Conclusions

The Minimum Hardware Modification (MHM) hybrid radio-inertial couplers achieved substantially lower turbulence and measurement noise-induced rms path errors than the conventional Autoland radio-coupler design. In turbulence of 10 fps (1 sigma), the MHM path errors were less than 50 ft rms compared to 100 ft rms for the conventional design. The reference MHM designs also displayed superior transient response characteristics. An MHM design utilizing inertial position and inertial velocity as feedback variables appears to provide the best combination of transient response characteristics and stochastic performance.

The value of the optimized solutions is seen in providing a performance bound on a system, against which the performance of a realizable configuration may be measured.

References

[1] Multiple Papers, *Conference Proceedings No. 59 on Hybrid Navigation Systems*, NATO AGARD, AGARD-CP-54-69, Sept. 1969.

[2] MacKinnon, D., "Some Applications of Mathematical Optimization Theory to Automatic Landing Systems," R-651, Nov. 1969, MIT Instrumentation Lab., Cambridge, Mass.

[3] Cherry, G., Desai, M., DeWolf, B., Keene, D., MacKinnon, D., and Madden, P., "Space Shuttle Trajectory Management, Guidance, and Control during Approach and Landing," R-662, May 1970, MIT Charles Stark Draper Lab., Cambridge, Mass.

[4] MacKinnon, D. and Madden, P., "Path Accuracy Limitations of Inertially-Based Flight Control Systems in a Turbulent Environment, *Automatica*, Vol. 8, No. 1, Jan. 1972, pp. 23–34.

[5] MacKinnon, D. and Madden, P., "Performance Limits of an Aircraft Inertial-Based Stochastic Lateral Guidance System," *Journal of Aircraft*, Vol. 9, No. 3, March 1972, pp. 217–222.

AUGUST 1972 J. AIRCRAFT VOL. 9, NO. 8,

Automated Aircraft Scheduling Methods in the Near Terminal Area

LEONARD TOBIAS*
NASA Ames Research Center, Moffett Field, Calif.

A general scheduling algorithm for aircraft from terminal area entry to touchdown is developed. The method has the following novel features: 1) Many speed classes of aircraft are considered and speed variations within classes and along portions of the flight path are permitted. 2) Multiple paths are considered which may merge or diverge—the analysis is not restricted to a single runway nor to departures only. 3) Landings are scheduled along conflict free flight paths in minimum time. The algorithm is currently being incorporated in a fast-time simulation of a STOL air traffic system.

Introduction

THIS paper considers the problem of automated scheduling of aircraft in the near terminal area. The need for such scheduling techniques is widely recognized[1]; this is especially true, as Roberts[2] points out, in a commercial STOL environment where we can expect higher traffic density, tighter scheduling and all-weather operations. Previous work has been done in this area for simple situations such as one runway or all aircraft having the same speed; Athans and Porter[3] have presented an optimal control solution for the problem of landing a merging string of vehicles while maintaining adequate separation between aircraft. A computer aided metering and spacing approach to be used with ARTS III can be found in Ref. 4. This paper presents a more general scheduling algorithm in that many speed classes and multiple runways and approach routes are considered. For each aircraft entering the terminal area, the time to landing is minimized; note that this does not obviate the need for priority rules if some over-all objective such as minimum average waiting time/passenger or maximum landing rate is desired. To illustrate this, the effect of mixing speed classes is briefly discussed in the last section.

The terminal area is modeled as a set of nodes of three types: source nodes which represent entry points into the near terminal area; sink nodes which represent the outer markers, and intermediate nodes which represent other key points in the terminal area such as path intersection points and key holding and reporting points. The phrase "entry point" is used here in a more general sense. An entry point or source node can represent not just one point, but a single feeder fix at which many altitude levels are possible. It is assumed, to simplify the discussion, that these alternate paths will merge in the vicinity of the feeder fix and that from the merge point to each outer marker there is a unique nominal air route structure specified. It is further assumed that there is a minimum separation distance d_M that must exist between aircraft and that the en route center has spaced all incoming aircraft by at least this distance. For example, in Fig. 1, the path from entry point S1 to the outer marker O1 is via the intermediate nodes I7, I8, I2, and I3. As seen in Fig. 2, which depicts the vertical flight profile of the nominal air route, I7 is the merge point of the altitude levels. I8 is the point where paths diverge towards each of the runways, I2 is a reporting point prior to the turn onto the base leg. It might be the point at which holding or a delay maneuver is called for

Received January 13, 1972, presented as Paper 72-120 at the AIAA 10th Aerospace Sciences Meeting, San Diego, Calif., January 17–19, 1972; revision received May 1, 1972. The author would like to thank H. Erzberger for many fruitful discussions during the course of this work.

Index category: Air Navigation, Communication, and Traffic Control Systems.

* Research Scientist.

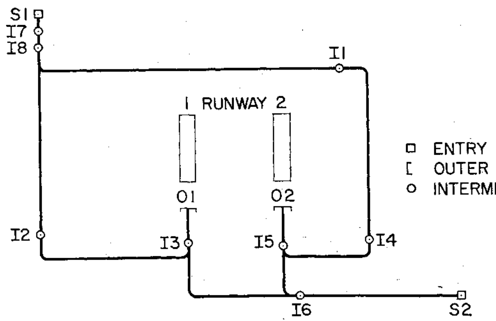

Fig. 1 Terminal area—horizontal view.

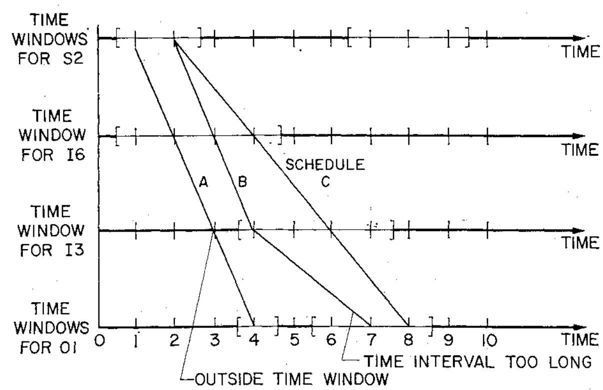

Fig. 3 Finding a feasible schedule.

because aircraft from the other feeder fix are off schedule. I3 represents the merge point of paths from the two feeders. The route structure shown in Fig. 1 is unconventional, since it permits alternate landing approaches from each of two feeder fixes. This is shown to demonstrate the generality of the formulation.

Next we shall define a feasible schedule for an aircraft which has just arrived at a feeder fix. Associated with each node is a set of conflict free time windows. These represent the time slots during which aircraft can be scheduled conflict free at a node. Scheduling an aircraft from entry point to runway will mean specifying for each node in the path a conflict free time at which the aircraft is to pass through that node. The problem of determining the details of the flight path along the route between adjacent nodes will not be considered here; this has been thoroughly studied by Erzberger and Lee.[5,6] However, it is assumed that, using methods such as those developed by Erzberger and Lee, it is possible to specify for each pair of adjacent nodes on a path a pair of numbers, c and d, where c represents the minimum time interval between nodes and d is the maximum time interval to travel between the nodes. These numbers are determined by the distance along the nominal three dimensional flight path between the nodes and possibly some time for path stretching. If holding is necessary for scheduling purposes, it is assumed that it will be done at the feeder fixes. Note that this assumption is made for ease of presentation only. If holding is desired at an intermediate point, two schedules can be computed; the first is from the feeder fix to the intermediate point (treated as a sink), the second from the intermediate point (treated as a source) to the runway. Thus a feasible schedule $\{t_1, \ldots, t_N\}$ for an aircraft along path $\{\alpha_1, \ldots, \alpha_N\}$ is one which has the following properties: 1) the scheduled time t_i for node α_i falls within some conflict free time window of α_i. 2) the time interval between nodes α_i and α_{i+1}, denoted c_1, falls within the allowable range $[c_i, d_i]$.

Before deriving the general scheduling algorithm, the selection of a feasible schedule is illustrated with a simple example.

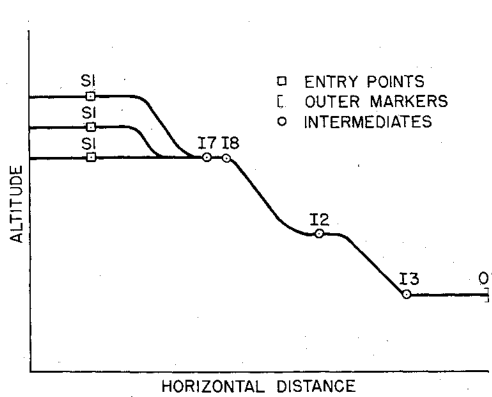

Fig. 2 Vertical flight profile path from S1 to O1.

Consider the path {S2, I6, I3, O1} of Fig. 1, where the conflict free time windows for each node are indicated in brackets in Fig. 3. Consider three possible schedules—A: {1,2,3,4}, B:{2,3,4,7} and C: {2,4,6,8}. Each schedule in Fig. 3 is indicated by a schedule line connecting the time points. The slope of the schedule line between time points is proportional to the time interval between the points. A steep slope corresponds to a short time interval. Suppose, for illustrative purposes, that for each pair of adjacent nodes, the range of permissible time intervals between nodes is $[\frac{1}{2}, 2]$. For this case, and with the time windows given in Fig. 3, only schedule C is feasible. Schedule A is infeasible because $t_3 = 3$, the time to pass through node I3, is outside the time window. Schedule B is infeasible because the time interval between I3 and O1 is 3, which is outside the allowable range $[\frac{1}{2}, 2]$. In terms of Fig. 3, scheduling can be thought of as fitting aircraft into schedule lines which pass through allowable time windows with acceptable slopes between the windows.

Scheduling Algorithm

The general scheduling problem will now be formulated. Consider an air terminal network represented by N_e source, N_b intermediate and N_o sink nodes. Let each node be assigned a unique index number. We assume that there is at most one path between a source and a sink. This path is defined by specifying the source, sink and intermediate nodes. Then we can define the path P_{lm} by a set of numbers as follows:

$$P_{lm} = \{p_1, \ldots, p_N\}$$

where $p_1 = l$ and $p_N = m$ and $p_2, \ldots, p_{N-1}$ are the intermediate nodes. Associated with each node are a set of conflict free time windows; these represent times at which aircraft can be scheduled. For each node i, the set of time windows is of the form

$$W_i = \{[a_1{}^i, b_1{}^i] \cup [a_2{}^i, b_2{}^i] \cup \cdots \cup [a_{k_i}{}^i{}_{-1}, b_{k_i}{}^i{}_{-1}] \cup [a_{k_i}{}^i, \infty)\} \quad (1)$$

where

$$0 \leq a_1{}^i < b_1{}^i < a_2{}^i < b_2{}^i < \cdots < a_{k_i}{}^i{}_{-1} < b_{k_i}{}^i{}_{-1} < a_{k_i}{}^i < \infty$$

k_i = number of windows for node i. By taking the union of these windows, we obtain the total set of conflict free times available at a node i. This is as shown in Eq. (1) where $\cup$ denotes set union. Aircraft can be scheduled at any time $t \in W_i$. Note that the last window guarantees that, for all $t \geq a_{k_i}{}^i$, aircraft can be scheduled through node i.

Let N_t be the number of types of aircraft. "Types" refer to aircraft with similar flight characteristics. Commercial jets would be considered one type aircraft, STOL would constitute another. For each type aircraft and for each adjacent pair of nodes on a nominal flight path, there is a maximum and minimum time interval permitted; if $c_1(i,j,m)$ is the time interval between nodes i and j for a type m aircraft, then

$$c(i,j,m) \leq c_1(i,j,m) \leq d(i,j,m) \tag{2}$$

The quantities c and d depend on the permissible speed range and the amount of path stretching between nodes i and j.

General Scheduling Problem (P1)

An aircraft of type k arrives at source s at time t_a. Schedule the aircraft from the source to any available sink (runway) at the destination airport so that the time to landing is minimized, subject to the constraints on available time windows given by Eq. (1) and the constraints on allowable time intervals between the nodes given by Eq. (2).

This is the key problem to be solved in the scheduling algorithm; it will now be considered in some detail. Following that, the problem of updating the time windows after a new aircraft has been scheduled will be considered; finally, the scheduling algorithm will be outlined.

A simpler problem (P2) of scheduling an aircraft through only two nodes will first be considered; it will then be generalized to solve (P1). Let t_i = time scheduled through node i, $i = 1,2$.

Two Node Problem

(P2): Minimize t_2 subject to the constraints

$$t_2 - t_1 = c_1 \tag{3}$$

$$c \leq c_1 \leq d \tag{4}$$

and each t_i falls within one of the windows of W_i. To solve (P2), first consider a single time window of node 2, denoted by $[g,h]$. Let us find those times at node 1 which can reach the window $[g,h]$ with a value of c_1 satisfying Eq. (4); let us call those times the *reflection* of the window $[g,h]$. It is easily seen that the reflection of $[g,h]$ is given by $[g-d, h-c]$. If any solution through $[g,h]$ is feasible, then its reflection must overlap with one of the time windows* of W_1; that is,

$$[g-d, h-c] \cap W_1 \neq \phi \tag{5}$$

This overlap condition is illustrated in Fig. 4.

Denote the overlap interval by $[p_1,p_2]$. Each $t \in [p_1,p_2]$ has the property that there is some permissible value of c_1 such that $t + c_1$ lies in the window $[g,h]$. The set of permissible c_1 for some $t \in [p_1,p_2]$ is denoted $[e(t), f(t)]$ and has been derived in the appendix. It is also shown in the appendix that the earliest time that can be reached in $[g,h]$ is given by

$$t_2 = p_1 + e(p_1) \tag{6}$$

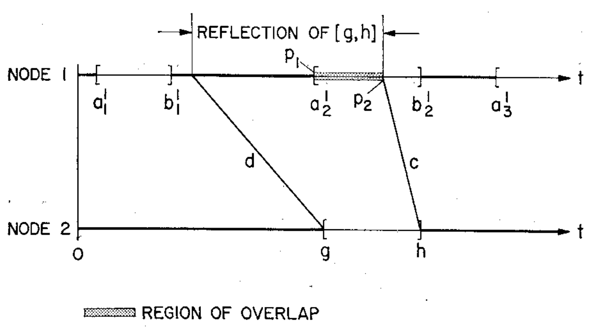

Fig. 4 Reflection of time windows.

*It may actually overlap with more than one time window; in this case, each of the overlap portions would be examined consecutively starting with the earliest time window.

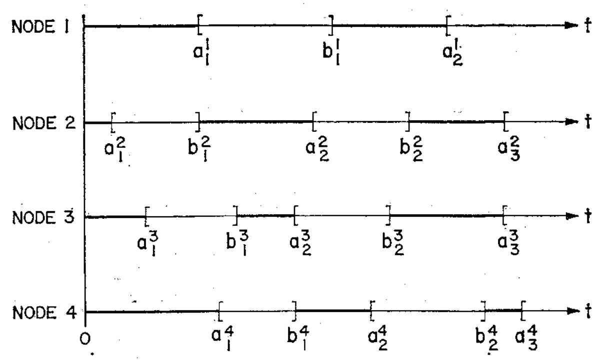

Fig. 5a Time windows for four node problem.

Thus to solve (P2), the following iterative procedure is proposed. Initialization: Start with the earliest window of W_2. Backwards Iteration: Determine if there is any overlap of the reflection of the window of W_2 with any window of W_1 [condition (5)]. If no overlap, try the next window of W_2. Forwards Iteration: Starting with the earliest point of overlap, determine the earliest time that can be reached [condition (6)]. Note that there will always be a solution found via this iteration, since the last intervals considered $[a_{k1}, \infty)$ and $[a_{k2} - d, \infty)$ do overlap.

The problem of determining a schedule through a path with N nodes is also solved via the iteration given above. A sample solution is given in Fig. 5. Figure 5a shows the time windows at each of the four nodes. In Fig. 5b the first window of node 4 is investigated. Starting with $[a_1{}^4,b_1{}^4]$, the reflection of $[a_1{}^4,b_1{}^4]$ is found and the overlap $[p_1{}^3,p_2{}^3]$ is found. Then the reflection of $[p_1{}^3,p_2{}^3]$ is found and its overlap $[p_1{}^2,p_2{}^2]$ with windows of node 2 determined. This process is repeated for node 1 but it is seen that there is no overlap of the reflection of $[p_1{}^2,p_2{}^2]$ and the windows of node 1. Hence, no feasible solution exists through the window $[a_1{}^4,b_1{}^4]$ of node 4. Figure 5c then considers the next window $[a_2{}^4,b_2{}^4]$ and this time feasible solutions exist.

Having determined the time interval in node 1 that leads to a conflict free window in node 4, we can perform the last step of the scheduling algorithm, namely the forward iteration, in which the actual times the aircraft must pass through the four nodes are determined. To guarantee landing in minimum time, we choose the time at node 1 as the earliest time in the overlap interval, denoted by t_1. Times t_2, t_3, t_4 are found by successive application of the equation

$$t_{i+1} = t_i + e(t_i), \; i = 2,3,4$$

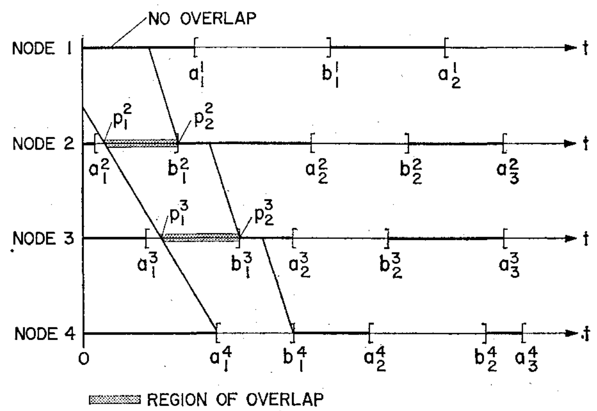

Fig. 5b Backwards iteration—no solution.

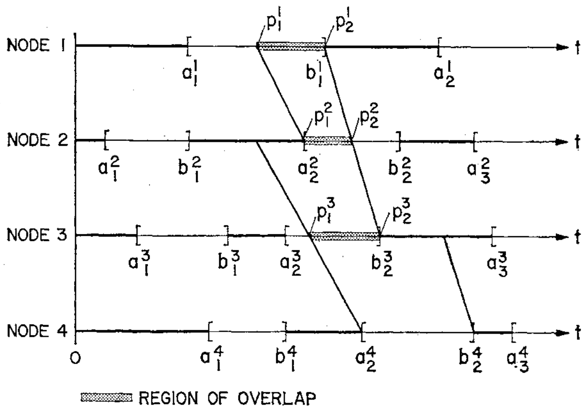

Fig. 5c Backwards iteration—feasible solution.

In this case, the minimum time $t_{\min} = t_4$ (Fig. 5d).

It should be noted that this method determines the minimum time as well as a feasible schedule to attain that minimum; however, there may be many solutions that attain that minimum. To exhibit all possible minimum time solutions, start at the sink node $t_{\min}$ and proceed backwards to the source via the steepest permissible slopes.

Thus the general scheduling problem is solved as follows: Find the minimum time solution $t^i_{\min}$ to each sink i, $i = 1, \ldots, N_0$, via the method given above, then $t_{\min} = \min_{i=1,N_0} t^i_{\min}$, and suppose sink K is the one that can be reached at $t_{\min}$, then choose the schedule desired to reach node k at $t_{\min}$.

Restriction of Time Windows

Another major part of the scheduling problem is the updating method. Once an aircraft has been scheduled, it is necessary to update (restrict) the time windows of the nodes through which it is scheduled to pass in order that the next arriving aircraft can be scheduled conflict free. This must be done so that conflict with future incoming aircraft is avoided not only at the nodes but also at points between the nodes. In other words, a minimum separation distance must always be maintained between aircraft. Suppose an aircraft of speed class k has been scheduled along path $\{\alpha_1, \ldots, \alpha_N\}$ at times $\{t_1, \ldots, t_N\}$. Then for node α_i, no aircraft of class j can be scheduled at any time t in the interval $t_j - l(\alpha_i,k,j) < t < t_i + u(\alpha_i,k,j)$ where $l(\alpha_i,k,j)$ and $u(\alpha_i,k,j)$ are the restrictions imposed on scheduling a type j aircraft at node α_i after a type k aircraft has aleady been scheduled at α_i at t_i. These numbers, $l(\alpha_i,k,j)$ and $u(\alpha_i,k,j)$, depend on the relative speeds of the aircraft and the distance between adjacent nodes; they can be chosen to produce a conflict free schedule. Methods of determining l and u will now be discussed.

Consider two nodes separated by a distance d and suppose that an aircraft with velocity v_1 has been scheduled through

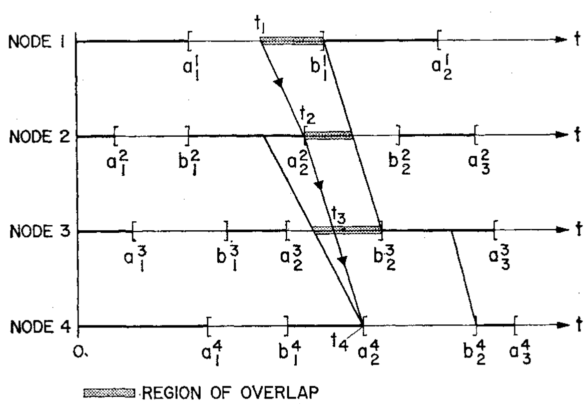

Fig. 5d Forwards iteration.

nodes 1 and 2 at times t_1 and t_2, respectively. We seek to restrict passage through node 1 for the interval $(t_1 - l_1, t_1 + u_1)$ and through node 2 for the interval $(t_2 - l_2, t_2 + u_2)$ so that the minimum separation distance d_M is not violated when other aircraft are scheduled through those nodes. Obviously, the minimum separation distance must be satisfied at the nodes; so that l_1, u_1, l_2, u_2 must all be greater than or equal to d_M/v_1. In fact, if a second aircraft with velocity v_1 is to be scheduled through these nodes, then

$$l_1 = u_1 = l_2 = u_2 = d_M/v_1$$

Consider an aircraft, velocity $v_f > v_1$, which is to be scheduled. The following must be prevented: Aircraft with velocity v_f is scheduled at node 1 after the aircraft with velocity v_1, but at node 2 before the aircraft with velocity v_1. Overtake is prevented if

$$v_f < d/[(t_2 - l_2) - (t_1 + u_1)]$$

Using the fact that $t_2 - t_1 = d/v_1$, we obtain

$$l_2 + u_1 > (v_f - v_1/v_f v_1)d$$

as an additional constraint that must be satisfied.

Consider an aircraft, velocity $v_s < v_1$, which is to be scheduled. The following must be prevented: Aircraft with velocity v_s is scheduled at node 1 before the aircraft with velocity v_1, but at node 2 after the aircraft with velocity v_1. This is prevented if

$$v_s > d/[(t_2 + u_2) - (t_1 - l_1)]$$

and thus

$$u_2 + l_1 > (v_1 - v_s/v_s v_1)\, d$$

Random effects such as wind or instrumentation errors may lead to more conservative values of l and u. Suppose that two aircraft both with velocity v_1 have been scheduled consecutively through node 2 at times $t_2{}^1$ and $t_2{}^2$, but the actual passage times through the node are given by

$$y_1 = t_2{}^1 + \varepsilon_1, \qquad y_2 = t_2{}^2 + \varepsilon_2$$

where ε_1 and ε_2 are independent identically distributed zero mean random variables. Additional time denoted by Δ must be allotted if the minimum separation distance requirement is to be met with a specified probability p. This additional time is determined from

$$\text{Prob}\,[(t_2{}^2 + \varepsilon_2) - (t_2{}^1 + \varepsilon_1) < (d_M/v_1)] \leq 1 - p$$

Since

$$t_2{}^2 - t_2{}^1 = (d_M/v_1) + \Delta$$

$$\text{Prob}\,[\varepsilon_2 - \varepsilon_1 < -\Delta] \leq 1 - p$$

For given distributions of ε_1 and ε_2 and given value of p, the additional time Δ can be obtained.

Additional factors will enter into the final choice of the variables. For simpler computations in the scheduling part of the algorithm, it is convenient to form these restrictions not for each possible velocity but for a range of velocities; also velocity changes may be required near nodes. Hence, more conservative constraints will be obtained.

Effect of Mixed Speed Classes on Landing Rate

To illustrate the use of the scheduling procedure in complex terminal areas, landing rates were investigated. A worst case study was performed to examine the effect of different speed classes of aircraft using the same airways. The path structure of Fig. 1 was used and two speed classes were considered. For the slower, the speed range was 110–130 knots, whereas for the faster it was 170–190 knots. The separation distance was 2 miles. One hundred aircraft were scheduled to arrive in the terminal area in rapid succession and so holding would be necessary. It was assumed that aircraft would be stacked at the entry points until they could be scheduled through the terminal network. Scheduling was on a first come-first served basis. The worst case of arrivals was considered—with as much intermixing of speed classes as possible. The results are given in Table 1. As expected the worst landing rate is achieved for a 50–50 mixture of speed classes; however, even if only 20% of the aircraft are of a different class landing rate is about $\frac{2}{3}$ of what it is when only one speed class is used in the terminal area. This suggests either having separate paths until as close to the outer marker as feasible or possibly some priority landing scheme. This is a subject of further investigation.

Conclusion

A conflict free scheduling procedure has been presented in which landing time is minimized for each aircraft entering the terminal area. The method is suitable for studies of complex terminal areas in which there are multiple runways and approach routes and in which many speed classes of aircraft are present. The scheduling method is currently being incorporated in a simulation of a STOL traffic system. Two related areas need further consideration. First, when various speed classes of aircraft are present, priority rules need to be developed to minimize overall objective functions such as maximization of landing rate or minimization of average waiting time of all aircraft entering the terminal area. Second, a hierarchy of rules needs to be developed to handle the real time rescheduling of aircraft for collision avoidance and emergency procedures.

Appendix

Determine the Set $[e(t), f(t)]$

Let $[g,h]$ be a time window and $[g-d, h-c]$ its reflection. Then for each $t \in [g-d, h-c]$, there is a $c \in [c, d]$ such that $t \in [g,h]$. Let $[e(t), f(t)]$ be the set of allowable values of c for some $t \in [g-d, h-c]$.

Lemma 1: Suppose that $h - g \geq d - c$.

(i) If $g-d \leq t \leq g-c$, then $e(t) = g - t$, $f(t) = d$

(ii) If $g-c \leq t \leq h-d$, then $e(t) = c$, $f(t) = d$

(iii) If $h-d \leq t \leq h-c$, then $e(t) = c$, $f(t) = h - t$

This is depicted in Fig. 6 where $r(t) = f(t) - e(t)$ is plotted against time. Note that $e(g-d) = f(g-d) = d$ and that $e(h-c) = f(h-c) = c$.

Proof: (i) To prove (i) we show that for every $c \in [e(t), f(t)]$, $t + c$ lies in the window $[g,h]$. For the minimum allowable c, $e(t)$, we have

$$t + e(t) = t + (g - t) = g \in [g,h]$$

Table 1 Mixing aircraft speed classes

Percentage of faster class of aircraft	normalized landing rate
100	1.000
80	0.638
50	0.574
20	0.628
0	0.980

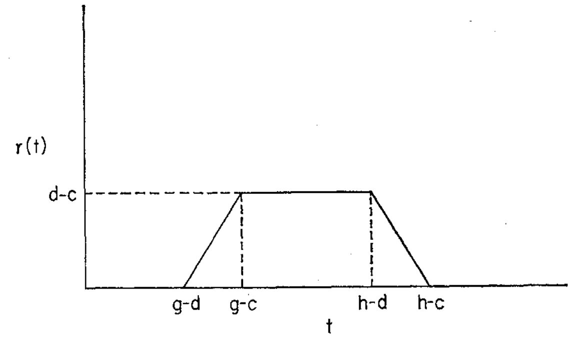

Fig. 6 Allowable intervals.

Consider next $t + f(t)$

$$t + f(t) = t + d \geq (g - d) + d = g \in [g,h]$$

Also $t + f(t) = t + d \leq (g - c) + d \leq h \in [g,h]$. Note also that if $t < g - d$, then $t + c < g - (d - c) < g$. A similar analysis demonstrates (ii) and (iii). For $(h - g) < d - c$, the following values are substituted for $e(t)$ and $f(t)$:

(i) If $g-d \leq t \leq h-d$, then $e(t) = g - t$, $f(t) = d$

(ii) If $h-d \leq t \leq g-c$, then $e(t) = g - t$, $f(t) = h - t$

(iii) If $g-c \leq t \leq h-c$, then $e(t) = c$, $f(t) = h - t$

Lemma 2: Consider $t_a, t_b \in [p_1, p_2]$, the overlap interval. If $t_a > t_b$, then $t_a + e(t_a) \geq t_b + e(t_a)$.

Proof: If both t_a, t_b satisfy (i) of lemma 1, then

$$t_a + e(t_a) = t_b + e(t_b) = g$$

If t_a satisfies (ii) or (iii) of lemma 1, then

$$t_b + e(t_b) = g, \text{ while } t_a + e(t_a) = t_a + c \geq g - c + c = g$$

If t_a and t_b satisfy (ii) or (iii),

$$[t_a + e(t_a)] - [t_b + e(t_b)] = t_a + c - t_b - c = t_a - t_b \geq 0$$

References

[1] "Report of the Department of Transportation Air Traffic Control Advisory Committee," Dept. of Transportation, Washington, D.C., Dec. 1969.

[2] Roberts, L., "The Promise of Aeronautics," *Astronautics and Aeronautics*, May 1971, Vol. 9, No. 5, p. 24.

[3] Athans, M. and Porter, L. W., "An Approach to Semiautomated Scheduling and Holding Strategies for Air Traffic Control," *Proceedings 1971 Joint Automatic Control Conference*, Automatic Control Council, Washington Univ., St. Louis, Mo., Aug. 1971, p. 536.

[4] Computer Systems Engineering, "Computer-Aided Metering and Spacing with ARTS III," FAA Rept. RD-70-82, Systems Research and Development Service, Washington, D.C., Dec. 1970.

[5] Erzberger, H. and Lee, H. Q., "Optimum Horizontal Guidance Techniques for Aircraft," *Journal of Aircraft*, Vol. 8, No. 2, Feb. 1971, pp. 95–101.

[6] Erzberger, H. and Lee, H. Q., "Terminal Area Guidance Algorithm for Automated Air Traffic Control," TN D 6773, Feb. 1972, NASA.