aircraft are in conformance; The need to monitor the traffic in SCA and detect potential traffic conflict with another aircraft. The SAPA is designed to automate most of these tasks for pilots, so as to decrease their workload and increase their decision-making capacity.

Figure 4 illustrates the modular layout of the SAPA and its interaction with pilots, external data resources, and hardware. It consists of five major
modules and the interface and integration of these different modules is implemented around a central data object that is used to coordinate the data communication between the different modules. The data object also contains a navigation database extracted from the Jeffson database, and it serves as the interface of the SAPA with eternal data recourses.

Figure 4. Small Aircraft Pilot Assistant

3.2 External Data Inputs

There are four major inputs to the SAPA: flight clearances and instructions from the ATC; landing notifications and the SCA status from the AMM; ownship states information and flight plans; and ADS-B messages from other aircraft in the vicinity. One ADS-B message is composed of three parts [8]: the state vector report contains aircraft three-dimensional GPS position and velocity vector information; the mode status report contains aircraft operational information; and the on condition report contains aircraft sequence status and intended route information.

3.3 The Flight Segment Interpreter (FSI)

The FSI determines the current flight segments of all the participating aircraft in an SCA, including that of the ownship. Its algorithm, as illustrated in Appendix I, is based on the information in the ADS-B state vector and on condition report.

A more sophisticated version of the FSI is now under development. The new FSI classifies the current flight segment based on the measurements of aircraft flight states, e.g., positions, airspeeds, obtained from the sensors. These measurements are regarded as “state variables,” which define a state space for the airplane. The flight segments are modeled in terms of the flight variables, which decompose the state space into several partitions. The current flight segment is determined by within which state-space partition the vector of flight variable measurements falls. However, state-space partitions for different flight segments may overlap, which makes modeling flight segments not always unique. A decision method based on Fuzzy Logic decision theory is used to resolve this ambiguity. For each flight segment, a hypertrapezoidal fuzzy membership function [4] is defined upon the state space. When called with a measured vector, the function returns a value between zero and unity indicating the probability of the aircraft within a specific flight segment. The current flight segment is the one with the highest probability. Hypertrapezoidal fuzzy membership functions have been successfully applied in GAPATS system [4], which they are used to segment an aircraft’s flight variables into the different predefined modes of operation, e.g., cruise, initial approach, final approach, etc. Figure 5 shows a typical segmentation of the state space using 2-dimensional hypertrapezoidal fuzzy functions [4].

Figure 5. 2-D Hypertrapezoidal Fuzzy Functions

3.4 The Conformance Monitor (CM)

After the FSI infers the current flight segment of the ownship, the CM determines whether the aircraft is in conformance with the flight segment or not. For each flight segment, there are four major types of rules for conformance determination [8]. The first type is navigation related. It limits the deviation of the aircraft from its NAP within a containment volume (CV) of predefined size and shape around the NAP. For example, Figure 6 illustrates the vertical and horizontal parts of the Holding Pattern containment volume [8]. The second type is speed related. It restricts the
airspeed and vertical rate of the aircraft within a speed profile range that is dependent on the flight segment and the aircraft type. The third type is traffic surveillance related: the aircraft should not be in a mid-air collision conflict with any other aircraft. The last type is the HVO procedure related: the aircraft should follow the normal HVO flight procedures. For example, the aircraft is considered to be out of conformance if it holds too long at the IAF.

Figure 6 Holding Pattern Containment Volume

The CM is implemented as an expert system that consists of a collection of conformance rules, using a common expert system building tool – the CLIPS [10]. The output of the CM includes a Boolean value indicating the conformance status and a list of strings indicating which conformance rules are violated.

3.5 The Traffic Conflict Detector (TCD)

The implementation of the TCD follows the SCA Conflict Detection Logic provided by NASA LaRC [8]. Its algorithm is illustrated in Appendix II. Note that the TCD uses two types of aircraft trajectory projection: if an aircraft is out of conformance, its trajectory projection is state-based, which assumes it flies a constant speed at the current heading for a look-ahead time; if an aircraft is in conformance, its trajectory projection is intent-based, which assumes it follows its NAP and remains within the Contain Volume. Using two different projection methods regarding to the aircraft’s conformance status is to reduce the false alarm rate. A potential traffic conflict between two aircraft occurs if in the predefined look-ahead time (2 minutes), the protected zones (1.5 nautical miles in radius and 250 feet in height) of two aircraft overlap.

3.6 The Pilot Advisor (PA)

Based on the current flight states, the PA decides what advice should be presented to pilots and at which alert level. The advice is of two major types: HVO procedure support and flight operation support. The former assists pilots in performing the immediate task required by the HVO and the latter assesses the manner pilots fly the airplane and advises possible pilot mis-operations. There are three alert levels: warning, caution, and advisory, ordered in decreasing degree of severity. The output of the PA is various types of symbology sets and alerts, via Pilot Interface Manager, to be displayed on the Head Up Display (HUD) and the Head Down Display (HDD).

The HVO procedure support has two sub-modules: An altitude determination tool and an approach spacing tool. The altitude determination function aids the pilot determine the lowest available altitude, both, upon entry to the SCA and when flying a missed approach procedure. An approach spacing tool determines whether the pilot is able to initiate approach without violating the spacing rule, which is to maintain 3 nautical miles behind the leading aircraft throughout the approach. The spacing algorithm we use is dependent on the approach speeds of the leading aircraft and ownship, as illustrated in Appendix III.

The flight operation support assists pilots in safely maneuvering the aircraft by detecting and advising possible pilot mis-operations during the approach. Examples of these advice are as follows: reminding pilots putting down the gear and setting up flaps in final approach; warning pilots when aircraft approaching stall speed; indicating approaching MAP so that pilots can decide whether to do a missed-approach or not. This part of the PA is implemented as an expert system using the CLIPS.

3.7 The Pilot Interface Manager (PIM)

The PIM determines what to and how to present information on the hardware displays. Currently, the SAPA is connected to two displays, a Head Down Display (HDD) and a Head Up Display (HUD).

The HDD is a multi-functional information display for navigation, traffic, and other necessary
flight information. Most of the icons and texts displayed on its moving map are explained in Figure 7. The circle around a traffic aircraft implies its protected zones, which must not overlap with the one around the ownship. If the aircraft has a potential conflict with the ownship, the circle will flash red to warn pilots.

**Figure 7. Head Down Moving Map Display**

Though using HUD is not considered of necessity in the SATS program, pilot test results from our previous research show that using HUD can greatly improve pilot situation awareness. Figure 8 shows the HUD we are using.

**Figure 8 Head Up Display Symbology**

**4 HVO Simulation and SAPA Evaluation**

Implementing a pilot-in-the-loop SATS HVO simulation is the prerequisite for evaluating the SAPA. We developed an open, distributed simulation system called Multi-agent Intelligent Distributed Airspace Simulation (MIDAS) that allows different software entities connected to a central database via the TCP/IP (Figure 9). We developed a software program simulating the AMM, a software program simulating the ATC, and a collection of software agents that we called autonomous pesudo-aircraft simulating the traffic in the SCA. The SAPA is installed on a real-time, nonlinear, six degree-of-freedom fixed base engineering flight simulator (EFS) (Figure 10). The aircraft model we used is that of the Rockwell Commander 700, a light twin-engine, typical GA aircraft. The visual environment of the EFS has the capacity to display out-of-window view of the simulated traffic (Figure 11).
We created a virtual SCA around TSTC airport (KCNW) in Waco, TX. For the convenience of test pilots, we designed a SATS approach plate (Figure 12) based on the current KNCW GPS approach plate. Note that fixes LEROI and TITAH are the actual FAF and MAP of the existing GPS approach, however, fixes RAZVY, FORTR, and LOUIE, which serve as the IAF-R, IF, and IAF-L of the SCA, are created ones.

There are two kinds of test scenarios for evaluating the SAPA: normal HVO scenarios and non-normal HVO scenarios. In normal HVO scenarios, we assume all the aircraft follow exactly the HVO procedures. As it is proved that the flight safety in the SCA is assured if all participating aircraft follow HVO procedures [6], the Conformance Monitor, the Traffic Conflict Detector, and the Pilot Advisor, can only be tested in non-normal scenarios. In these scenarios, we assume, due to the failure of aircraft systems or pilot mistakes, some of the aircraft in SCA either fail to follow the HVO procedures, or fly out of conformance, or incur traffic conflicts with other aircraft. Table 1 lists some test scenarios. In each scenario, there are at least four aircraft, one of which is the piloted aircraft.

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lateral entry for piloted aircraft with missed-approach</td>
</tr>
<tr>
<td>2</td>
<td>Vertical entry for piloted aircraft with successful landing</td>
</tr>
<tr>
<td>3</td>
<td>Leading aircraft has a loss of power, incurring loss of separation with the piloted aircraft</td>
</tr>
<tr>
<td>4</td>
<td>Piloted aircraft initiates approach too early, incurring conflict with its leading aircraft</td>
</tr>
<tr>
<td>5</td>
<td>Pilot aircraft flies to a wrong MAHF, incurring conflict with the holding aircraft</td>
</tr>
</tbody>
</table>

By far we have two test pilots participate in the simulation evaluation of the SAPA, a certificated flight instructor and a licensed private pilot. The pilot tests show that, with the assistance of the SAPA, pilots can successfully follow the HVO procedures. General response from both pilots is that using the SAPA can improve their situation awareness, decrease their workload, and help them make correct decisions in non-normal situations. However, both pilots pointed out that in some situations, there are too many messages presented on the HDD display, which leads to poor situation awareness due to information overwhelming. As humans can absorb and make use of only limited quantities of information, the next challenge of the SAPA is, given a situation, to determine what information is truly significant and present critical information without overwhelming the pilot.

5 Conclusions and Future Work

The High Volume Operation (HVO) concept of the Small Aircraft Transportation System (SATS) is based on a distributed decision-making environment within which pilots are left with most of the decision-making responsibility. In this paper,
we presented an onboard pilot decision aid system, called the Small Aircraft Pilot Assistant, which is dedicated to SATS aircraft conducting HVO. It aims at increasing the cockpit decision-making capacity by automating part of pilot decision-making process, especially in its early stages of information acquisition and analysis. The functionalities of the SAPA are as follows: assisting pilots in following the HVO procedures; identifying current flight segment; monitoring pilot performance; advising possible pilot mis-operation; and advising potential traffic related hazards. The desired characteristics of the SAPA are attributed to Artificial Intelligence techniques used such as Fuzzy Logic and Expert Systems. For the pilot evaluation of the SAPA, we developed a real-time, multi-aircraft, pilot-in-the-loop simulation system called the Multi-agent Intelligent Distributed Airspace Simulation. It is presently capable of middle-fidelity HVO simulation. Preliminary pilot test results show that the SAPA is a promising system to satisfy the cockpit system requirements of the SATS HVO.

Our next step is conducting large-scale pilot tests and human factors evaluation for the SAPA. Moreover, a new task of installing the SAPA on a real aircraft has been proposed to the NASA SATS program.

6 Acknowledgment
This research is funded by NASA Small Aircraft Transportation System Program. The authors gratefully acknowledge this support.

7 Reference:

8 Email Addresses
Jie Rong, jierong@tamu.edu
John Valasek, valasek@aero.tamu.edu
function Flight-Segment-Interpreter (AC) returns a string value
    inputs: AC, a data structure of the aircraft’s information obtained from its ADS-B state vector and on condition reports
    if AC.MissedApproachBit = true return Missed Approach
    else
        switch AC.WayPointType
        case IF
            return BaseSegment
        case FAF
            return IntermediateSegment
        case MAP
            return FinalSegment
        case DP
            return Departure
        case IAF
            if InContainmentVolume(AC_Position, HoldingPattern) = true
                return Holding Pattern
            else
                if AC.EntryType = LEN return LateralEntry
                else return VerticalEntry
10 Appendix II. Traffic Conflict Detector Algorithm

function Traffic-Conflict-Detector(Intruder, OwnAC) returns CDOutput
inputs: Intruder, a data structure of another aircraft’s information obtained from its ADS-B message
        OwnAC, a data structure of own aircraft’s information
outputs: CDOutput, a data structure
local variables: IntruderTrajectory, intruder’s trajectory
                OwnTrajectory, own aircraft’s trajectory
                Instruder2OwnAC, distance between intruder and own aircraft at each time step
                t, current time step
                LookAhead, look-ahead time steps
                ProtectedZone, size of the protected zone
initialize t, LookAhead, ProtectedZone
if Intruder.ConformanceBit = true
   IntruderTrajectory=IntendedPathProjection(Intruder, LookAhead)
else
   IntruderTrajectory=LinearPathProjection(Intruder, LookAhead)
if OwnAC.ConformanceBit = true
   OwnTrajectory=IntendedPathProjection(OwnAC, LookAhead)
else
   OwnTrajectory=LinearPathProjection(OwnAC, LookAhead)
do t < LookAhead
   Instruder2OwnAC(t) = Distance(IntruderTrajectory(t), OwnTrajectory(t))
   if Instruder2OwnAC(t-1) < Instruder2OwnAC(t)
      CDOutput.DistanceOfClosestApproach = Instruder2OwnAC(t-1)
   if Instruder2OwnAC(t-1) > ProtectedZone and Instruder2OwnAC(t) <= ProtectedZone
      CDOutput.StartTimeInterval = t
   if Instruder2OwnAC(t-1) <= ProtectedZone and Instruder2OwnAC(t) >ProtectedZone
      CDOutput.EndTimeInterval = t
   t=t+1
return CDOutput

function IntendedPathProjection(AC, T) returns a list of T 3-D waypoints
inputs: AC, a data structure of the aircraft’s states
        T, look-ahead time steps

function LinearPathProjection(AC, T) returns a list of T 3-D waypoints
inputs: AC, a data structure of the aircraft’s states
        T, look-ahead time steps

function Distance(WP1, WP2) returns the distance between two waypoints
inputs: WP1, WP2, 3-D waypoints
Appendix III. Initiate Approach Algorithm

```plaintext
function Initiate-Approach(FollowingAC, OwnAC, SCA) returns a Boolean value
inputs: FollowingAC, a data structure of the following aircraft’s information obtained from its ADS-B message
        OwnAC, a data structure of own aircraft’s information
        SCA, a data structure of the SCA’s information
local variables: IAF2IF, distance from OwnAC.IAF to SCA.IF
                IF2FAF, distance from SCA.IF to SCA.FAF
                FAF2MAP, distance from SCA.FAF to SCA.MAP
                T1, time needed for own aircraft to fly from Own.IAF to the location 3.5 nautical miles away from SCA.MAP
                T2, time needed for the following aircraft to fly from its current location to the location 3 nautical miles away from SCA.MAP
IAF2IF = Distance(OwnAC.IAF,SCA.IF)
IF2FAF = Distance(Own.IF,SCA.FAF)
FAF2MAP = Distance(Own.FAF,SCA.MAP)
T1 = IAF2IF/OwnAC.HoldingSpeed + IF2FAF/OwnAC.InitialApproachSpeed + FAF2MAP/OwnAC.FinalApproachSpeed
if (FollowingAC.ID = NONE) return true
else
    switch (FollowingAC.IntendedWayPointType)
    case IF
        T2 = Distance(FollowingAC,SCA.IF)/FollowingAC.AirSpeed + IF2FAF/FollowingAC.MinInitialApproachSpeed + FAF2MAP/FollowingAC.MinFinalApproachSpeed
        if (T2<=T1) return true
        else return false
    case FAF
        T2 = IF2FAF/FollowingAC.MinInitialApproachSpeed + FAF2MAP/FollowingAC.MinFinalApproachSpeed
        if (T2<=T1) return true
        else return false
    case MAP return true
default return false

function Distance(WP1, WP2) returns the distance between two waypoints
inputs: WP1, WP2, 3-D waypoints
```