Analysis of Safety Benefits of NASA Aviation Safety Program Technologies

NS112S2

December 2002

Shahab Hasan
Robert Hemm
Scott Houser

Analysis of Safety Benefits of NASA Aviation Safety Program Technologies

NS112S2

December 2002

Shahab Hasan
Robert Hemm
Scott Houser

The views, opinions, and findings contained in this report are those of LMI and should not be construed as an official agency position, policy, or decision, unless so designated by other official documentation.

LOGISTICS MANAGEMENT INSTITUTE
2000 CORPORATE RIDGE
MCLEAN, VIRGINIA 22102-7805

Prepared pursuant to National Aeronautics and Space Administration Contract GS-23F-0304K
The views expressed here are those of the Logistics Management Institute at the time of issue but not necessarily those of NASA. Permission to quote or reproduce any part except for government purposes must be obtained from the Logistics Management Institute.

LOGISTICS MANAGEMENT INSTITUTE
2000 CORPORATE RIDGE
MCLEAN, VIRGINIA 22102-7805

Contents

Chapter 1 Introduction and Summary 1-1

Chapter 2 Modeling Changes and Improvements 2-1

General Software Enhancement 2-1

Human Factors Library 2-1

Modeling Time in the Action Sequence 2-4

Chapter 3 Synthetic Vision: Terrain Avoidance Scenario 3-1

Timeline Model 3-1

Action Sequence Model 3-2

Action Sequence File 3-3

Input Parameters and Metrics 3-5

Chapter 4 Weather Accident Prevention: Turbulence Avoidance and Mitigation Scenario 4-1

Timeline Model 4-1

Action Sequence Model 4-2

Action Sequence File 4-3

Input Parameters and Metrics 4-4

Chapter 5 Synthetic Vision: Collision Avoidance Scenario 5-1

Timeline Model 5-1

Action Sequence Model 5-2

Action Sequence File 5-3

Input Parameters and Metrics 5-4

Instruments 5-7

Other Scenario Characteristics 5-7

Output Metrics 5-7

Results 5-8

Chapter 6 Synthetic Vision: Runway Incursion Avoidance Scenario 6-1

Timeline Model 6-2

ACTION SEQUENCE MODEL 6-3
ACTION SEQUENCE FILE 6-3
INPUT PARAMETERS AND METRICS 6-5
OUTPUT METRICS 6-6
RESULTS 6-6

Chapter 7 System-wide Accident Prevention Runway Incursion Avoidance Scenario 7-1
DECISION MAKING AND THE HUMAN-FACTORS ACTION SEQUENCE 7-2
TIMELINE MODEL 7-3
ACTION SEQUENCE MODEL 7-4
ACTION SEQUENCE FILE 7-5
INPUT PARAMETERS AND METRICS 7-7
OUTPUT METRICS 7-8
RESULTS 7-8

FIGURES

Figure 2-1: A Human Factors Action Sequence 2-4
Figure 2-2: Time as a Random Geometric Variable 2-5
Figure 2-3: Time as an Exponential Random Variable 2-6
Figure 2-4: Time as a Weibull Random Variable 2-6
Figure 3-1: Timeline Model for Terrain Avoidance with Synthetic Vision 3-2
Figure 3-2: Action Sequence Model for Terrain Avoidance with Synthetic Vision 3-3
Figure 4-1: Timeline Model for Turbulence Avoidance and Mitigation Scenario 4-2
Figure 4-2: Action Sequence Model for Turbulence Avoidance and Mitigation Scenario 4-2
Figure 5-1. Timeline Model for the Collision Avoidance Scenario with Synthetic Vision 5-2
Figure 5-2. Action Sequence Model for Turbulence Avoidance and Mitigation Scenario 5-3
Figure 5-3. Major Input Parameters for Collision Avoidance Scenario 5-5
Figure 5-4. Grid of Relative Starting Locations for Aircraft 5-6
Figure 5-5. Collision Avoidance Scenario's Envelopes 5-6

Figure 5-6. Incident Probabilities for Collision Avoidance Scenario 5-9
Figure 5-7. Safety Gain of the Variant Case over the Baseline Case 5-9
Figure 5-8. 24-Bin Partition of Intruder Initial Location (not to scale). 5-10
Figure 5-9. Safety Gain at Most Hazardous Subset of Initial Intruder Locations 5-11
Figure 6-1. Runway Incursion Scenario 6-1
Figure 6-2. Timeline for the Runway Incursion Scenario with Synthetic Vision 6-2
Figure 6-3. Action Sequence Model for Runway Incursion Scenario 6-3
Figure 6-4. Comparison of Incursion Probability Between Baseline and Variant Cases 6-7
Figure 7-1. Runway Incursion Scenario for SWAP 7-2
Figure 7-2. Timeline Model for the SWAP Runway Incursion Scenario 7-4
Figure 7-3. Action Sequence Model for Runway Incursion Scenario 7-5
Figure 7-4. Comparison of Incursion Probabilities for Correct and Incorrect ATC Initial Actions (SV and Correct ATC Actions Shown for Comparison) 7-9
Figure 7-5. Safety Gain from Implementing Correct ATC Actions as Opposed to Incorrect ATC Actions 7-10

Chapter 1
Introduction and Summary

Through its Aviation Safety Program, the National Aeronautics and Space Administration (NASA) aims to develop and demonstrate technologies that contribute to reducing the rate of aviation fatal accidents by a factor of five by the year 2007 and by a factor of 10 by the year 2022. Six projects comprise NASA's safety program—Synthetic Vision (SV), Weather Accident Prevention, System-Wide Accident Prevention (SWAP), Single Aircraft Accident Prevention, Aviation System Monitoring and Modeling, and Accident Mitigation. NASA asked LMI to build a safety benefit software model and analyze the first three projects.

Our safety analysis comprises two principal components: a reliability model and a simulation model. In the reliability model, we categorize the technology into basic components, such as hardware, software, and human agents; define how those components interact; and then determine the failure modes and rates of the components. The results determine how different technologies and systems will perform in normal, degraded, and failed modes of operation. In the simulation, we model an operational scenario and use Monte Carlo methods and specific failures (based on the reliability analysis) to investigate the performance of the technology and identify areas warranting further exploration.

Our previous report[1] for this task documented our overall approach and method in detail. It also presented the results of our analysis for the following scenarios and technologies:

- *SV: Terrain avoidance*. This scenario models a classic controlled flight into terrain (CFIT) accident and assesses the reduced probability of an accident if NASA's SV technology were used.

- *Weather accident prevention technologies: Turbulence avoidance*. This scenario models an aircraft that may encounter a turbulence cell and assesses the additional warning time and reduced chance of encountering turbulence provided by NASA's Turbulence Prediction and Warning System.

We do not reproduce the results of these two scenarios in this report because they have not changed. For completeness, however, we review the parameters of these scenarios because we have updated the software model and we should compare

[1] Shahab Hasan, Robert Hemm, and Scott Houser, *Preliminary Results of an Integrated Safety Analysis of NASA Aviation Safety Program Technologies: Synthetic Vision and Weather Accident Prevention*, LMI Report NS112S1, March 2002.

the new scenarios with the previous scenarios. The new scenarios are the following:

- *SV: Collision avoidance*. This scenario models two aircraft on final approach to independent runways when one pilot blunders and intrudes on the flight path of the other aircraft. The results show the benefit of NASA's SV technology in augmenting the safety level provided solely by an air traffic controller (ATC) monitoring the situation.

- *SV: Runway incursion avoidance*. This scenario models an aircraft landing on a runway when a taxiing aircraft mistakenly starts to cross the active runway. The results show the benefit of NASA's SV technology in detecting the incursion earlier and the corresponding additional time for maneuvering to avoid an accident.

- *SWAP: Runway incursion avoidance*. The setup of this scenario is identical in most respects to the one above. In this case, however, instead of using SV technology, we assess the potential benefits of faster and better decision-making enabled by the training initiatives of NASA's SWAP project.

Before discussing the scenarios, we discuss the new human factors model and its effects on the scenarios. We then describe each scenario, diagram its timeline and action sequence, describe its input parameters and output metrics, and present the results of our analysis.

Our analysis for the new scenarios show the technology is significantly beneficial. In the SV: collision avoidance scenario, we measure the increase in safety by computing the safety gain, which we define as the baseline (no SV) probability of an incident divided by the variant (with SV) probability of incident. The safety gain increases non-linearly as the initial separation between the two aircraft in the scenario increases. As an example, with 4,500 feet initially separating the two aircraft, the safety gain is 140; i.e., in this scenario, an accident is 140 times less likely when the SV technology is used to augment the ATC's monitoring.

For the SV: runway incursion avoidance scenario, the safety gain is 5 for a 2,000-foot simulation envelope (the distance from the aircraft to the runway threshold at which the pilots or the ATC should recognize a problem and go around). At first glance, this seems like a relatively low value, implying a small benefit. However, in the variant case (with SV), our simulation shows no incursions at all when the simulation envelope is increased beyond 2,000 feet. In contrast, in the baseline (no SV) case, incursions continue to occur until the simulation envelope is increased to 12,000 feet or more.

For the SWAP: runway incursion avoidance scenario, we do not compute actual safety benefits attributable to SWAP. Without information about specific decision-making processes, and types of operational errors being targeted, as well as the

goals or expected improvements, we are limited to designing our software to accept such parameters when they are available. Without such information, we run variations of the runway incursion avoidance scenario in which the ATC either correctly recognizes that the intruder is on the active runway, or the ATC mistakenly believes the intruder is on a clear runway. If ATC makes a mistake, he recognizes the mistake after some amount of time and corrects it. The safety gain is defined as the baseline (mistake always made) probability of an incursion divided by the variant (mistake never made) probability of incursion. The safety gain increases nonlinearly as the simulation envelope increases. As an example, with a 10,000-foot simulation envelope, the safety gain is 12.

In a more realistic (albeit still notional) SWAP scenario, the ATC that has been trained under a SWAP initiative may still make the mistake we modeled. Obviously, even without such training, an ATC would not make the mistake we modeled every time. Each ATC would have a probability of making a mistake; presumably, both would be far smaller than 1.0. As the SWAP project continues and values can be assigned for the baseline and the “with SWAP” probabilities of mistakes, our simulation enables calculating an actual SWAP benefit.

Chapter 2
Modeling Changes and Improvements

GENERAL SOFTWARE ENHANCEMENT

Since building and delivering the prototype simulation software model, we have made numerous changes and improvements. The overall design has evolved, the code programming has been cleaned up, and more clarifying comments have been incorporated. One significant change is that we have made the aircraft data structure more focused and less broad: it now models only the flight characteristics of an aircraft. The earlier prototype software used an aircraft data structure which also contained code for controlling and managing the overall simulation. We've moved that sort of executive functionality to a new simulation control object.

HUMAN FACTORS LIBRARY

Another significant change we made was revamping the software that implements human factors modeling to make it more generic and user friendly. The new human factors library makes modeling easier for both the programmer and the user. These changes do not affect the high-level design of the human factors models; the user can still model them using the diagrams in our previous report. Some of the input files have changed, however. We modified the Reconfigurable Flight Simulator (RFS)[1] configuration file slightly, and we created a new human factors file that specifies the relevant action sequences for each scenario. The new data structures for the human factors are a standard way to proceed through the actions of each scenario.

Each scenario now requires the following input files:

- An RFS configuration file
- A human factors action-sequence file
- One or more instrument files (e.g., radar for detecting terrain)
- One or more obstacle files (e.g., terrain objects, turbulence cells, and intruding aircraft).

[1] Shahab Hasan, Robert Hemm, Scott Houser, and Owen Lanterman, *Selection of a Simulation Programming Environment for Air Safety Simulations: The Reconfigurable Flight Simulator*, LMI Report NS115-01, August 2002.

To create a new scenario, the programmer need only provide entry points—distance thresholds between the aircraft and an obstruction—that start the action sequence. These entry points vary for each scenario, and must be custom coded. The full action sequence no longer needs to be programmed, however. The action sequence is a part of the base scenario object, and each scenario uses the same generic function calls to execute it.

The user now can model all the human factors in a scenario using a standard file format. This format also gives users the freedom to use a range of random distributions to model response times, instead of having to use the geometric variables as in the prototype software. Although we've written a separate design guide for future software programmers, the remainder of this chapter explains the philosophy and implementation of the new human factors software design for the user.

The new human factors library takes concepts used throughout this project and standardizes them in software. As a result, the user can create a new *action sequence*—the set of all human actions required to complete a scenario—at run time. The programmer still needs to write custom code that establishes what event or events start the action sequence. However, the code is significantly less than that required to program an entire action sequence. Further, the required custom code is confined to a single object in the simulation, thus improving the maintainability and extensibility of the code.

The basic element of the human factors library is the *task*. The task consists of three phases: a detection (or diagnosis) phase, a decision phase, and a reaction phase. Each of these phases consists of random variables for time. Our previous report[2] described the basic format for a task. Human actors can do one or more of these tasks sequentially. A set of sequential tasks is called a *series*.

The series is the lowest level functional object in the library. Each series becomes active when a specific event occurs in the simulation. When a series is active, it generates detection, decision, and reaction times for each of its tasks to create a completion time.

A set of series can occur in parallel. For example, the pilot can use his instruments to determine his proximity to an obstruction, such as another aircraft. At the same time, the air traffic controller can use his instruments to determine the same proximity, and communicate what he knows to the pilot. If the aircraft gets too close to the obstruction, the pilot will take corrective action when the first of these two series is complete: when the pilot sees from his instruments that he is too close, or when he hears from the ATC that he is too close.

A set of series performed in parallel is called a *group*. A group is complete when one series in the group is completed. Each series in a group has its own activation

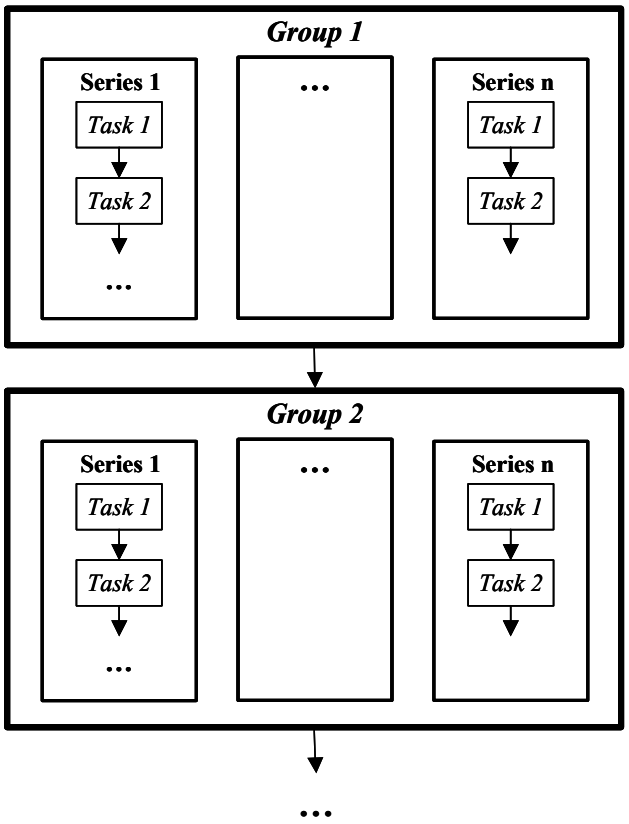
[2] Shahab Hasan, Robert Hemm, and Scott Houser, *Preliminary Results of an Integrated Safety Analysis of NASA Aviation Safety Program Technologies: Synthetic Vision and Weather Accident Prevention*, LMI Report NS112S1, March 2002.

event, and each series is independent of the others in the group. That means that we do not model multitasking effects, such as interference and prioritization. Because of the tactical time frame of the scenarios we model, modeling these effects is unnecessary.

A single group can model any scenario in which several parties monitor the same condition, and the same actions are required as soon as one of the parties discovers that the condition exists. One example of such a scenario is the terrain avoidance scenario with synthetic vision. As soon as the pilot discovers he is too close to a terrain obstruction, or the ATC discovers it and informs the pilot, the pilot must correct his course to avoid the terrain. Other scenarios require more than one group of actions. For example, the turbulence avoidance scenario requires two groups in series. The first group covers the time needed to discover a turbulence cell in the aircraft's path. Once the cell is discovered, the pilot will immediately begin to change the speed of the aircraft to the ideal penetration speed. A second group covers the time needed for the pilot and ATC to discuss a change in altitude. If such a change is allowable, the pilot will attempt to fly over or under the turbulence. A sequence of groups is called the *action sequence.*

Each scenario has only one action sequence. The sequence can be as simple as a single task, or can be as complicated as a long string of groups (Figure 2-1). Each series in a group can be activated when its group is active. When the group is active, each series can be activated separately, depending on the different stimuli from the simulation. When the pilot has completed all the actions he needs to begin a maneuver, the group is complete. For example, the goal of a group of actions may be to turn the aircraft. When the group is complete, the simulation commands the aircraft object to begin a turn. The actual turn may take much longer than the human actions required to begin the turn. The group models only the human actions required to begin the turn, not the turn itself.

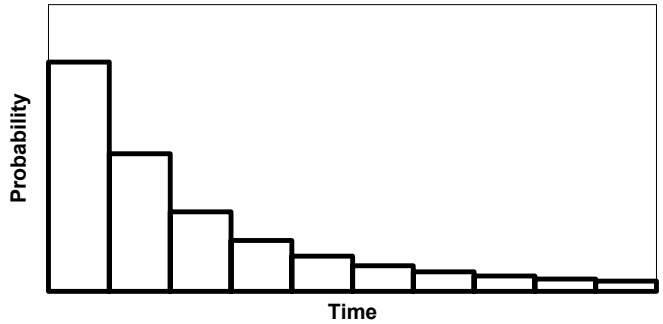
Figure 2-1. A Human Factors Action Sequence



MODELING TIME IN THE ACTION SEQUENCE

In the previous generation of the simulation, we modeled time as a geometric random variable (Figure 2-2). This discrete distribution has many properties that make it suitable for modeling time. Its support has a finite left tail and an infinite right tail, which means that an event modeled with a geometric distribution has a definite start time and an open-ended completion time. The probability of completing the modeled task after a certain time diminishes at a geometric rate, meaning that it is unlikely, but possible, that an event will take significantly longer than its expected duration. Many tasks are a sequence of repeating, independent trials spaced at equal intervals from each other. For example, when an ATC monitors his radar, it is reasonable to model the time required for him to notice an off-course aircraft as a series of Bernoulli trials: each trial is a scan of his screen.

Figure 2-2. Time as a Random Geometric Variable



The previous simulation's geometric model for time had a few limitations. First, the model is discrete: its resolution depends on the simulation time step. Second, the spacing of the Bernoulli trials in the previous version of the simulation is uniform: one trial occurs at each simulation time step. Often modeling the time that elapses between each trial as another random variable is more accurate. For example, a pilot may monitor an instrument frequently and regularly through most of the flight, but may monitor it infrequently and sporadically when distracted. Finally, a geometric probability density function is strictly (in fact, geometrically) decreasing. It would be more flexible to allow the possibility of distinct modes in the distribution at times greater than zero.

The new human factors library enables the user to pick a continuous distribution for time. We added two variables to the random variable library that model time well: the exponential and Weibull distributions. The exponential distribution has a strictly decreasing property, like the geometric distribution (Figure 2-3). However, this variable has the advantage of being continuous. The Weibull distribution can be strictly decreasing, or it can have a bell-shaped, non-zero mode (Figure 2-4). Both distributions have a definite start time and an indefinite completion time, which is critical for modeling time realistically. Weibull random variables are more flexible than the exponential random variables, but they may be a little more difficult for inexperienced users to use.

Figure 2-3. Time as an Exponential Random Variable

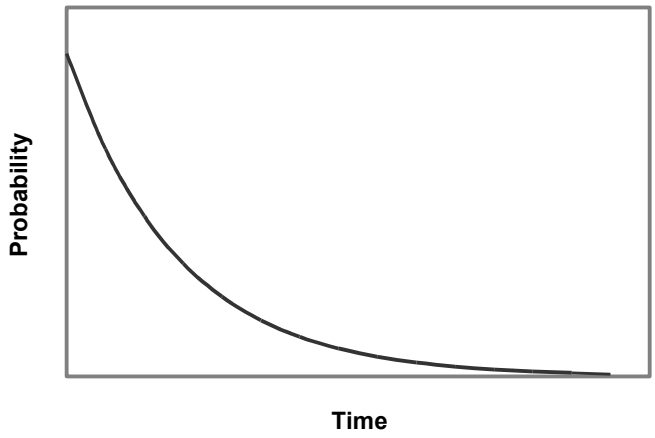
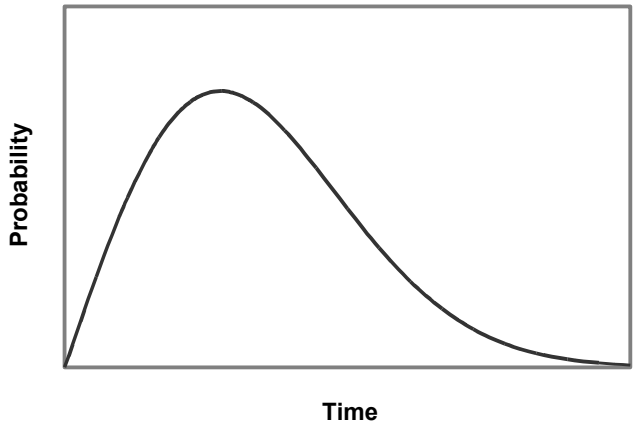


Figure 2-4. Time as a Weibull Random Variable



Chapter 3
Synthetic Vision: Terrain Avoidance Scenario

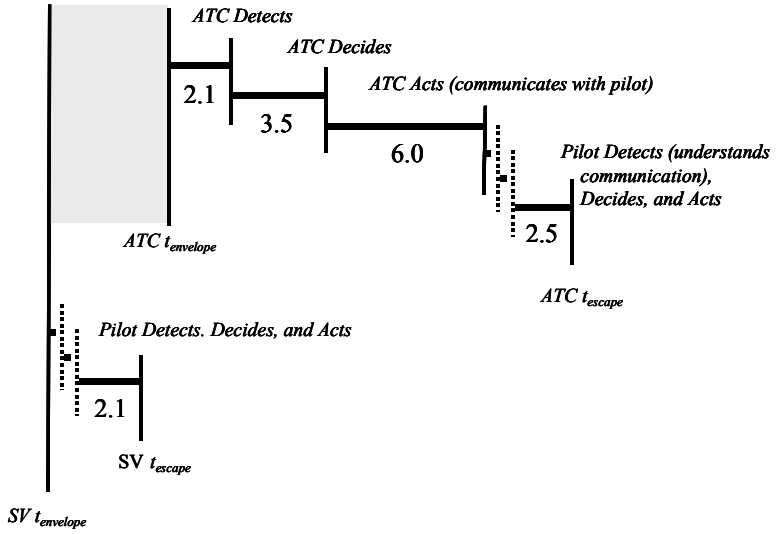
Our previous report[1] discussed the aircraft maneuvers, instrumentation, geometry, and results of the terrain avoidance scenario in great detail. For completeness, in this chapter we review the scenario's human factors model, and describe how we implement that scenario using the new human factors library. Since we have generated and reported results for this scenario previously, we do not present any new results for this scenario in this report.

TIMELINE MODEL

The high-level human factors model remains essentially unchanged from the previous version (Figure 3-1). In the baseline case, the ATC detects that the pilot is off course and heading towards a terrain obstruction by observing his radar. The following time values are nominal expected values, which vary stochastically. We expect that the ATC will take about 2.1 seconds to see the problem once it appears on his scope, 3.5 seconds to decide that it is a problem and he needs to act on it, and 6.0 seconds to act. He acts by informing the pilot of the problem and directing a response. The pilot then hears the response, understands it, and reacts to correct his course. In the synthetic vision case, the ATC continues to monitor for a problem, so the baseline series of tasks remains active. In addition, the pilot uses synthetic vision to determine a problem and react by changing course. The gray area at the beginning of the ATC series of tasks indicates the possible lag time between a radar with a five second sweep and a synthetic vision monitor with a near-real-time update rate.

[1] Shahab Hasan, Robert Hemm, and Scott Houser, *Preliminary Results of an Integrated Safety Analysis of NASA Aviation Safety Program Technologies: Synthetic Vision and Weather Accident Prevention*, LMI Report NS112S1, March 2002.

Figure 3-1: Timeline Model for Terrain Avoidance with Synthetic Vision

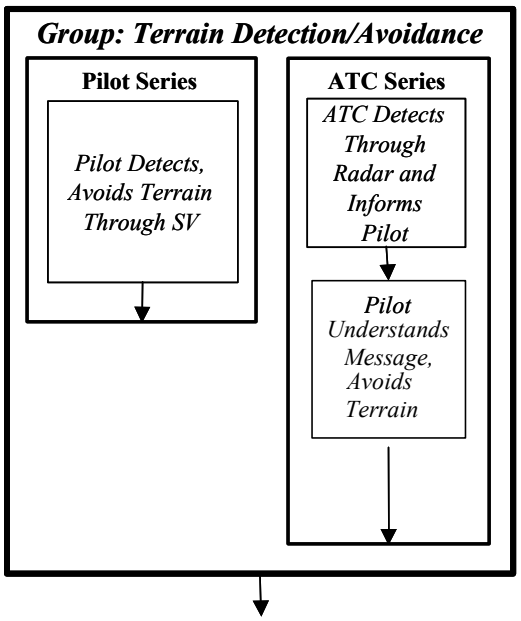


To make the tasks more clear, we explicitly show each as a series of three subtasks in the timeline. The dashed timeline segments indicate zero time subtasks.

ACTION SEQUENCE MODEL

The action sequence model shows how the timeline model fits into the data structure for an action sequence. The action sequence model, therefore, helps design the action sequence's input file. In the terrain avoidance scenario, SV and ATC radar are two independent instruments that monitor for a problem simultaneously. The action sequence, therefore, is a single group structure composed of two series. The first series, which we call the pilot series, contains a single task: detecting and avoiding a terrain encounter as discovered by monitoring the synthetic vision equipment. The second series contains two tasks. The first models the ATC's actions from the point of discovering the problem by monitoring the radar to the point where he communicates the problem to the pilot. The second task models the pilot's actions from the point that he first hears the communication to the point where he begins his course correction.

Figure 3-2: Action Sequence Model for Terrain Avoidance with Synthetic Vision



When SV is active, both series operate in parallel. The simulation activates the series when the modeled equipment (SV or ATC radar) indicates that the aircraft is within a critical threshold distance of a terrain obstacle. Whenever the first of these two series is completed, the aircraft begins an emergency turn to avoid the obstacle.

ACTION SEQUENCE FILE

The input file for the terrain avoidance action sequence appears below, in Courier font. A section-by-section description explains how the file's contents implement the action sequence. The action sequence is a list of its groups. A group is, in turn a list of its series, and a series is a list of its individual tasks. Each of the human factors data structures has the following general format in the file:

- Name of the data structure
- Number of substructures in its list (if applicable)
- Each substructure in the list, in the appropriate order

```
TerrainAvoidanceSequence
NUMBER_OF_GROUPS
1
```

```
GROUP_LIST
```

Header for the group list

```
TerrainAvoidanceGroup
NUMBER_OF_SERIES
2
```

Name of first (and only) group and number of series in the group

```
SERIES LIST
```

Header for the series list

```
ATCSeries
NUMBER_OF_TASKS
2
```

Name of first series in the group and number of tasks in the group (This is the two-task ATC series.)

```
TASK_LIST
```

Header for the ATC series list

```
ATCDetectsAndComms
Detection_Random_Variable
0
2.1
Decision_Random_Variable
0
3.5
Reaction_Random_Variable
0
6.0
```

First task in the ATC series (The ATC detects a problem on the radar screen, decides what to do, and contacts the pilot. All time random variables are exponential (denoted by "0," followed on the next line by the expected value.)

```
PilotCommsAndMoves
Detection_Random_Variable
3
0.0 0.0
Decision_Random_Variable
3
0.0 0.0
Reaction_Random_Variable
0
2.5
```

Second task in the ATC series (The pilot receives the message from the ATC that a problem exists, and reacts to it. We model all three subtasks as one exponential random variable of mean = 2.5 seconds. The detection and decision random variables are null [uniformly distributed with extremums = 0.0 seconds].)

```
PilotSeries
NUMBER_OF_TASKS
1
```

Name of second series in the group and number of tasks in the group (This is the single-task pilot series.)

```
TASK_LIST
```

```
PilotDetectsAndMoves
Detection_Random_Variable
3
0.0 0.0
Decision_Random_Variable
3
0.0 0.0
Reaction_Random_Variable
0
2.1
```

Only task in the pilot series (The pilot reacts to a problem that his SV equipment reveals. We model all three subtasks as one exponential random variable of mean = 2.1 seconds.)

INPUT PARAMETERS AND METRICS

Our previous report described the scenario's physical input parameters in detail. We did not change the parameters in this version of the scenario. The parameters include the aircraft's speed, the threshold distance at which the ATC or pilot should determine that a problem exists, the speed of the aircraft, its turn rate, and its accident distance.

We also did not change the scenario's output metrics. These include the times when each series becomes active, the completion time of each completed subtask, the total action sequence time, and the minimum distance between the aircraft and terrain.

Chapter 4
Weather Accident Prevention: Turbulence Avoidance and Mitigation Scenario

In our previous report,[1] we discussed the aircraft maneuvers, instrumentation, geometry, and results of the turbulence avoidance and mitigation scenario in great detail. In this chapter, we review the scenario's human factors model, and describe how we implement that scenario using the new human factors library. Because we have generated and reported results for this scenario previously, we do not present any new results for this scenario in this report.

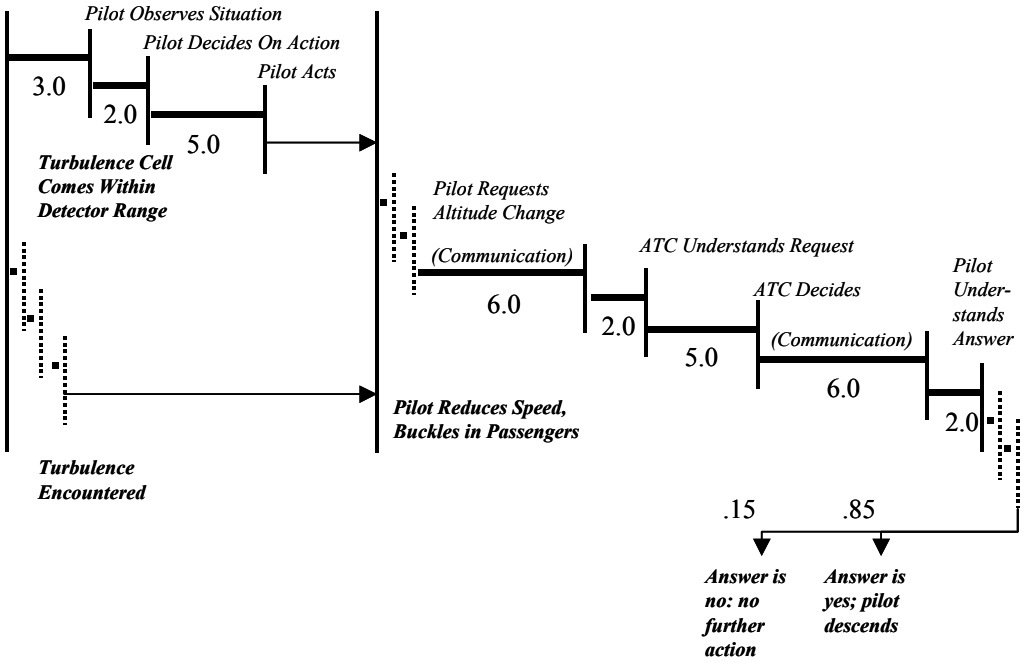
TIMELINE MODEL

The timeline model for the turbulence avoidance and mitigation scenario has the same flow as the original version, but we changed it slightly to fit the data structures in the new human factors library better (Figure 4-1). Once the weather accident prevention suite of tools shows turbulence ahead, the pilot completes a task that starts with observing the indication and results in reducing the speed of the aircraft while ordering the passengers to buckle their seatbelts. This task is expected to take 10 seconds. If the turbulence object is never detected, then the pilot will immediately reduce speed and order the passengers to buckle their seatbelts. We note that this task is instantaneous by representing all three of its subtasks as zero-time subtasks.

Once the pilot reduces speed, he and the ATC coordinate to change the altitude of the aircraft, if that is feasible. Ideally, the pilot would like to fly above or below the turbulence without encountering it at all. The first of the three tasks that this coordination requires is the pilot's request for an altitude change to the ATC. We expect this task to take 6 seconds. We have created detection and decisions subtasks in this task because the pilot's previous actions have already made clear the need to take this action. In the second task, the ATC must understand the pilot's request, decide whether to grant it, and communicate his answer to the pilot. In the final task, the pilot must understand the ATC's response and act on it. The timeline shows that the pilot will either do nothing or change the aircraft's altitude, depending on the ATC's response. Because he initiated this response with his own request, we assume that the decision and reaction subtasks for this task occur instantly. The following time values are nominal expected values, which vary stochastically.

[1] Shahab Hasan, Robert Hemm, and Scott Houser, *Preliminary Results of an Integrated Safety Analysis of NASA Aviation Safety Program Technologies: Synthetic Vision and Weather Accident Prevention*, LMI Report NS112S1, March 2002.

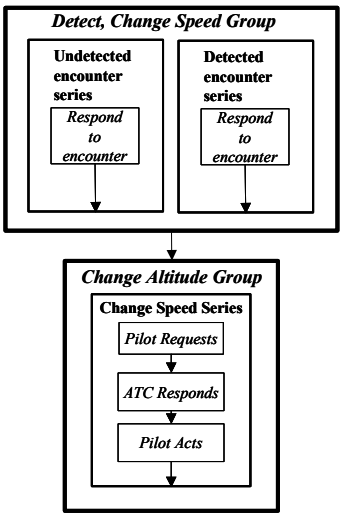
Figure 4-1: Timeline Model for Turbulence Avoidance and Mitigation Scenario



ACTION SEQUENCE MODEL

The action sequence model converts the timeline shown above into a sequence of two groups (Figure 4-2). Detected encounters and undetected encounters are each modeled as a series in the first group. When the first group is completed, the aircraft changes its speed. The second group has one series of three tasks. This series models the request for a change in altitude and the ATC's response to the request.

Figure 4-2: Action Sequence Model for Turbulence Avoidance and Mitigation Scenario



This action sequence shows how to model multiple actions through multiple groups. When the first group is completed, the aircraft begins reducing speed; when the second group is completed, the aircraft may change altitude. We do not model the ATC's response (yes or no) in the action sequence itself. That particular random variable is modeled external to the human factors portion of the scenario.

ACTION SEQUENCE FILE

```
TurbulenceAvoidanceSequence
NUMBER_OF_GROUPS
2

GROUP_LIST

EncounterGroup
NUMBER_OF_SERIES
2

SERIES LIST

UndetectedEncounter
NUMBER_OF_TASKS
1

TASK_LIST

UndetectedEncounterResponse
Detection_Random_Variable
3
0.0 0.0
Decision_Random_Variable
3
0.0 0.0
Reaction_Random_Variable
3
0.0 0.0
```

The assumption is that the response to an undetected encounter (i.e., the pilot becomes aware of turbulence only when the aircraft flies into it) is instantaneous.

```
DetectedEncounterResponse
NUMBER_OF_TASKS
1

TASK_LIST

DetectedEncounterResponse
Detection_Random_Variable
0
3.0
Decision_Random_Variable
0
2.0
Reaction_Random_Variable
0
5.0
```

Each of the three subtasks for this task has an exponential distribution. The subtasks are defined so 10 seconds is required from first seeing the turbulence in the cockpit to taking action to mitigate its effects.

```
ChangeAltitudeGroup
NUMBER_OF_SERIES
1

SERIES LIST

ChangeAltitudeSeries
NUMBER_OF_TASKS
3

TASK_LIST

PilotRequests
Detection_Random_Variable
3
0.0 0.0
Decision_Random_Variable
3
0.0 0.0
Reaction_Random_Variable
0
6.0
```

It takes 6 seconds, on average, for the actual request communication to occur. Because the pilot would realize the need for such a request while completing previous actions, performing the first two subtasks takes no time.

```
ATCResponds
Detection_Random_Variable
0
2.0
Decision_Random_Variable
0
5.0
Reaction_Random_Variable
0
6.0
```

It takes an average of 13 seconds for the ATC to hear the pilot's request, consider its implications, and respond to it. Because of the complexity of the requested action, the decision subtask takes a relatively long time to complete.

```
PilotActs
Detection_Random_Variable
0
2.0
Decision_Random_Variable
3
0.0 0.0
Reaction_Random_Variable
3
0.0 0.0
```

The pilot is anticipating the ATC's response. Therefore, we assume that the pilot takes no time to perform the decision and reaction subtasks.

INPUT PARAMETERS AND METRICS

Our previous report described the scenario's physical input parameters and output metrics in detail. We have not changed these values for the current version.

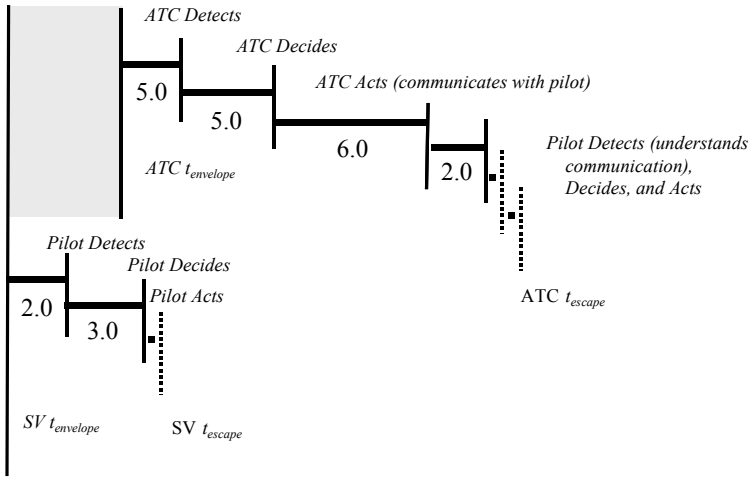
Chapter 5
Synthetic Vision: Collision Avoidance Scenario

This scenario models a 30-degree blunder by an intruding aircraft into the principal aircraft's final approach path. The scenario is based on the assumption that both aircraft are on final approach to a pair of independent, parallel runways, and are at level flight. Further, we assume that the aircraft start at level flight, at the same altitude and speed. This example is contrived; it is highly unlikely that aircraft would fly in this particular configuration by design. As a result, the probabilities of collision that we generate do not have meaning other than as a method for comparing the relative safety of the variant (SV and ATC monitoring) and the baseline (ATC monitoring only) scenarios. This method of comparison is consistent with the objectives of this study and reflects our approach to all of the scenarios. The results for this scenario indicate a significant potential for improving safety if SV is available.

TIMELINE MODEL

The timeline model for the collision avoidance scenario is rather simple (Figure 5-1). It resembles the SV scenario for terrain avoidance. Two series of tasks are parallel. Both series follow the response to the information from a specific piece of monitoring equipment. The following time values are the nominal expected values, which vary stochastically. In the baseline series, the ATC takes 5 seconds to detect a problem, 5 seconds to decide what to do about the problem, and 6 seconds to inform the pilot of the problem. The pilot takes 2 seconds to understand the communication, and no time to act on the information by initiating an emergency turn-climb. When synthetic vision is available, the possibility exists that the pilot will detect the intrusion himself, decide that he should do something about it, and act by initiating a turn-climb.

Figure 5-1. Timeline Model for the Collision Avoidance Scenario with Synthetic Vision

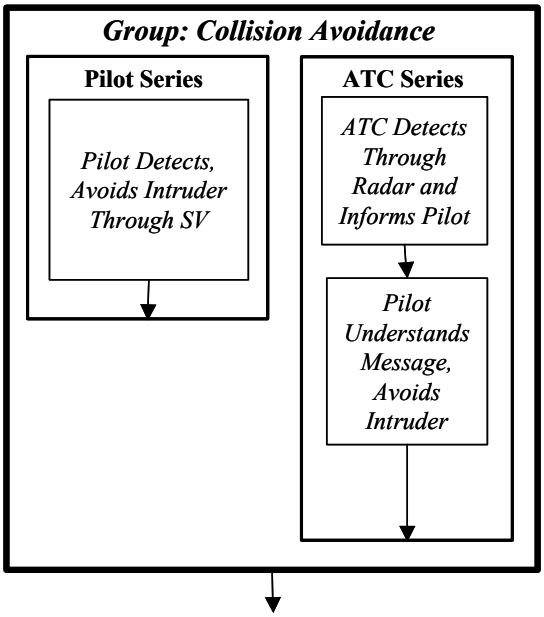


The only difference between this scenario and the terrain avoidance scenario, from a human factors perspective, is that the timeline for this scenario has longer sub-task times. We assume that these times will be longer because the situation is more complicated: rather than a single aircraft moving towards a large, stationary target, we have two small, fast-moving targets on a potential collision course.

ACTION SEQUENCE MODEL

The action sequence model converts the timeline above into a sequence of one group with two series (Figure 5-2). This model is practically identical to the action sequence model for the terrain avoidance scenario. The only difference is that "terrain" has been replaced with "intruder." We contend that the human factors for both of these scenarios are quite similar; the geometry and physics of these scenarios make them different.

Figure 5-2. Action Sequence Model for Turbulence Avoidance and Mitigation Scenario



ACTION SEQUENCE FILE

This file looks quite similar to the terrain avoidance scenario's file. However, the times for each subtask differ slightly. The differences in times are caused by the increased complexity of diagnosing a potential mid-air collision instead of diagnosing a potential terrain collision. Our verification and validation committee have reviewed and accepted the times but the Synthetic Vision Project Office has not yet reviewed them.

```
CollisionAvoidanceSequence
NUMBER_OF_GROUPS
1

GROUP_LIST

CollisionAvoidanceGroup
NUMBER_OF_SERIES
2

SERIES LIST

ATCSeries
NUMBER_OF_TASKS
2

TASK_LIST

ATCDetectsAndComms
Detection_Random_Variable
```

```
0
5.0
Decision_Random_Variable
0
5.0
Reaction_Random_Variable
0
6.0

PilotCommsAndMoves
Detection_Random_Variable
0
2.0
Decision_Random_Variable
3
0.0 0.0
Reaction_Random_Variable
3
0.0 0.0

PilotSeries
NUMBER_OF_TASKS
1

TASK_LIST

PilotDetectsAndMoves
Detection_Random_Variable
0
2.0
Decision_Random_Variable
0
3.0
Reaction_Random_Variable
3
0.0 0.0
```

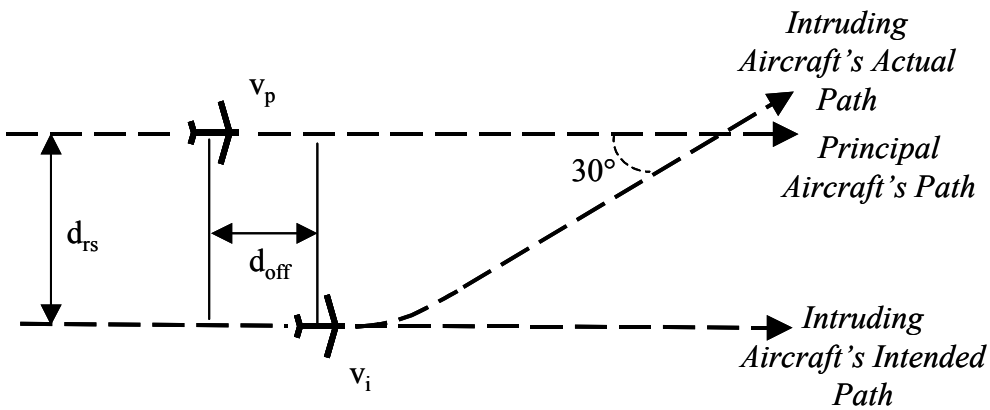
Second task in the ATC series (We assume that as soon as the pilot understands the communication from the ATC [the result of the first task], he initiates an escape maneuver.)

Only task in the pilot series (The pilot reacts to an intrusion that his SV equipment reveals. In this situation the pilot takes a little while to decide that a problem exists. As soon as the problem is apparent, however, he reacts to correct it.)

INPUT PARAMETERS AND METRICS

Two aircraft are used for this scenario: a principal aircraft and an intruding aircraft. The principal aircraft flies a straight line at a constant speed v_p, similar to an aircraft flying a final approach after being aligned with the runway. The intruder flies a straight line parallel to the principal aircraft at a speed v_i, then veers into the principal aircraft's path at a 30-degree angle. The initial separation distance between the two aircraft is d_{is}; the initial offset distance between the two is d_{off}. The scenario is illustrated in Figure 5-3.

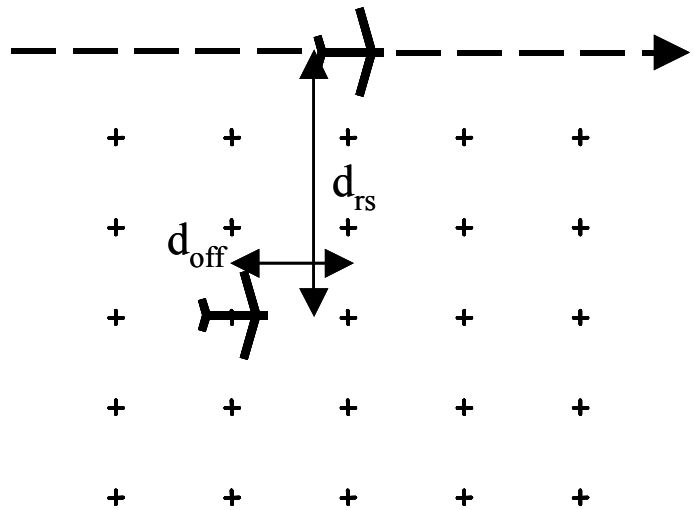
Figure 5-3. Major Input Parameters for Collision Avoidance Scenario



If the pilots of either aircraft do not take corrective action, a collision will occur when both aircraft reach the intersection point of their respective paths at the same time. Both distances and both speeds affect the likelihood of a collision. Of the four parameters, the distance parameters are most interesting, particularly the nominal initial separation distance. The separation distance represents an important tradeoff of safety and capacity. The larger the distance, the safer it will be to perform independent parallel operations. The smaller the distance, the greater the number of airports that will be able to use independent parallel approach procedures, and the greater the capacity of those airports. The speeds of the aircraft are less important, largely because approach procedures typically call for aircraft to fly at the same speed at the same point in the approach.

Understanding the trade off between safety and capacity when choosing a separation distance is critical. Therefore, we simulate several nominal separation distances. At each separation distance, we position the intruder within a range of offset distances. We also perturb the initial separation distance from the nominal separation distance by a random amount for each simulation. This strategy results in testing a grid of relative starting locations for the aircraft in the simulation (Figure 5-4). Each grid point represents a bin of initial conditions for the simulation. By running several simulations with and without synthetic vision on board the principal aircraft, we can compute the collision probabilities of each bin. The figure shows those collision bodies by bin and overall for each nominal initial separation distance in 500–foot increments from 1,500 to 5,000 feet.

Figure 5-4. Grid of Relative Starting Locations for Aircraft



We could painstakingly produce a similar set of contours using analytic calculations, if the performance of the instrumentation is fixed, the initial locations and the speeds are exact, and the time that the principal aircraft begins his escape maneuver and the geometry of the maneuver are known. Such an exercise would produce a solution to specific conditions, leaving all other variables unexplored. Thus, a simulation gives a much more informative set of results.

In addition to establishing the initial conditions of the simulation, we also must enter information about the simulation's action envelope and incident envelope (Figure 5-5). The action envelope is the volume inside of which the intruder should be viewed as a threat. When the intruder enters the action envelope, the ATC and pilot of the primary aircraft should act immediately to evade the intruder. The user can artificially expand or constrict the envelope to simulate external conditions that affect the actors' ability to monitor their situation. If the intruder flies into the incident envelope, then the evasive action has failed.

Figure 5-5. Collision Avoidance Scenario's Envelopes

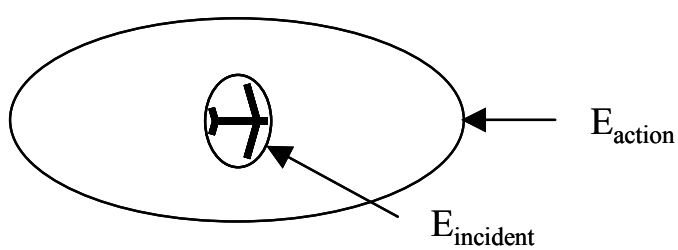


Figure 5-5 represents both envelopes conceptually as ellipsoids. In fact, they can take on any solid shape, ranging in complexity from a sphere to a freeform volume. We model the action envelope as a half space whose border is the plane *Y = actual initial separation distance*. What this means is that the intruder will trigger the action sequence that will lead to evasive maneuvers as soon as it starts its blunder

towards the principal aircraft. We model the incident volume as a sphere with a 500–foot radius. If the intruder intersects that sphere at any time in the simulation, an incident (for our purposes, a collision) occurs.

INSTRUMENTS

We model the SV detector and the air traffic controller's primary radar similarly to the detectors in the terrain avoidance scenario. The SV detector updates every simulation cycle, and has an absolute distance-error model with a 30–foot radius. That means that any location shown by this instrument may be as far as 30 feet away from the true location. The ATC's primary radar updates every 5 seconds, and has a relative distance error model of two percent of the distance of the measured location. That means, for example, that the radar's accuracy in providing the location of an object 25 miles away is within a quarter of a mile; within 12.5 miles away, it is accurate to within an eighth of a mile.

As we found in the terrain avoidance scenario, the difference in accuracy between the two instruments is usually much less important than the difference in update rates. The 5-second sweep time of the ATC's primary radar means that he may receive the first indication of a problem as much as 5 seconds after the pilot if the pilot has a SV system onboard.

OTHER SCENARIO CHARACTERISTICS

The standard escape maneuver for this scenario is a turn or a turn-climb. Because both the baseline and SV cases use the same escape maneuver, the specific maneuver we choose is immaterial. The safety improvement, or the change in safety from the baseline case to the synthetic vision case, should be the same if the maneuver is a turn or a turn-climb. We assume that the escape maneuver will a turn and climb at the maximum turn and throttle rate available from the RFS aircraft model.

Because this is a final approach scenario, we allow the user to specify a speed and altitude reduction for both aircraft. The changes in speed and altitude have little effect on the collision geometry, however, because both aircraft would reduce their speed and altitude at roughly the same rate. The changes may affect the geometry of the principal aircraft's escape maneuver, however. For this scenario, we model both aircraft as flying level and decelerating from 210 to 170 knots.

OUTPUT METRICS

The simulation provides the following output data for each simulation trial:

- *Principal aircraft initial location*: initial location of the principal aircraft in relation to the runway threshold location.

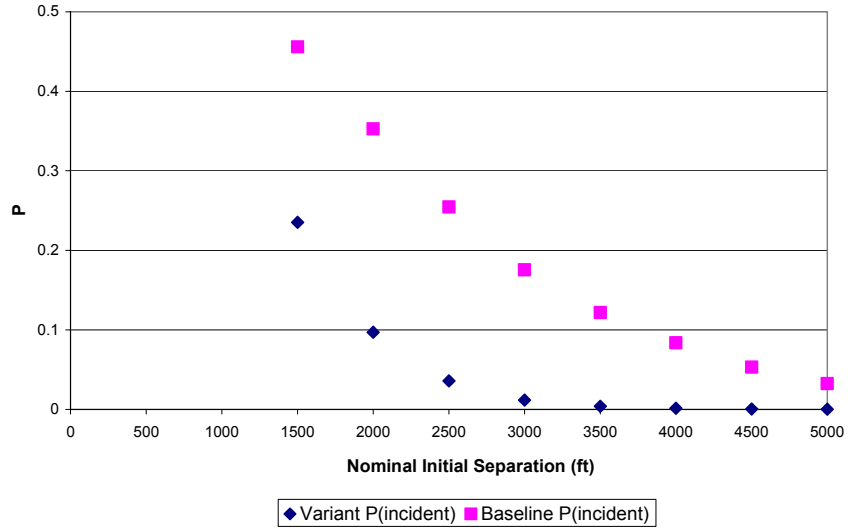
- *Intruding aircraft initial location*: initial location of the intruding aircraft in relation to the runway threshold location.
- *Total human-action sequence time*: the time required to progress from the scenario envelope to the time when the principal aircraft begins its escape actions.
- *Instrument that prompts actions*: tells which instrument initiated the human actions that lead to the escape actions, if any: the ATC primary radar or the synthetic vision system. We assume that visual indication is unavailable because of poor visibility.
- *Minimum separation distance*: the minimum separation distance between the two aircraft during the trial.
- *Collision*: determines whether the intruder comes within a user-specified incident distance of the principal aircraft. If it does, the simulation presumes that a collision occurs.

RESULTS

We ran 100,000 trials at each nominal separation distance, for both the variant (SV and ATC monitoring) and baseline (ATC monitoring only) cases. The main objective of these runs is to compute the probability of an incident (collision) for each separation distance, and to measure the safety effects of the SV technology. We measure the safety effects by computing the safety gain, which we define as the baseline probability of an incident divided by the variant probability of an incident.

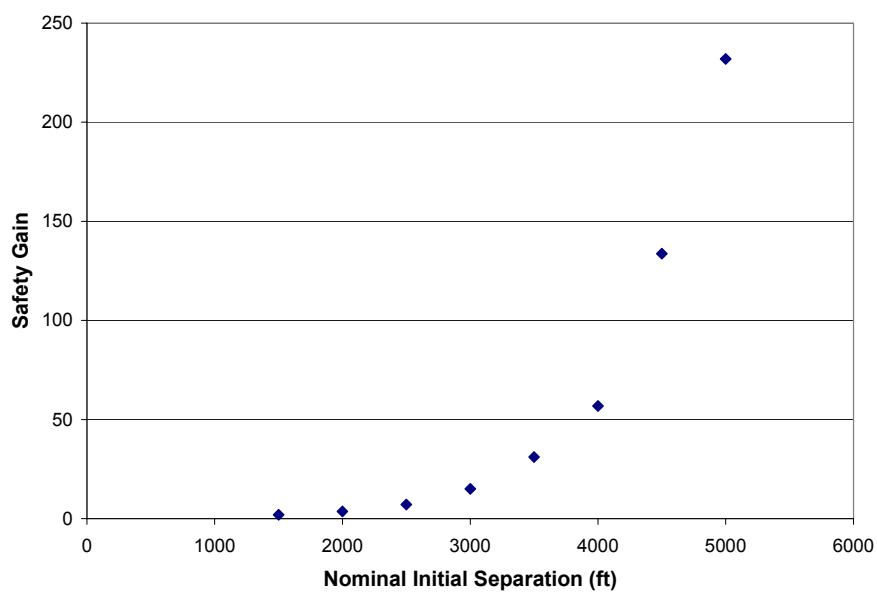
Figure 5-6 shows that synthetic vision greatly reduces the incident probability at the smaller separation distances. The incident probabilities also diminish faster when the synthetic vision is available than when it is not. The incident probabilities decline because the process of discovering the intruder's blunder, diagnosing it as such, and acting on that diagnosis is much simpler for the pilot with SV than it is for the ATC. SV gives with the information directly to the pilot that he needs to act on, eliminating the need for communications between the ATC and the pilot. In addition, we believe that an SV display will usually have more current information than the ATC's primary radar (depending on when the pilot reads the information in relation to the last radar sweep), and receives it in a simpler format than the ATC.

Figure 5-6. Incident Probabilities for Collision Avoidance Scenario



The ATC's less capable instrument forces him to work harder to diagnose a problem and decide what actions he must take to resolve the problem. Once he does decide what to do, he must communicate that decision to the pilot, who must understand what the ATC wants, and then do it. The additional complexity means that the duration for the required human actions in the baseline case has both a higher expected value and a larger variance than the duration for the required human actions in the variant case. Both characteristics favor the synthetic vision technology over the baseline technology. The safety gain is shown in Figure 5-7.

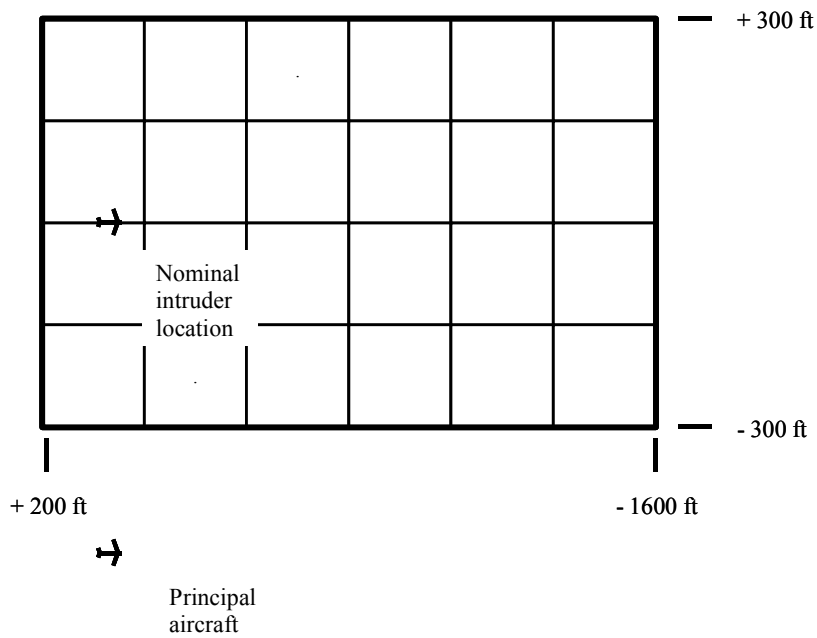
Figure 5-7. Safety Gain of the Variant Case over the Baseline Case



For each trial, we perturbed the intruding aircraft's initial location laterally using a uniform random variable with a range of -300 feet to +300 feet. By convention, a negative perturbation means that the intruder and principal aircraft are closer than the nominal separation distance. We also perturbed the intruder longitudinally using a uniform random variable with a range of -1,600 feet to +200 feet. By convention, a negative longitudinal perturbation means that the intruder starts the simulation closer to the airport than the principal aircraft. We chose a grid of these dimensions because preliminary testing showed that it includes, for each nominal separation distance, a set of initial locations that will result in an incident if no action is taken.

We partitioned the set of all potential initial locations for the intruder into 24 uniform bins (Figure 5-8). Beccause the aircraft's initial location results from random draws from two uniform distributions, all locations inside the 24-bin area are equally likely. In reality, this distribution does not represent actual conditions. All longitudinal perturbations would be equally likely if the aircraft are controlled independently. However, according to the pilot from the validation and verification team, the lateral separation would be distributed tightly about the nominal separation distance. Why, then, did we model the lateral separation variance with a uniform distribution? Because the simulation compares a variant case with a baseline case, the actual distribution does not matter as long as it is the same for both cases. The uniform distribution is the most informative choice because it shows what subsets of the set of possible initial locations cause the greatest hazard. The uniform distribution also shows how those subsets shift as the nominal separation distance changes.

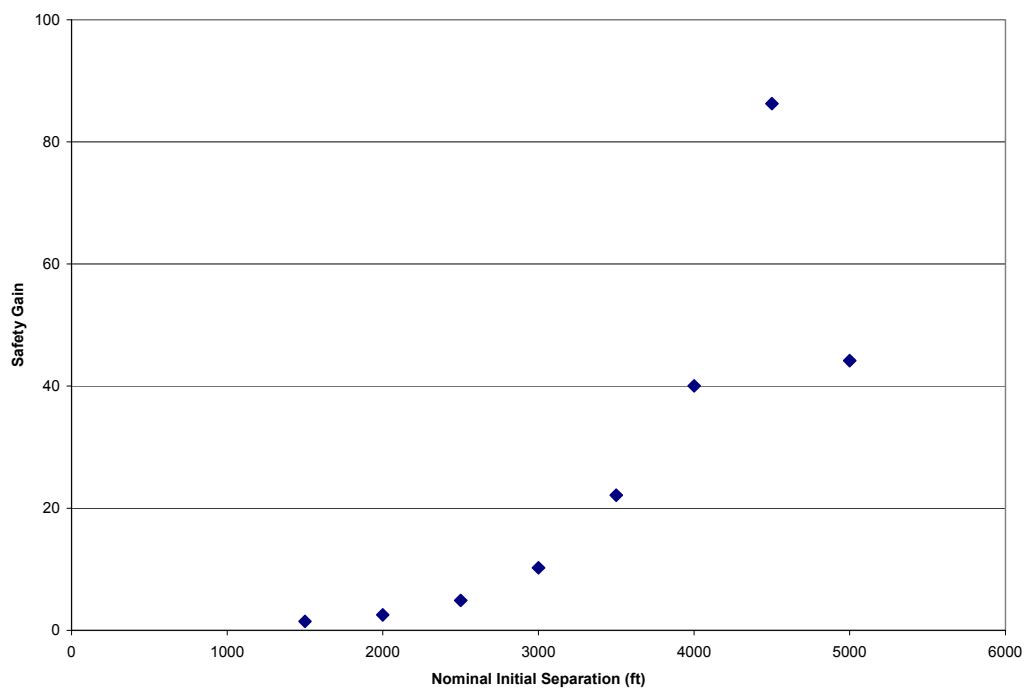
Figure 5-8. 24-Bin Partition of Intruder Initial Location (not to scale)



By creating these bins of initial intruder locations, identifying the subset that causes the greatest hazard for the principal aircraft is possible. As expected, this subset falls in the bottom row of the diagram in Figure 5-8; that row is closest to the principal aircraft. As the nominal separation increases from 1,500 feet to 5,000 feet, the location of the most hazardous bin shifts toward the right (further away from and ahead of the principal aircraft). This shift also is as expected: as the initial lateral separation increases, the intruder must lead the principal aircraft by a greater and greater distance to ensure that their paths will intersect. This study of the hazard geometry gives us greater confidence that the simulation models the scenario adequately.

Computing the safety gain for any one of the subsets in the set of 24 is possible. Figure 5-9 shows the safety gain at the subset of initial intruder locations that represent the greatest hazard to the principal aircraft at each nominal initial separation. With the exception of the data point at the 5,000-foot nominal separation distance, the trend looks similar to that of Figure 5-7. The values of the safety gains are smaller, but that is expected. As Figure 5-7 shows, the safety gain rises at a greater than linear rate with nominal initial separation. By restricting our observation to the most hazardous subset of initial intruder locations, we reduce the effective distance between the intruder and the principal aircraft.

Figure 5-9. Safety Gain at Most Hazardous Subset of Initial Intruder Locations



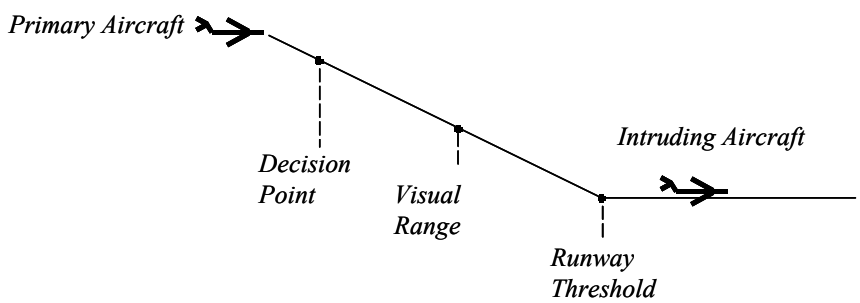
The data point at the 5,000-foot nominal initial separation is interesting because that data point does not appear to follow the trend. This apparent aberration occurs

because at an initial separation of 5,000 feet, we are pushing the limit of the simulation's ability to generate a significant number of collisions for the variant case. We used 100,000 trials at each separation distance; at the 5,000-foot data point, those 100,000 trials produced 14 total collisions, eight of which originated from the most hazardous subset. A loss of only three or four of those recorded collisions would bring the data point back in line with the trend. Because the probability of a collision is so small, each additional collision has an exaggerated affect on the safety gain.

Chapter 6
Synthetic Vision: Runway Incursion Avoidance Scenario

In this scenario, a principal aircraft attempts to land on a runway. Another aircraft or ground vehicle occupies that runway for an unspecified amount of time during the principal aircraft's descent. If the intruder remains on the runway when the principal aircraft reaches a final decision distance, the principal aircraft should not land (Figure 6-1).

Figure 6-1. Runway Incursion Scenario



Two types of runway incidents can occur. In the first, an aircraft is waiting on the runway to take off when the principal aircraft is cleared to land. The aircraft on the runway may or may not take off on time. In this case, both the controller and the pilot are presumably aware of the aircraft on the runway. The most interesting feature of such a scenario, therefore, is the decisions that the ATC or the pilot must make to determine when this normal condition becomes a problem requiring action. The instruments used to monitor the situation are much less significant. Therefore, we do not model this particular runway incursion scenario. Instead, we model the second type of runway incident in which the intruder crosses the runway by mistake as the principal aircraft attempts to land. In this case, neither the principal aircraft nor the ATC know about the aircraft beforehand, and must detect the problem on their instruments before acting. In this scenario, the instruments have a major influence on the ability of the principal aircraft to begin its escape actions on time to avoid a problem.

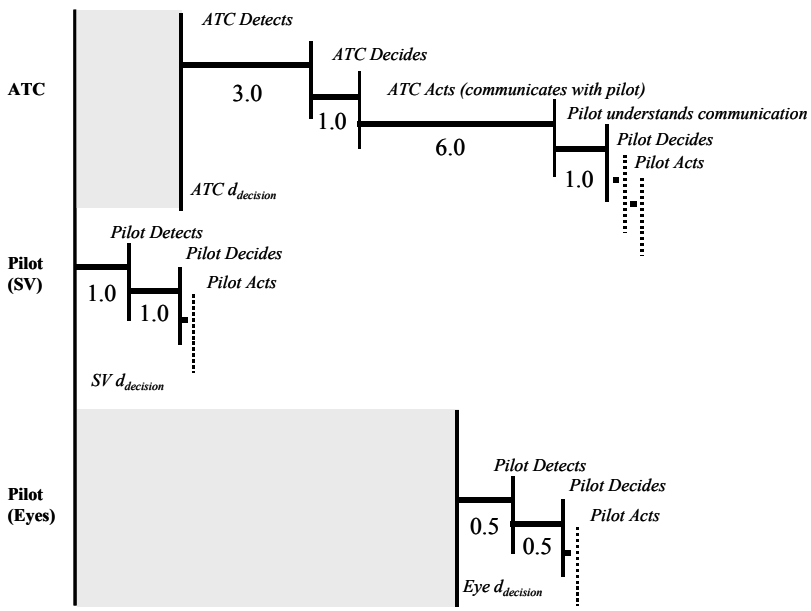
Both the air traffic controller and the pilot may have equipment at their disposal for monitoring the runway. For the controller, the tools include airport surface detection equipment (ASDE) radar or an ASDE-based advisory system. The pilot may have a synthetic vision-based view of the runway. The pilot also has natural vision, but if visibility is limited, natural vision may not give him sufficient time to see the intruder and go around. In this scenario, the baseline case is one in

which the ATC's ASDE radar is the sole means of detecting a potential runway incursion, and he must communicate the problem to the pilot before actions can occur. In the variant case, the pilot also can detect the problem by using an on-board SV system.

TIMELINE MODEL

The timeline model is similar to the other synthetic vision scenarios: several simple series of tasks operate in parallel. The SV series begins when the aircraft reaches the decision distance. The ATC must monitor the aircraft on the ground with ASDE radar and the principal aircraft with primary air traffic radar. As a result, the ATC series begins 1 to 5 seconds after the aircraft reaches the decision distance (the first primary radar update inside of the decision distance). The natural vision series begins when the aircraft is inside visual range of the runway. The following time values are nominal expected values, which vary stochastically.

Figure 6-2. Timeline for the Runway Incursion Scenario with Synthetic Vision

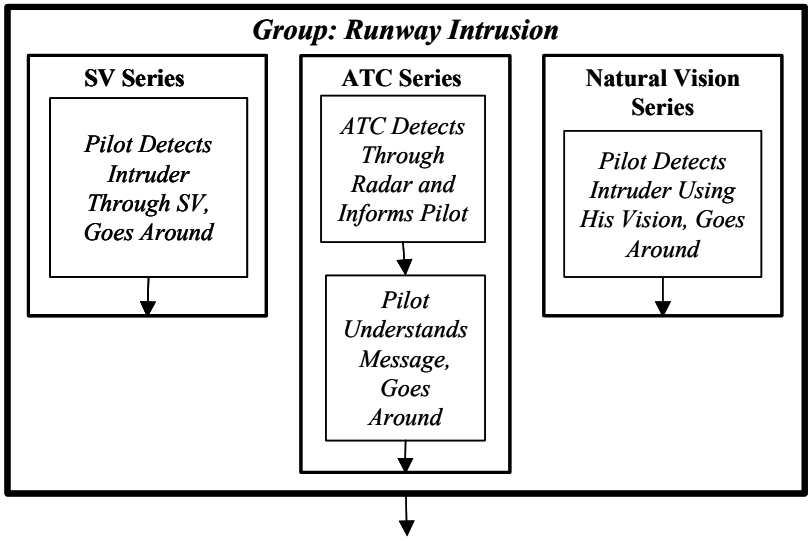


The timeline model, as shown above, is based on the assumption that the problem will be recognized rapidly, particularly by the pilot. This timeline handles the case of an aircraft on the runway, trying to take off. Because both the pilot and the ATC knew the aircraft on the runway was there at the start of the final approach, they are aware of the possibility that it might still be there when the principal aircraft reaches the decision distance. A timeline that covered the case of an intruder crossing the runway without clearance would have longer detection and decision times, particularly for the ATC and SV series.

ACTION SEQUENCE MODEL

The action sequence model converts the timeline into one group with three series (Figure 6-3). This model is almost identical to the action sequence model for the terrain avoidance scenario. The only difference is that "terrain" has been replaced with "intruder." We contend that the human factors for both of these scenarios are similar; the geometry and physics of these scenarios make them different.

Figure 6-3. Action Sequence Model for Runway Incursion Scenario



ACTION SEQUENCE FILE

This file looks quite similar to the terrain avoidance scenario's file. However, the times for each subtask differ slightly. The differences in times are caused by the increased complexity of diagnosing a potential mid-air collision instead of diagnosing a potential terrain collision. Our verification and validation committee have reviewed and accepted the times but the Synthetic Vision Project Office has not reviewed them yet.

```
RunwayIntrusionSequence
NUMBER_OF_GROUPS
1

GROUP_LIST

RunwayIntrusionGroup
NUMBER_OF_SERIES
3

SERIES LIST

ATCSeries
NUMBER_OF_TASKS
```

```
2

TASK_LIST

ATCDetectsAndComms
Detection_Random_Variable
0
3.0
Decision_Random_Variable
0
1.0
Reaction_Random_Variable
0
6.0

PilotCommsAndMoves
Detection_Random_Variable
0
1.0
Decision_Random_Variable
3
0.0 0.0
Reaction_Random_Variable
3
0.0 0.0
SyntheticVisionSeries
NUMBER_OF_TASKS
1
TASK_LIST

PilotDetectsAndMoves
Detection_Random_Variable
0
1.0
Decision_Random_Variable
0
1.0
Reaction_Random_Variable
3
0.0 0.0

NaturalVisionSeries
NUMBER_OF_TASKS
1

TASK_LIST

PilotDetectsAndMoves
Detection_Random_Variable
0
0.5
Decision_Random_Variable
0
0.5
Reaction_Random_Variable
3
0.0 0.0
```

The pilot's response is particularly quick because the message the ATC gives is relatively common and, therefore, not unexpected.

This task amounts to an extremely rapid reflex to the what the pilot sees: another aircraft on his runway. Because the pilot's focus is on landing the aircraft on the runway, we expect him to discover the problem almost immediately, and decide what to do just as quickly.

INPUT PARAMETERS AND METRICS

This scenario requires a full, three-dimensional approach path. Unlike the other SV scenarios, altitude is a significant parameter. We have coded the aircraft to fly on a due west heading to the runway, but the user can choose its initial altitude, initial distance from runway threshold, initial speed, glide slope intercept speed, and landing speed. The simulation computes the glide slope intercept position on the assumption that the aircraft flying in a level flight until it intercepts the glide slope. The user is responsible for choosing an initial altitude, speed, and distance from the runway that will enable the aircraft to intercept the glide slope at the proper speed and location.

The user can specify several other variables that establish the gross dynamic characteristics of the aircraft, and the location of the intruder's taxiway as a distance from the runway threshold. The following are the remaining input parameters that are most critical for determining the outcome of a simulation trial:

- *Simulation envelope*: the distance from the aircraft to the runway threshold at which the human actors should recognize a problem and go around. The user should consider this envelope as a distance at which the pilot must go around by procedure: neither the pilot or the ATC has discretion to continue to fly the approach if the intruder is on the runway at the simulation envelope.

- *Incident altitude*: the altitude (with respect to the runway threshold) at which the simulation registers an incident. (In this scenario, an incursion.) We allow this altitude to change to handle situations in which large intruders occupy the runway near the threshold, effectively causing a hazard if the principal aircraft flies any lower than the intruder's height.

- *True ground visibility*: the distance that the pilot can see ahead.

- *Intrusion start time*: this is a pair of parameters: the user must specify a range, in seconds from the start of the simulation, for the time at which the intruder first occupies the runway. The simulation will determine the actual start time for the intrusion by using a uniform random variable defined on the provided range.

- *Intruder crossing speed*: the amount of time that the intruder should remain on the runway. This speed is based on a user-specified random distribution. The speed can be as slow or as fast as the user likes. We use a very slow speed to force the intruder to be on the runway at the simulation envelope, and to remain on the runway throughout the simulation trial.

OUTPUT METRICS

The simulation provides the following output data for each simulation trial:

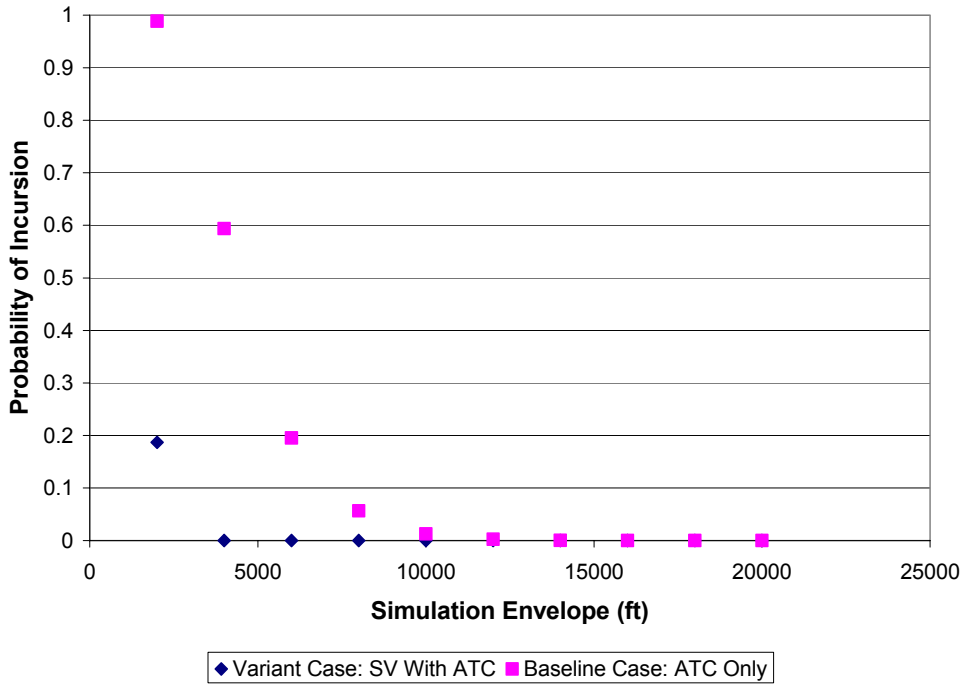
- *Total human-action sequence time*: the time required to progress from the scenario envelope to the time when the principal aircraft begins its escape actions.
- *Instrument that prompts actions*: tells which instrument initiated the human actions that lead to the escape actions, if any: the ATC ASDE radar, the synthetic vision system, or the pilot's own eyes (once the aircraft comes into visual range with the intruder on the runway).
- *Intruder on/off the runway times*: is the time, in seconds, that the intruder first enters and exits the runway.
- *Final minimum altitude*: the principal aircraft's final minimum altitude (relative to the runway threshold) during the trial.
- *Incursion*: determines whether the principal aircraft comes within a user-specified incident altitude relative to the runway. If it does, the simulation presumes that an incursion occurs. (The simulation does not differentiate between incursions and collisions.)

RESULTS

We ran the baseline case (ATC monitoring with ASDE radar) and the variant case (ATC monitoring with ASDE radar and pilot monitoring with SV) at ten evenly spaced envelope distances starting at 2,000 feet and ending at 20,000 feet. In every set of trials we ran, we specified that the intruder be on the runway throughout the entire trial. We based the number of trials per case and range of envelope from 10,000 to 500,000 feet on preliminary simulation and analysis that gave gross estimates of the probability of an incursion.

Our goal was to produce safety gain values similar to the ones we calculated in the other SV scenarios. However, we could not compute a range of safety gains for this scenario because the synthetic vision case did not produce an incursion beyond the 2,000-foot envelope (Figure 6-4). In comparison, the baseline probability of incursion remains above 1.0^{E-3} until the envelope increases above 12,000 feet. Further, the baseline case had at least one incursion at every simulation envelope below 20,000 feet. For the 2,000-foot simulation envelope, the SV safety gain is 5. As with other synthetic vision scenarios, the effect of the SV instrumentation was to reduce both the expected duration of required corrective actions and the dispersion of that duration.

Figure 6-4. Comparison of Incursion Probability Between Baseline and Variant Cases



Chapter 7
System-wide Accident Prevention Runway Incursion Avoidance Scenario

System-wide accident prevention (SWAP) can affect a tactical safety scenario in many ways. It can

- improve the response time to a hazard by one or more of the human actors,
- reduce the probability of a hazard through better practices, and
- increase the possibility that the human actors will take the optimal course of action. This may mean avoiding mistakes, or choosing the best of several acceptable actions.

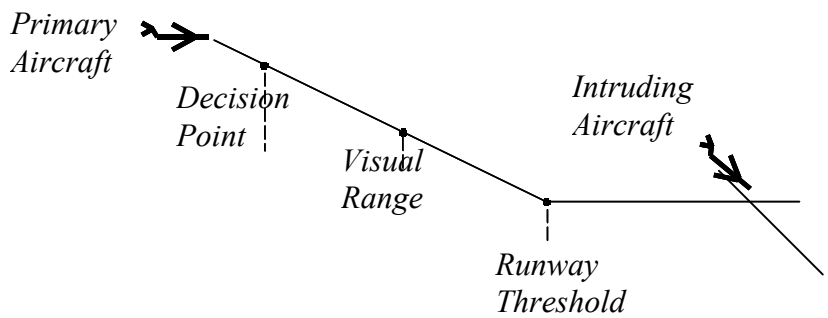
The SWAP Project Office gave us a set of narrative scenarios that describe hypothetical flights in which circumstances, operational errors, and poor decision-making result in accidents. Essentially, these narrative scenarios describe baseline situations in which SWAP is intended to be beneficial. However, the narrative gives no description of a specific item (such as a decision point or one of the error sources) that SWAP is targeting, nor does it state a goal for a SWAP benefit (e.g., "Decision A is made incorrectly X percent of the time and SWAP will reduce this to Y percent."). An additional mismatch is that the SWAP-provided scenarios describe complete flights, spanning many minutes, if not hours, whereas the types of scenarios that our model is suited for are tactical, lasting several seconds or minutes. Regardless of these difficulties, we did modify our software so that the structure enables assessing SWAP benefits if or when appropriate values for the parameters are available.

Without a concrete example of a SWAP-instituted practice that describes its effects in a way that we could model in a safety scenario, we were limited to creating a notional scenario that shows how SWAP fits into our software design. Modeling changes in response times to a set action sequence is trivial; the user can change these times in the human action sequence file. Further, in our approach, the actual probability of a hazard is seldom relevant to a scenario we simulate. Rather, most of our simulations start with inserting a hazard; thus, the probability of a hazard is 1.0. Therefore, we have chosen to supplement our runway incursion scenario with an enhanced human action sequence model that accommodates a human actor, the ATC, making a mistake.

In this scenario, a primary aircraft attempts to land on a runway. Another aircraft (the intruder) crosses that runway as the primary aircraft lands. If the intruder remains on the runway when the primary aircraft reaches a final decision distance,

the primary aircraft should go around (Figure 7-1). If the ATC sees this situation develop, he should warn the primary aircraft of the problem and order a go-around. This scenario is identical in most respects to that of the original runway incursion avoidance scenario. In fact, we run that scenario and this one with the same input files. The user can choose which scenario to run by inserting a set of input flags in those files.

Figure 7-1. Runway Incursion Scenario for SWAP



A controller who has received the proper amount and type of training (presumably as prescribed by a SWAP initiative) should be able to determine when an intruder blunders onto a runway. He should be able to do so quickly and without error. A controller who is less familiar with the airport and its procedures may mistakenly believe that the intruder is crossing a different runway—one that is free of landing traffic. The ATC may catch his error after a few seconds, but this would delay his ability to relay instructions to the pilot in the landing aircraft. How would this delay affect the safety of the landing aircraft?

In this scenario, we focus entirely on the ATC's actions. The pilot in the landing aircraft is not an independent actor, and will not discover the potential incursion without ATC intervention. The ATC does not know the intruding vehicle's intentions. The durations of the tasks in the action sequence reflect this. In either case, the action sequence starts when the aircraft reaches the simulation envelope.

In this scenario, we do not compare a baseline and variant technology suite; rather, we compare the correct set of ATC actions with a set of ATC actions that start with a mistaken runway identification. The ATC has an opportunity to discover the mistake after a given amount of time (which is varied stochastically) and correct it.

DECISION MAKING AND THE HUMAN-FACTORS ACTION SEQUENCE

Decisions with multiple resulting actions can be incorporated into our human factors model in three ways. The first, and simplest, way to model a decision is to execute a decision external to the human-factors action sequence, and use that decision to decide how the aircraft responds after the sequence is over. For example,

the aircraft may need to change speed and altitude in response to some hazard. After executing a human-factors action sequence to determine that the hazard exists, the pilot decides what to do: the right answer is to change direction and altitude, but he also could change direction only, change speed only, or do nothing.

The second way to model a decision is to build different task series in a group structure in the human-factors action sequence. An initial decision is made, then only the series that corresponds to that decision is activated. Unlike the method discussed above, this method enables users to specify different tasks of different lengths, depending on the situation. For example, if the pilot can decide between reducing speed or reducing speed and reporting a problem to the ATC, this method allows the pilot to take longer to complete his actions if he chooses to contact the controller. The user also can specify an action sequence in which one series represents the actions after a good decision, and another series represents initial actions and corrective actions after a bad decision (with an implied recognition of the bad decision).

We use this type of action sequence in this scenario. The ATC recognizes that the intruder is on the runway, or he mistakenly believes he is on a clear runway. If he makes a mistake, he recognizes the mistake after some amount of time and corrects it. The result is the same—the ATC warns the primary aircraft about the runway intrusion. The time taken to reach that result could vary significantly, however.

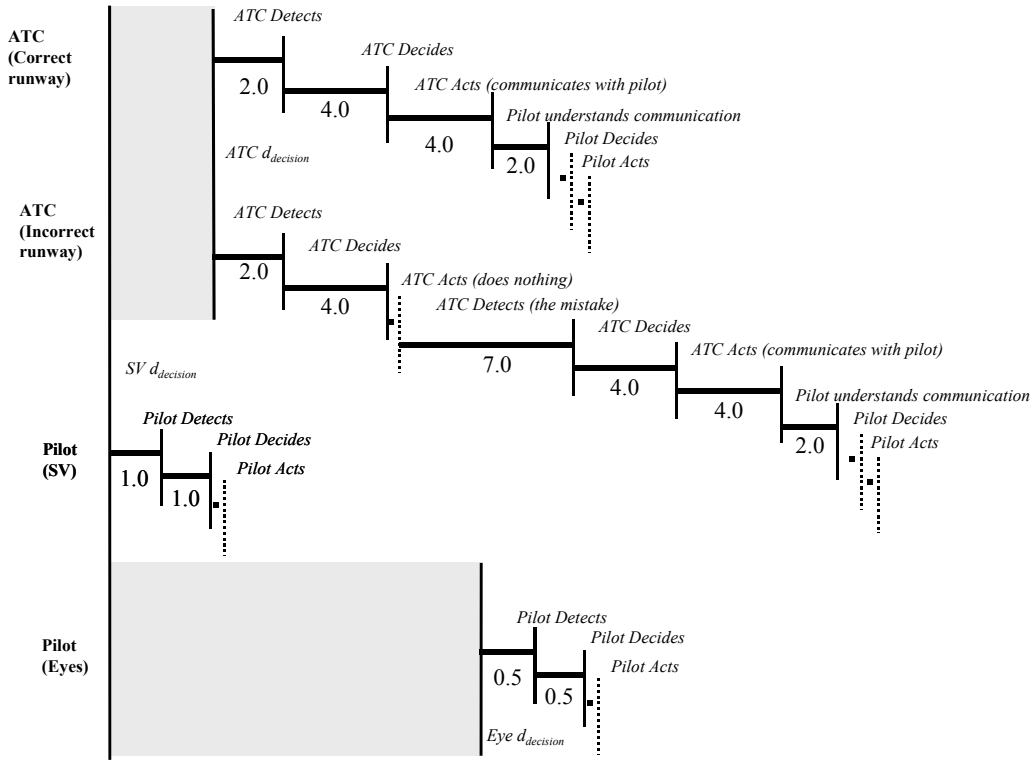
The third way to model decision making in human-factors action sequences is to build a set of action sequences that combine with decisions to form a branching, tree-like super-sequence. Decisions may correspond to task series; they also may correspond to entirely new action sequences. The super-sequence has the most flexibility for setting up a complex system of decisions and actions, but such systems may be too complex for a reasonable tactical scenario. We do not implement a super-sequence because its additional complexity is unnecessary for demonstrating how SWAP affects this runway incursion scenario. If such a super-sequence ever appears to useful in a new scenario, we can build one with our existing data structures.

TIMELINE MODEL

The timeline model is similar to the basic runway incursion scenario for synthetic vision. Although we don't use synthetic vision (or natural vision) in this scenario, we include SV and natural vision series in this model to make it easier to compare this SWAP-enhanced scenario with the original scenario (Figure 7-2). Two task series are associated with the ATC: one for when he correctly determines that the intruder is on an active runway, and one for when he incorrectly determines that the intruder is on a clear runway. The following time values are nominal expected values, which vary stochastically. If the ATC takes correct actions, he will need about 2 seconds to detect the problem, about 4 seconds to decide what action to take, and 4 seconds to act on that decision (contact the incoming pilot and tell him

to go around). If the ATC makes a mistake first, he will diagnose the hazard, then take about 7 seconds to catch his mistake, 4 seconds to decide what to do about it, and 4 seconds to communicate the problem to the pilot. In either case, the pilot needs about 2 seconds to react to the message and go around.

Figure 7-2. Timeline Model for the SWAP Runway Incursion Scenario



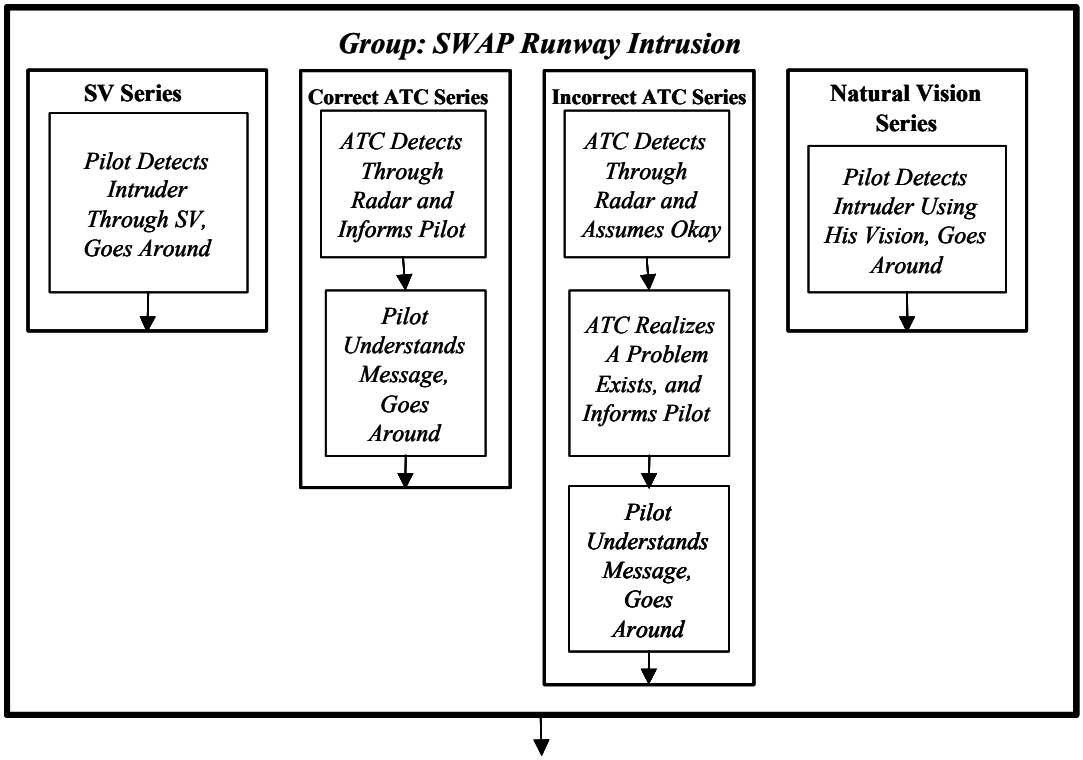
ACTION SEQUENCE MODEL

The action sequence model converts the timeline above into one group with four series (Figure 7-3). The presence of two possible ATC action series implies a right or wrong decision during the ATC's actions. The software models this decision by generating its outcome first, then activating the ATC series that corresponds to that outcome. The other ATC action series remains inactive.

The actual decision that leads to the right or wrong outcome is implied. That decision is external to the action sequence, and does not appear in the action sequence file. The scenario file holds the decision parameters. The actual probability of a wrong outcome may be quite small—a small fraction of a percent—for both the SWAP and non-SWAP baseline. The non-SWAP baseline probability may be several times higher than with SWAP. The numbers that we supply for these values are notional.

We do not use the SV or natural vision series in this scenario, but we show them in the action sequence model to make comparing the model with the original one easier.

Figure 7-3. Action Sequence Model for Runway Incursion Scenario



ACTION SEQUENCE FILE

The following file represents the action sequence model for the SWAP runway incursion scenario. The decision times, at present, are notional.

```
SwapRunwayIntrusionSequence
NUMBER_OF_GROUPS
1

GROUP_LIST

RunwayIntrusionGroup
NUMBER_OF_SERIES
4

SERIES LIST

ATCCorrectSeries
NUMBER_OF_TASKS
2

TASK_LIST
```

```
ATCDetectsAndComms
Detection_Random_Variable
0
2.0
Decision_Random_Variable
0
4.0
Reaction_Random_Variable
0
4.0

PilotCommsAndMoves
Detection_Random_Variable
0
2.0
Decision_Random_Variable
3
0.0 0.0
Reaction_Random_Variable
3
0.0 0.0

ATCIncorrectSeries
NUMBER_OF_TASKS
3

TASK_LIST

ATCIncorrectRunway
Detection_Random_Variable
0
2.0
Decision_Random_Variable
0
4.0
Reaction_Random_Variable
3
0.0 0.0

ATCCatchesProblem
Detection_Random_Variable
0
7.0
Decision_Random_Variable
0
3.0
Reaction_Random_Variable
0
4.0

PilotCommsAndMoves
Detection_Random_Variable
0
2.0
Decision_Random_Variable
3
```

The pilot's response is particularly quick because the message the ATC gives is relatively common and, therefore, not unexpected.

```
0.0 0.0
Reaction_Random_Variable
3
0.0 0.0


SyntheticVisionSeries
NUMBER_OF_TASKS
1

TASK_LIST

PilotDetectsAndMoves
Detection_Random_Variable
0
1.0
Decision_Random_Variable
0
1.0
Reaction_Random_Variable
3
0.0 0.0

NaturalVisionSeries
NUMBER_OF_TASKS
1

TASK_LIST

PilotDetectsAndMoves
Detection_Random_Variable
0
0.5
Decision_Random_Variable
0
0.5
Reaction_Random_Variable
3
0.0 0.0
```

This task amounts to an extremely rapid reflex to the what the pilot sees: another aircraft on his runway. Because the pilot's focus is on landing the aircraft on the runway, we expect him to discover the problem almost immediately, and decide what to do just as quickly.

INPUT PARAMETERS AND METRICS

The input parameters for this scenario will be much the same as those of the synthetic vision runway scenario. Additional human factors information is in the action sequence file.

- *Decision distance*: the distance, along the ground from the aircraft to the head of the runway, at which the human actors should recognize a problem and go around.
- *Visual range*: the distance, along the ground from the aircraft to the head of the runway, at which the pilot can see the head of the runway. For this scenario, we assume that once the pilot can see the head of the runway, he can

see the entire runway. Therefore, if the user wants to simulate an intruder on the runway 1,000 feet from the head, he should reduce the visual range by 1,000 feet.

- *Intrusion start time*: this is a pair of parameters: the user must specify a range for the time, in seconds from the start of the simulation, at which the intruder first occupies the runway. The simulation will determine the actual start time for the intrusion by using a uniform random variable defined on the provided range.

- *Intrusion duration*: this is another pair of parameters that defines the range of a uniform random variable. A selection from this uniform random variable determines how long the aircraft will remain on the runway.

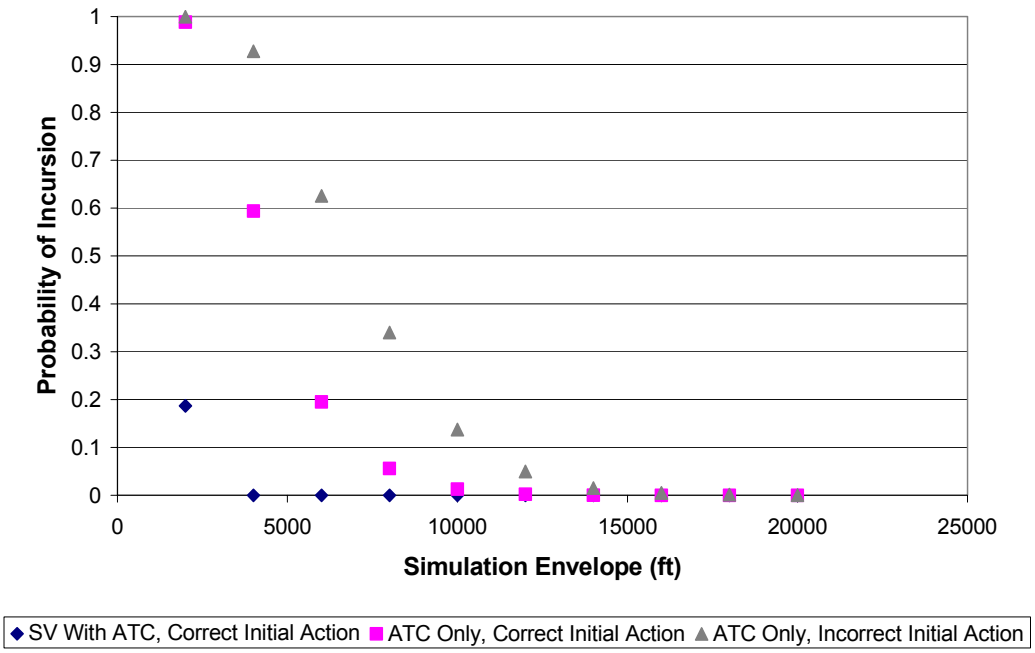
OUTPUT METRICS

This scenario gives the same output information as that of the original scenario described in Chapter 6.

RESULTS

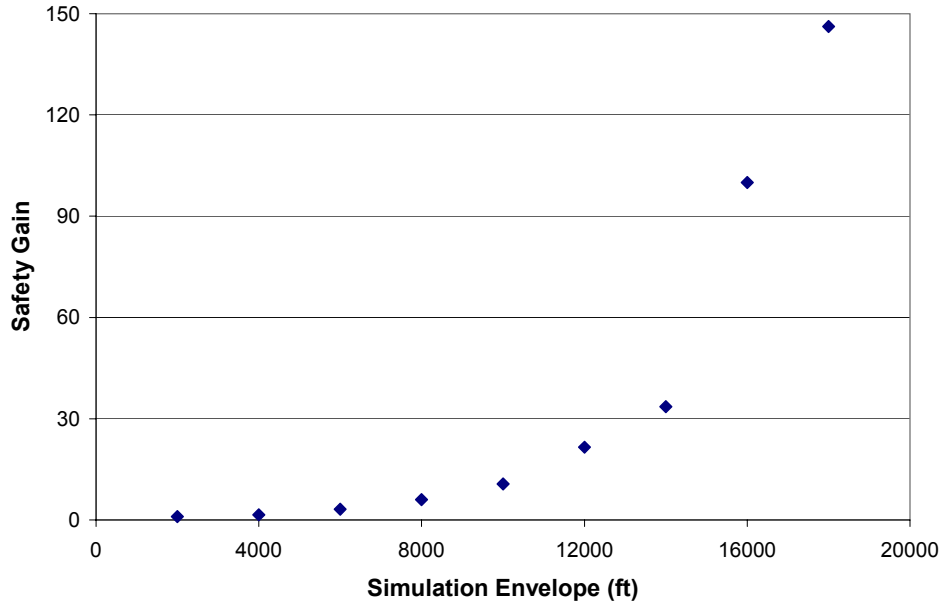
As expected, when the ATC makes the mistake as modeled, the mistake increases both the expected duration of the required actions and its dispersion. These changes dramatically increase the probability of a runway incursion. Figure 7-4 overlays a set of data in which the ATC makes this mistake on the results of the original scenario for avoiding a runway incursion (Figure 6-4). If the ATC makes a mistake at the start of the simulation, the chances are significantly higher that an incursion will occur than if he takes correct actions. For example, no incursions occurred at the 20,000-foot envelope when the ATC takes correct actions, but the probability of an incursion is 5^{E-4} at the same envelope when the ATC makes a mistake.

Figure 7-4. Comparison of Incursion Probabilities for Correct and Incorrect ATC Initial Actions (SV and Correct ATC Actions Shown for Comparison)



Because both ATC cases have a non-zero number of incursions for all runs below the 20,000-foot envelope, computing the safety gain of mistake-free actions relative to actions that include a mistake for every run with an envelope below 20,000 feet (Figure 7-5) is possible. Safety increases at a greater than linear rate as the simulation envelope increases. At the 16,000- and 18,000-foot envelopes, the correct initial actions case produces very few incursions, so we cannot use those numbers with any confidence; we can only conclude that the gains in safety at those envelopes roughly follow the trend of the gains at the smaller envelopes.

Figure 7-5. Safety Gain from Implementing Correct ATC Actions as Opposed to Incorrect ATC Actions



In a more realistic (albeit still notional) SWAP scenario, the ATC who has been trained under a SWAP initiative may still make the mistake we modeled. Obviously, even without such training, an ATC would not make the mistake we modeled every time. Each ATC would have a probability of making a mistake; presumably, both would be far smaller than 1.0. We modeled the cases in this way so the effects of a mistake could be seen with as little simulation time as possible.

More plausible mistake probabilities can be incorporated in this scenario in two ways. The first is to input this value into the scenario using an input parameter in the scenario file. For example, the user could run a SWAP case with the probability of a mistake set to x (if he had good reason to believe that a correct number) and a non-SWAP case with the probability set to $y(>x)$. If x and y are similar in value, many trials at each simulation envelope would be required to produce a significant difference. Depending on the speed and memory capabilities of the computer used, the number of trials may be prohibitive. The second way to use these probabilities is to interpolate between the correct action case and the incorrect action case. For example, if the analyst generates probability of incursion based on correct actions and a probability of incursion based on incorrect actions, then we could calculate the probability of incursion $P(I)$ for the ATC with SWAP training as

$$P(I)_{SWAP} = xP(I)_{Incorrect\ actions} + (1-x)P(I)_{Correct\ actions}\ . \qquad \text{[Eq. 7-1]}$$

Similarly, the probability of incursion $P(I)$ for the ATC that lacks SWAP training would be

$$P(I)_{NO\,SWAP} = yP(I)_{Incorrect\,actions} + (1-y)P(I)_{Correct\,actions} \quad . \qquad \text{[Eq. 7-2]}$$

The user can use these equations to compute the increased safety between the SWAP case and the no-SWAP case with a minimum number of simulation trials.

If x and y are similar, the probabilities of incursion for both cases are likely to be similar as well. This would imply that SWAP's effect on safety is low. That implication is misleading, but we mention it to show the limitations of evaluating a systemwide initiative like SWAP in a tactical simulation. This scenario modeled the consequences of one incorrect tactical decision by one human actor in a tightly focused and narrow set of events, based on one SWAP activity: training. SWAP could improve hundreds of decisions made by many classes of aviation personnel. Those decisions could be tactical or strategic. Further, SWAP includes a focus on better maintenance and management practices that could substantially reduce the hazards of flying. If the likelihood of being in a situation that might lead to an incident is reduced, then the likelihood of an incident also is reduced.

Modeling all of the effects of SWAP with this simulation is infeasible. Tens or hundreds of scenarios similar to this one would be needed to sample the full spectrum of safety improvements that SWAP could provide. At best, it is only possible to model a small number of these improvements as applied to a single scenario.

REPORT DOCUMENTATION PAGE

Form Approved
OPM No.0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources gathering, and maintaining the data needed, and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Information and Regulatory Affairs, Office of Management and Budget, Washington, DC 20503.

1. AGENCY USE ONLY *(Leave Blank)*

2. REPORT DATE

December 2002

3. REPORT TYPE AND DATES COVERED

Final

4. TITLE AND SUBTITLE

Safety Benefit Analysis of NASA Aviation Safety Program Technologies

5. FUNDING NUMBERS

GS-23F-0304K

6. AUTHOR(S)

Shahab Hasan, Robert Hemm, Scott Houser

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Logistics Management Institute
2000 Corporate Ridge
McLean, VA 22102-7805

8. PERFORMING ORGANIZATION REPORT NUMBER

LMI–NS112S2

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Mary S. Reveley
NASA Glenn Research Center
Aerospace Engineer - Aviation Safety Program
MS60-7
Cleveland, OH 44135

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

To be distributed only with permission from task sponsor (Mary Reveley, NASA Glenn Research Center)

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

NASA's Aviation Safety Program (AvSP) seeks to reduce hull loss and fatal accidents. In this task, we developed a method to perform safety benefit analyses of the AvSP technologies. The method comprises two principal analytic components: a reliability model and a simulation model. In the reliability model, we break down the technology into basic components, such as hardware, software, and human agents; define how those components interact; and then determine the failure modes and rates of the components. The results determine how different technologies and systems will perform in normal, degraded, and failed modes of operation. In the simulation, we model an operational scenario and use Monte Carlo methods and specific failures (based on the reliability analysis) to investigate the performance of the technology and identify areas warranting further exploration. The report presents details of the method and results for the Synthetic Vision and System Wide Accident Prevention projects of the AvSP.

14. SUBJECT TERMS

aviation safety, NASA, simulation, reliability, Synthetic Vision, System Wide Accident Prevention, Monte Carlo, Markov

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL