RUNWAY OPTIMIZATION FOR GENERALIZED CONTINUOUS PAYOFF FUNCTIONS

Dr. William S. Beebee

GE Aerospace, Valley Forge, Pennsylvania

Submitted to The

33rd Annual Air Traffic Control Association Fall Conference

October 31 - November 3, 1988

Hyatt Regency Crystal City, Arlington, VA

RUNWAY OPTIMIZATION FOR GENERALIZED CONTINUOUS PAYOFF FUNCTIONS

Dr. William S. Beebee
GE Aerospace, Valley Forge, Pennsylvania

ABSTRACT

To an air traffic controller, allowing a particular takeoff or landing operation to occur produces a relative performance payoff which can vary with the number of passengers on the aircraft, the relative safety of the operation, the amount of delay already imposed on the aircraft, the amount of fuel remaining in an aircraft trying to land, etc. Within a given time span, the controller can achieve better traffic handling performance and a more balanced treatment of these factors if he or she maximizes the total payoff of all takeoff/landing operations occurring, subject to safety-related constraints. The optimized variables are the times at which the individual operations are allowed to occur. Some of the constraints which must be obeyed are that each operation occupies a given time window on any specified runway and that no two operations on the same runway are allowed to occur so closely in time as to be concurrent. Also, the waiting time for any aircraft running low on fuel is not allowed to exceed a safety limit.

This paper presents a technique for choosing optimal times for the operations, such that all constraints are obeyed and the total performance payoff, taken over all aircraft involved, is maximized. Three features of particular utility in the optimum seeking technique are: any continuous functions can be used to represent the individual operation payoffs, the search progress is easy to visualize, and trade studies are simplified since sensitivity results in the neighborhood of the final solution are immediately available without recomputing the optimum. The latter two features enhance the user understanding of why the solution is correct. This, in turn, increases confidence in the optimum obtained.

INTRODUCTION

The intent of this paper is two-fold:

1. To present a useful and flexible formulation of an important air traffic control problem seen in the terminal area
2. To give a computational approach for solving that problem.

The issue addressed is the allocation of runway space and time to competing flights attempting to land and take off to and from busy terminals. The study approach illustrated here offers flexible formulation of the allocation payoff function so that many different allocation policies can be assessed and compared.

As seen in Ref. 1, the air transit system has made progress in reducing the number of aircraft kept waiting to land in terminal control areas (TCAs). This has been accomplished, in part, by performing country-wide resource allocations on the flow to the most busy airports, so that aircraft destined for busy TCAs are delayed at their departure points. This eases the airborne loitering burden at the overloaded terminals.

Nevertheless, improvements in national traffic flow depend in part on the success of traffic flow into and out of the TCAs. If the TCA flow is improved at the beginning and end of flights, the end-to-end flow is also improved. Also, any excess capacity generated by better TCA scheduling may allow emergency delays in the national system to be temporarily removed at the local level.

The particular formulation of the TCA problem assumed here and the solution approach taken offer flexibility in performing studies of widely differing scheduling philosophies. There are other rationale for granting TCA service to a waiting a/c which may be more desirable than the presently-used "first come, first served" policy. For instance, it may be important to grant higher priority to large a/c with many passenger which has already seen severe delays than to a small commuter plane which is on time. This approach may be driven by the motivation to reduce the average delay per passenger.

The optimization approach used below can deal with any continuous payoff function. Thus, in deciding the policy to be used for scheduling, there is a wide latitude of choice.
The above issue and many others, fundamental to improving the success of the TCA scheduling effort, can be studied and the results compared.

PROBLEM STATEMENT

Assume that there are n aircraft trying to use m runways. If the event of the ith runway servicing the jth a/c is allowed to occur, as opposed to using a different runway, there is a payoff due to accomplishing that operation. Let the payoff for this ijth event be denoted Vij. The payoff can be a function of many variables. However, other than making the decision as to whether or not a given event is to occur at all, the decision parameter of interest is time. Thus, the payoff function is denoted Vij(tij), meaning the ij event payoff, dependent on the tij assigned time of occurrence.

Every runway operation occupies the runway for some time window, a fact causing serious constraints on aircraft operations. This window varies, generally, with the particular a/c and runway involved and can be very complex to calculate. For a/c landings the window includes data on time required to exit the TCA and land. Takeoff and ingress into the TCA influences this window size for departure operations. References 2-4 discuss challenges and opportunities with these operations and in one case, (Reference 2), provide optimal paths for doing so. Reference 5 discusses an even more complex issue: the "cross-coupled" constraints on close, adjacent runways due to turbulence.

For the purposes of this study, the window duration, Tij, is considered to be constant over schedule times. The decision parameter for a given event, tij, is defined for the center of the Tij window. This convention is adopted to facilitate the mathematics of safety constraints, discussed below.

To help prevent accidents, the windows of two a/c using the same runway are not allowed to overlap. This is expressed, for the jth and kth a/c using the ith runway, by the constraint:

$$| tij - tik | > (Tij + Tik)/2 \qquad (1)$$

The absolute value brackets account for the fact that we do not know, before performing the optimization, which of these events will occur before the other. Defining the event time, tij, at the center of the associated Tij window allows a more simple expression of the constraint.

The above constraint concerns simultaneous use of the same runway. Due to turbulence generated near the ground by a large a/c using a runway, a nearby runway may become temporarily dangerous to use, particularly for a smaller a/c, as noted above. It is possible to model this type of constraint. If the ith runway, contemplated for use by the jth (large) a/c, is near the qth runway, with possible use by the rth (small) a/c, the constraint follows.

$$| tij - tqr | > (Tij + Tqr)/2 \qquad (2)$$

The information needed to choose the T quantities and to decide on which constraints, of the form of equation (2), to write can be considerable. However, despite the fact that this formulation can encompass much information about the problem, the formulation is still simple to work with.

In addition to these constraints, which deal with interactions between a/c, there are constraints on the earliest and latest time that an a/c can be serviced. The early time limit relates to the set of Tij windows which the jth a/c faces over all of the runways. Because tij is defined for the center of the Tij window and this window describes the time consumed in performing the contemplated landing or takeoff operation, the window cannot begin before the present instant. This means that

$$tij > Tij/2 \qquad (3)$$

in all accepted events. The late time limit does not relate to the operation window but, rather, expresses the maximum time allowed for the a/c to remain aloft. This is ultimately set by the amount of fuel remaining onboard. As noted above, the maximum time is denoted by Wj for the jth a/c, and produces the constraint:

$$tij < Wj \qquad (4)$$

Sometimes, the value of Wj is so large, relative to the times of strong payoff available for that a/c, that the value has virtually no effect on the optimization solution. In other instants, an example of which is seen below, the size of Wj can severely restrict the allowable size of the tij space available for searching.

The goal of the optimization is to maximize the total payoff of all of the a/c-runway events which are allowed to occur. Thus, the optimization has two aspects which must be considered simultaneously:

1. The event describing which runway serves each a/c must be decided

2. After all such events have been chosen, the time of each must be selected, subject to a relevant subset of the above constraints, so as to maximize the total

payoff over all allowed events.

As discussed below, generation of optimal times may change the set of events selected, requiring the choice of a new set of allowed events. Also, time constraints, ignored for events which have not been selected, may become relevant if those events are "activated".

SOLUTION APPROACH

To be sure of obeying all constraints, the first starting point is chosen to lie on the outer W constraint boundary. Thus, the times are chosen to be, collectively, as large as possible.

As seen in the low-order example below, the overall time region is not simply connected. The region is composed of at least two subregions which have no common points. Thus, it is impossible to reach some parts of the region from other parts. This occurs because the constraints in equations (1) and (2) have absolute value signs on the time difference between each pair of events. The absolute value sign reflects the fact that there is generally, no reason to believe that one a/c in the constraint will land ahead of the other one. Thus, both possibilities must be checked for each constraint.

A gradient search is used to find the optimal tij set for a region, once the allowed events have been selected. (See Reference 6 for discussions of the gradient techniques used.) The implication of the disconnected subregions phenomenon on the gradient search is that, in general, a new search path must be started and carried through to its conclusion for each subregion. (Thus, once we have begun a search with one of the two a/c landing ahead of the other, it is not possible to reverse this order without restarting the search with the other order assumed.) Once all subregions have been explored, the optima found in all cases are compared and the "best" of all is selected.

There is more discussion of this point, below. The balance of this section is devoted to treatment of the search of a single subregion.

Starting at the initial point, the first activity is to select the events to be considered. This is done by choosing the runway usage event for each a/c which has the highest payoff.

The next action is to compute the gradient over all of the events, whether selected or not. We do not, as yet, need to consider the effect of any constraints since no such surfaces have been intercepted. The gradient defines a linear path forward. The intersections of this path with all constraints for selected events are computed with little effort, since they are solutions of simultaneous linear equations. (Intersections with constraints for events not selected are ignored, at present.)

The first intersection along the path is identified, and a step of length sufficient to reach that point is taken. The times for nonselected events are carried forward using their respective gradient components.

At the constraint surface intersection, a check is made to see if all of the selected events are still dominant for their respective aircraft. If some events have been surpassed by others, we need to know where along the path traveled the changes occurred. The gradient is computed at the end of the step and the projection, or "shadow", of the gradient in the step path is computed. For each pair of events changing places, the corresponding gradient data at the start and end of the step, as well as the start and end points, are used to make cubic interpolation estimates of where the crossover occurred.

Once these estimates are made, the corresponding times are set equal to the estimates and the payoff contributors, reversing roles, are computed. In general, this process of stepping followed by cubic interpolation must be repeated until the points at which crossover occurs are located. If there is more than one crossover point, the one of interest lies closest to the original point of departure for the path. The step is made to that point and the new set of selected events is identified.

The search now proceeds with the new set of selected events. However, a new set of constraints has now been introduced, and one or more of them may be currently violated. If there are more than one, a series of correction steps is made, each of which lands on the appropriate constraint boundary; each, also, achieves the greatest overall payoff improvement consistent with reaching the constraint.

The direction of each correction step is calculated by taking the inner product of the gradient with the direction of most rapid progress to constraint satisfaction. If the inner product is negative or zero, the step is made in the direction perpendicular to the gradient which, itself, so as to minimize the inner product between that perpendicular and the direction of most rapid progress. If the inner product is

positive, the step is in the gradient direction; since the constraints are all linear, the intersection point is easily calculated.

Now, reconsider the earlier situation of first intersecting the constraint surface. If, at the end of the original unconstrained gradient step, the set of selected events has not changed, we check to see if we should be looking for a maximum back along the path we have travelled. Such a condition is checked by computing the payoff gradient at the constraint boundary and using it to form two inner products. The first is the inner product of the payoff gradient with the direction of most rapid improvement in constraint satisfaction. (The latter direction, discussed above, is normal to the constraint surface and pointing into the feasible region.)

The second inner product involves the gradient and the previous search direction. If this inner product is negative, we must search back in the direction we have come from. If this inner product is positive and if the inner product involving the constraint is negative, the gradient points into the constrained region and the next step should lie in the constraint surface. If this inner product is positive, the new gradient points away from the constraint. If the gradient points away from the constraint surface but does not call for reverse direction back on the prior path, we should take an unconstrained step along the new gradient direction.

If searching backwards along the unconstrained approach direction is called for, a similar approach to that for discovering the dominant event crossover point, discussed above, is followed. The beginning and ending points and gradients are used, with the terminal gradient projected into the search path. A cubic polynomial is fit to the data. The maximum is found by solving for that zero root of the quadratic derivative which has a negative second derivative.

A step is taken to the predicted maximum and the above process repeated a few times until satisfactory convergence is reached. At this point we are now at a maximum along the original search direction.

Since we have not, as yet, encountered any constraints, we can search the space with a more efficient routine which is designed for rapid convergence in unconstrained regions. The approach used is the conjugate gradient method. At the present maximum, we calculate a conjugate direction. (See Reference 6.) We step in that direction to the next constraint by solving for the step length as before. Also, as before, we check to see if the set of selected events has changed and if the constraint surface should be entered. The process follows exactly as described above.

If there is no change in the selected event set and we should search back along the path we have come, indications are that the maximum can still be sought with an unconstrained search. Thus, we find the maximum and compute another conjugate direction to search. If at any time we either intersect a constraint or find that the event selection changes, we, temporarily, take no more conjugate gradient steps. As soon as we can take the conjugate steps again, we do.

Return, again, to the situation at the end of the first, unconstrained step. If, where the constraint region is intersected, the optimum search direction takes us into it, we compute the gradient projection in the surface. The gradient projection, $\underline{P}$, is calculated from the unconstrained gradient, $\underline{G}$, and the above-mentioned (unitary) direction of constraint satisfaction, $\underline{C}$.

$$\underline{P} = \underline{G} - (\langle \underline{C} , \underline{G} \rangle / |\underline{G}|) \underline{G} \qquad (5)$$

This formula is based on the Gram-Schmidt orthogonalization procedure, in which that part of $\underline{G}$ parallel to $\underline{C}$ is removed.

A step can be taken in this surface without concern for leaving it "involuntarily", since both step path and surface are linear. The intersection of the step in the surface with the next constraint is calculated algebraically and the step is taken. Note that the new point satisfies both the old and the new constraints. The unconstrained gradient is computed at the new point.

Similar checks and decisions are now made as was done at the point of intersection with the first constraint. First, the set of selected events is examined to see if any other events have higher payoff and should be substituted. As with the unconstrained step, the gradient components for the events not selected carry those events to higher payoff values during the step. After the step is over, we may find that some of the selected events need to be replaced.

If this occurs, we look back in the opposite direction, using repeated cubic fits as before, to find the first point during this step at which the event set

changed. We move to that point and change the event set to include the first new event. In general, there will now be a new constraint set and it may be violated. As before, we need to take correction step(s) to obey the newly-activated constraint(s).

Any correction step is computed without attempting to stay in the old constraint surface, since such an operation may no longer be optimal. Whenever new constraints arise, the first effort is to obey them; then, the decision as to what surface to step in, if any, is made. One of the characteristics of a first-order gradient search is that the recent past is "forgotten" (although the second-order conjugate gradient approach does retain and use "memory" of the previous steps).

If the above-mentioned gradient projection step did not result in a change of the event set, other checks are required. The inner product of the gradient at the end is formed with the search direction taken. If this inner product is negative, the indication is that a maximum has been passed during the step. The same procedure is used, as seen above for the unconstrained step, to locate and step to this point.

If this inner product is positive, then the check is made to see if the unconstrained gradient points into either active constraint. For any constraints that this is true, the gradient projection into all such surfaces, simultaneously, is computed. The next step will be taken in that direction. If the unconstrained gradient points into no constrained regions and does not call for a retreat back the previous path, the indication is that this direction should be followed into the unconstrained region.

The above process is followed, stepping in constrained or unconstrained regions and changing selected event sets, as required. The process self-terminates at a local maximum from which no further optimizing steps are possible. This condition can occur in one of three ways:

1. The search can be locally unconstrained and reach a point at which the norm of the gradient is acceptably close to zero

2. The search can be in one or more constraint surfaces, simultaneously, and can reach a point in these surfaces at which the gradient projection is acceptably close to zero

3. The search can end with all selected event variables constrained and the unconstrained gradient calling for violations of all such constraints.

Note that the above discussion describes a search ending on a local maximum. This is the normal output of a gradient solution process. The question of whether or not this is a global maximum must be addressed. A standard technique is to start at several separated initial points and see if the solution always ends at the same location. However, this "brute force" technique may not be necessary when known payoff function characteristics are utilized and the space to be searched is understood.

Visualization of the search progress and insight into sensitivity relationships about the optimum solution point are important for building understanding of and confidence in the solution obtained. Since the solution approach is based on gradient search techniques, first-order sensitivity relationships are available for all variables at any step during the search process. One simply uses the local unconstrained and constrained gradients and the directions of constraint obedience for any active constraints to form any sensitivity function desired.

Visualizing the search progress is more difficult. Although the following example lies in a space of low enough dimension to be plotted, typical problems of interest are too complex for such methods. Fortunately, a technique has been devised which enables visualization regardless of the space dimension.

Consider two adjacent step vectors, $\underline{S}n$ and $\underline{S}n+1$, occurring in the search process. Each represents a search direction. The angle between these two directions, An-n+1, is found from the inner product of the two vectors divided by the product of their norms:

$$An\text{-}n+1 = \arccos(\langle \underline{S}n, \underline{S}n+1 \rangle / (|\underline{S}n| \; |\underline{S}n+1|)) \qquad (6)$$

The search is visualized in a 2-D cartesian frame by plotting the $\underline{S}0$ vector along the positive x axis with the tail at the origin. The point of the vector "lies" at the end point of the first step. From this point, a vector of length $|\underline{S}1|$ is drawn, at an angle of A0-1 from the prior vector. The end point of this vector represents

the location of the end point of the second step.

This process is repeated for the balance of the search. For each new step the new vector is drawn with tail at the point of the previous one and lying at an angle from the previous vector equal to the An-n+1 measured between the present and previous steps. This process is illustrated in the example given below.

Note that this visualization tool does not represent the search uniquely, because orientation in the full-dimension space is lost in the inner product operation. However, it is useful for building intuition about the optimization process.

LOW-ORDER EXAMPLE CASE

In this case three aircraft are attempting to use two runways. Figure 1 illustrates the problem. There are six possible a/c-runway events. The event parameters and payoff functions are identified by i,j indices, in which i represents the runway number (1 or 2) and j represents the a/c number (1, 2 or 3). The individual plots are of event payoff value over the time, in minutes, assigned for the event. For each plot the value of T on the t scale, and inner and outer time constraint boundaries due to T and W, respectively, are drawn.

The starting point for the initial search is the outer time boundaries seen in Figure 1. Throughout this search the set of maximum payoff events remains constant. The 1,1 as well as the 2,2 and 2,3 events are always dominant. This allows us to deal with only those times and to visualize the search space via the split-scale 3-D plot shown in Figure 2. There, the isoclines of constant <u>overall</u> payoff are presented, overlaid on the constraint boundaries. The diagonal forbidden region in the figure shows the effect of the "absolute value" time constraint prohibiting the time windows of a/c 2 and 3 from overlapping when they use runway 2. This type of constraint is seen in equation (1), above.

Two search paths are shown in Figure 3. They correspond to the initial search from the extreme time boundaries and another search covering the second, upper feasible region in the 2,2-2,3 plane. The interpretation of searching the lower feasible region in the figure is that a/c 3 is assumed to use runway 2 before a/c 2. For the other case, searching the region above implies that a/c 2 lands before a/c 3.

In each case the search starts at a circled point and takes an unconstrained

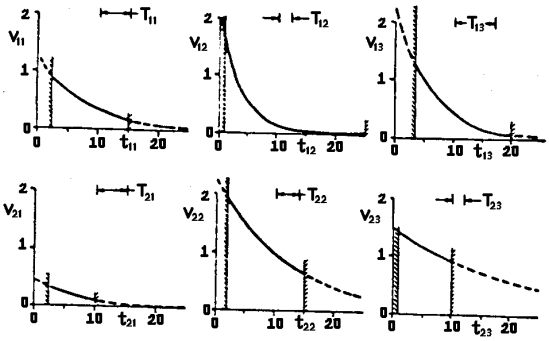

Figure 1: Two Runways For Three A/C

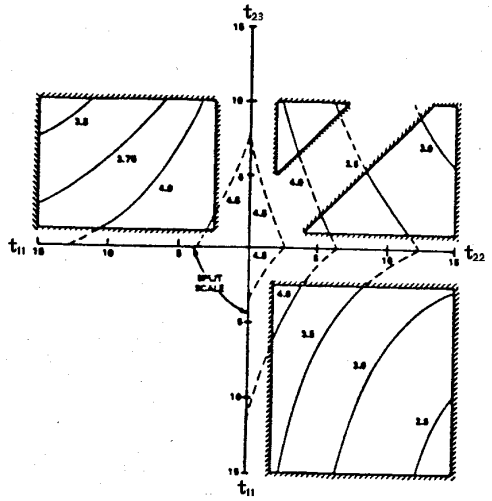

Figure 2: Time Boundaries and Overall Payoff Isoclines

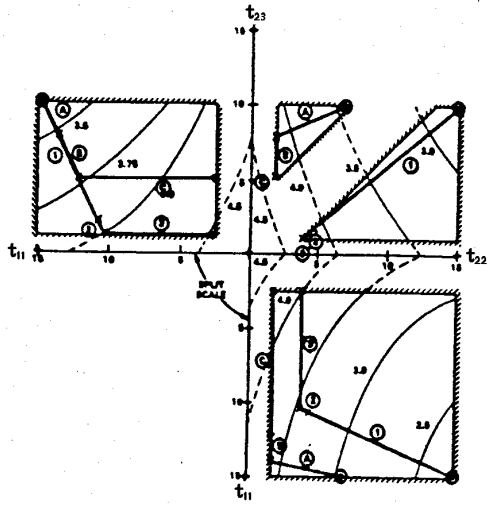

Figure 3: Searching Each Feasible Region

step to the first constraint intersection. The initial search takes step 1 to the "absolute value" constraint discussed above. Step 2 occurs in this constraint boundary until

the lower limit constraint on 2,3 time is reached. This fixes both the 2,3 and 2,2 times as the final step, 3, moves to the solution at the 1,1 time boundary. The total final payoff achieved is 3.92

The second search, implying that a/c 2 lands earlier, takes the first step, A, to the lower limit on 2,2 time. Step B then travels in the 2,2 boundary until the upper absolute value constraint is reached. There, the 2,2 and 2,3 times are fixed as before. The final step, C, goes to the 1,1 inner time boundary before stopping with a total final payoff of 3.98.

It appears to be a coincidence, caused by the particular relationship of event payoff and constraints, that the two payoff values are nearly the same. The isocline values seen on the plots appear to disagree with the payoff values achieved. However, this effect occurs because the isoclines are shown for the boundary planes and do not accurately reflect the isocline behavior in the internal region, away from these planes.

Figure 4 presents the above-described "inner product" visualizations of the two searches. Both plots are on the same scale and the angle between the two first steps is seen. The utility of this representation can be especially appreciated if we imagine it describing a 15-D search instead of the 3-D case, here.

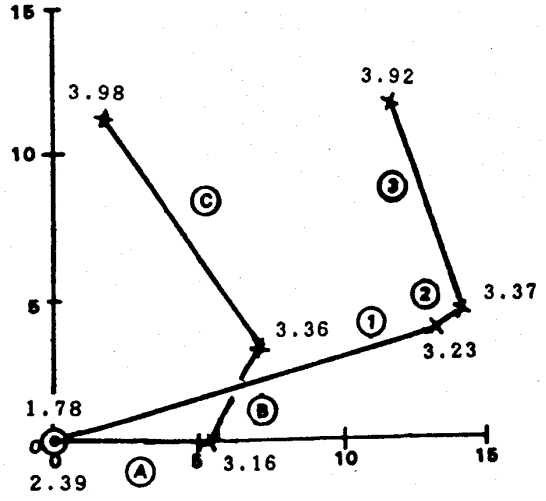

Figure 4: Visualizing Each Search

FUTURE PATHWAYS

The solution techniques presented above need to be developed further and applied to more typical ATC problems of higher dimensions. However, the robust handling of many different payoff functions, high visibility into the solution process and readily-available sensitivity functions for trade studies offered by this approach promise to be invaluable aids for implementing and understanding more advanced TCA allocation philosophies.

Full fruition of such efforts can be greatly assisted when this optimization approach is integrated with and incorporated into a well-designed, "smart" ATC work station.

ACKNOWLEDGMENT

The author is grateful for the assistance of one of the most competent mathematicians at GE in advising this research. Dr. Leslie Arnold provided several key suggestions, without which this work would have been unsuccessful. Especially important is his observation that, if the time constraints are obeyed, the optimum choice of runway to serve a given aircraft is the runway contributing the most payoff.

REFERENCES

1. Fischetti, Mark A. and Perry, Tekla S. "Our Burdened Skies" in IEEE Spectrum, November, 1986.

2. Simpson, Robert W. An Analytical Investigation of Air Traffic Operations in the Terminal Area, (PhD Thesis) Cambridge: MIT Department of Aeronautics and Astronautics, 1964.

3. US Department of Transportation, "Appendix B-7: Airport Design Considerations" in Report of Department of Transportation Air Traffic Control Advisory Committee - Vol. 1, December, 1969.

4. Cherry, George W., et al, Report R-654: A New Approach and Landing System: Help For Our Troubled Areas Cambridge: Charles Stark Draper Laboratory, March, 1970.

5. Istefanopulos, Yorgo, Analytic Study of Multiple Runway Operations, (PhD Thesis) Cambridge: MIT Department of Electrical Engineering, September, 1972.

6. Bryson, Arthur E., Jr. and Ho, Yu-Chi Applied Optimal Control Waltham (Mass.): Blaisdell Publishing Co., 1969.

William S. Beebee
Senior Consultant
GE Aerospace - Valley Forge, PA

Dr. Beebee was born in Centralia, Illinois, in 1942. He attended Princeton University and received a BSE (Cum Laude) in Electrical Engineering in 1964. He earned an MS in EE from the University of Pennsylvania in 1969 and an Sc.D. in Instrumentation from the MIT Department of Aeronautics and Astronautics in 1975. At the latter institution he received the MIT Creative Engineering Award for a zero-zero taxiway navigation system concept.

In 1964 he joined the Calspan Corporation and participated in the creation of an inflight simulator for the Air Force. Later, he joined United Technologies for the Apollo Program and, then, researched inertial systems for NASA while at The Analytic Sciences Corporation (TASC).

After receiving his Doctorate, Dr. Beebee did further work on inertial systems at the Charles Stark Draper Laboratory and the Northrop Precision Products Department. Prior to joining GE in 1984, he was at Bendix Guidance Systems Division, heading a group deriving requirements for millimeter wave radar systems.

At GE he has led groups investigating satellite navigation problems (MILSTAR and BSTS Programs) and has been the chief technical contributor to a growing GE effort to develop "smart" workstation technology.