NASA Technical Memorandum 100911
AIAA-88-3567

A Diagonally Inverted LU Implicit Multigrid Scheme

Jeffrey W. Yokota
Sverdrup Technology, Inc.
NASA Lewis Research Center Group
Cleveland, Ohio

David A. Caughey
Cornell University
Ithaca, New York

and

Rodrick V. Chima
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio

Prepared for the
First National Fluid Dynamics Congress
cosponsored by the AIAA, ASME, ASCE, SIAM, and APS
Cincinnati, Ohio, July 24–28, 1988

NASA

A Diagonally Inverted LU Implicit Multigrid Scheme

Jeffrey W. Yokota
Sverdrup Technology, Inc.
NASA Lewis Research Center Group
Cleveland, Ohio

David A. Caughey
Sibley School of Mechanical And Aerospace Engineering
Cornell University
Ithaca, New York

Rodrick V. Chima
NASA Lewis Research Center
Cleveland, Ohio

Abstract

A new Diagonally Inverted LU Implicit scheme is developed within the framework of the multigrid method for the three-dimensional unsteady Euler equations. The matrix systems that are to be inverted in the LU scheme are treated by local diagonalizing transformations that decouple them into systems of scalar equations. Unlike the Diagonalized ADI method, the time accuracy of the LU scheme is not reduced since the diagonalizing procedure does not destroy time conservation. Even more importantly, this diagonalization significantly reduces the computational effort required to solve the LU approximation and therefore transforms it into a more efficient method of numerically solving the three-dimensional Euler equations.

1. Introduction

Increasingly, attention is being focused upon implicit schemes as the need to develop more efficient numerical schemes becomes apparent. An implicit approximation is an attractive means of increasing the efficiency of the time-marching technique since an implicit scheme allows one to take much larger time steps than are normally allowed in most explicit methods. The drawback of most implicit schemes is that they are computationally more expensive, per time step, than explicit methods and thus interest is focused upon decreasing the computational effort required to solve the implicit approximation. Examples of schemes that address this issue are Chaussee and Pulliam's Diagonalized ADI[1], Obayashi and Fujii's LU-ADI[2] (based on the unsteady Euler and Navier-Stokes equations) and Yoon and Jameson's LU-SSOR[3] (for the steady equations).

The three-dimensional unsteady Euler equations can be discretized into an implicit approximation that, when written in delta-form, produces a large block banded matrix system that is impractical to solve without first being approximately factored. In the present work, the LU Implicit Multigrid algorithm developed by Yokota and Caughey[4] for the three-dimensional unsteady Euler equations has been made more efficient through a local diagonalizing procedure that reduces the computational effort required to solve the LU approximation. This new Diagonally Inverted LU scheme is related to the LU factorization first proposed by Jameson and Turkel[5] and applied to the two-dimensional Euler equations by Buratynski and Caughey[6]. The LU factorization produces two block triangular operators (one upper and one lower) which, through back substitution, can be solved by effectively explicit sweeps that require matrix inversions at every mesh cell in the domain. The efficiency of the LU scheme can be significantly increased by reducing the computational costs associated with these matrix inversions. This reduction can be achieved by a local diagonalizing transformation that decouples the matrix systems into scalar equations. Time conservation and stability are not altered by the decoupling since the LU scheme's differencing operators are not affected by the diagonalizing transformations. This result is in contrast to the Diagonalized ADI method, which is produced at the expense of time accuracy. Thus the solution of the Diagonally Inverted LU scheme requires less computational effort than the original LU scheme and remains both unconditionally stable and time accurate. The Euler equations are numerically transformed to a generalized coordinate system and then discretized into a finite volume approximation that is supplemented with an adaptive blend of second and fourth difference artificial dissipation. Numerical results show that the Diagonally Inverted LU scheme requires less than 1/4 the CPU time the original LU scheme would otherwise need. In addition, the diagonally Inverted LU scheme is developed within the framework of the multigrid method to accelerate the convergence rate for steady state calculations. Results for turbomachinery flows are presented to illustrate the Diagonally Inverted LU scheme's ability to calculate Euler flows and the convergence acceleration produced by the multigrid method.

2. Analysis

The unsteady three-dimensional Euler equations are written in divergence form and then transformed from the Cartesian coordinate system (x, y, z) to the generalized system (ξ, η, ς). The resulting equations can be written:

$$\frac{\partial \vec{W}}{\partial t} + \frac{\partial \vec{F}}{\partial \xi} + \frac{\partial \vec{G}}{\partial \eta} + \frac{\partial \vec{H}}{\partial \varsigma} = 0 \qquad (1)$$

with

$$\vec{W} = \begin{pmatrix} \rho D \\ \rho D u \\ \rho D v \\ \rho D w \\ \rho E \end{pmatrix} \qquad \vec{F} = \begin{pmatrix} \rho DU \\ \rho DUu + P(y_\eta z_\varsigma - y_\varsigma z_\eta) \\ \rho DUv + P(x_\varsigma z_\eta - x_\eta z_\varsigma) \\ \rho DUw + P(x_\eta y_\varsigma - x_\varsigma y_\eta) \\ DU(E+P) \end{pmatrix}$$

$$\vec{G} = \begin{pmatrix} \rho DV \\ \rho DVu + P(y_\varsigma z_\xi - y_\xi z_\varsigma) \\ \rho DVv + P(x_\xi z_\varsigma - x_\varsigma z_\xi) \\ \rho DVw + P(x_\varsigma y_\xi - x_\xi y_\varsigma) \\ DV(E+P) \end{pmatrix}$$

$$\vec{H} = \begin{pmatrix} \rho DW \\ \rho DWu + P(y_\xi z_\eta - y_\eta z_\xi) \\ \rho DWv + P(x_\eta z_\xi - x_\xi z_\eta) \\ \rho DWw + P(x_\xi y_\eta - x_\eta y_\xi) \\ DW(E+P) \end{pmatrix}$$

where ρ and P are the fluid density and pressure, (u, v, w) are the Cartesian velocity components and E is the total energy per unit volume. The total energy and pressure of a calorically perfect gas are related through the equation of state:

$$P = (\gamma - 1)\left(E - \rho\frac{\vec{v}\cdot\vec{v}}{2}\right)$$

The Jacobians of the coordinate transformation are as follows:

$$J = \begin{pmatrix} x_\xi & x_\eta & x_\varsigma \\ y_\xi & y_\eta & y_\varsigma \\ z_\xi & z_\eta & z_\varsigma \end{pmatrix}$$

$$J^{-1} = \begin{pmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \varsigma_x & \varsigma_y & \varsigma_z \end{pmatrix}$$

$$= \frac{1}{D}\begin{pmatrix} y_\eta z_\varsigma - y_\varsigma z_\eta & x_\varsigma z_\eta - x_\eta z_\varsigma & x_\eta y_\varsigma - x_\varsigma y_\eta \\ y_\varsigma z_\xi - y_\xi z_\varsigma & x_\xi z_\varsigma - x_\varsigma z_\xi & x_\varsigma y_\xi - x_\xi y_\varsigma \\ y_\xi z_\eta - y_\eta z_\xi & x_\eta z_\xi - x_\xi z_\eta & x_\xi y_\eta - x_\eta y_\xi \end{pmatrix}$$

where D is the determinant of the matrix J and the contravariant velocities are defined as:

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = J^{-1}\begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

The transformed Euler equations are discretized by a finite volume formulation that approximates the spatial differences as a net flux across the faces of each mesh cell. Global conservation and the admission of possible uniform flow solutions are insured by evaluating the flux vectors on the faces of the boundary-conforming mesh cells. This procedure requires that both the flow variables and the flux-embedded geometric quantities be defined on the faces of the mesh cells during the flux evaluations although it is the cell-averaged flow variables that are calculated during the time and spatial marching. The geometric quantities are evaluated directly on the cell faces while the flow variables are averaged over values found in adjacent cells. The unsteady equations can be discretized into an implicit approximation that when written in a linearized delta form, produces a numerical scheme whose steady state solutions are independent of the time step size used in the time marching. The delta form, produced by linearizing the changes in the flux vectors through a Taylor series expansion about a time level n, can be written as follows:

$$\left[I + \mu\Delta t(\delta_\xi A + \delta_\eta B + \delta_\varsigma C)\right]\Delta\vec{W}^n_{ijk} =$$
$$-\Delta t(\delta_\xi\vec{F} + \delta_\eta\vec{G} + \delta_\varsigma\vec{H})^n_{ijk} \qquad (2)$$

where

$$\Delta\vec{W}^n = \vec{W}^{n+1} - \vec{W}^n$$

and Δt is the time step size; $0 \le \mu \le 1$ is a parameter governing the degree of implicitness; δ and $\bar{\delta}$ are cell- and face-centered central differences; I is the identity matrix; and A, B, and C are the flux Jacobian matrices relative to the vectors $\vec{F}$, $\vec{G}$, and $\vec{H}$.

3. Artificial Dissipation

The finite volume formulation reduces to a central difference approximation on a uniform grid and therefore requires the addition of explicit artificial dissipation terms to suppress possible odd and even point oscillations and shock overshoots. Following the works of Jameson[7] and Pulliam[8], fourth difference terms are added throughout the flow field to prevent odd-even decoupling while second difference terms are used to stablize the flow calculation near shocks.

The conservatively added dissipative term is an adaptive nonlinear blend of second and fourth differences that acts to turn on the second and turn off the fourth difference terms near a shock. The dissipative terms added are of the form:

$$\vec{T}_i = -\bar{\delta}_\xi\kappa^{(2)}D\delta_\xi\frac{1}{D}\vec{W}_i + \bar{\delta}_\xi\kappa^{(4)}D\bar{\delta}_{\xi\xi\xi}\frac{1}{D}\vec{W}_i \qquad (3)$$

where a shock sensing term:

$$\nu_i = \frac{|P_{i+1} - 2P_i + P_{i-1}|}{P_{i+1} + 2P_i + P_{i-1}}$$

is used to define the nonlinear scaling factors:

$$\kappa^{(2)}_{i+\frac{1}{2}} = \frac{\chi^{(2)}}{\Delta t'_{i+\frac{1}{2}}} max(\nu_{i+2}, \nu_{i+1}, \nu_i, \nu_{i-1})$$

$$\kappa^{(4)}_{i+\frac{1}{2}} = max\left(0, \frac{\chi^{(4)}}{\Delta t'_{i+\frac{1}{2}}} - \kappa^{(2)}_{i+\frac{1}{2}}\right)$$

where $\chi^{(2)}$ and $\chi^{(4)}$ are scalar constants and $\Delta t'$ is a one-dimensional unit Courant number time step scaling factor.

4. LU Factorization

The block-banded implict delta operator is factored into two block triangular operators to circumvent the large amounts of CPU time and temporary storage needed to solve the unfactored implicit approximation. The LU factorization, which is based on one-sided, implicit, spatial differences, can be written as follows:

$$\left[I + \mu\Delta t(\delta^-_\xi A_1 + \delta^-_\eta B_1 + \delta^-_\varsigma C_1)\right]\cdot$$
$$\left[I + \mu\Delta t(\delta^+_\xi A_2 + \delta^+_\eta B_2 + \delta^+_\varsigma C_2)\right]\Delta\vec{W}^n_{ijk} =$$
$$-\Delta t(\delta_\xi\vec{F} + \delta_\eta\vec{G} + \delta_\varsigma\vec{H} + \vec{T})^n_{ijk} \qquad (4)$$

where δ^+ and δ^- are cell-centered forward and backward first differences and the flux Jacobian matrices are split and reconstructed as:

$$A_1 = \frac{(A + \beta|A|I)}{2} \qquad A_2 = \frac{(A - \beta|A|I)}{2}$$

where

$$|A| = max(|\lambda_A|)$$

is the maximum absolute-valued eigenvalue of the Jacobian matrix A, $\beta \approx 1$ is a scalar constant governing the amount of implicit dissipation produced by the matix reconstructions and I is the identity matrix. This splitting is chosen to insure a nonsingular diagonal inversion and is similar to the Jameson and Turkel[6] splitting that produces diagonally dominant implicit factors for each matrix inversion in every mesh cell[4]. The implicit system of equations is solved in the following two steps:

1) Lower Sweep

$$\left[I + \mu\Delta t(\delta_\xi^- A_1 + \delta_\eta^- B_1 + \delta_\varsigma^- C_1)\right]\Delta\vec{Y}_{ijk}^n =$$

$$-\Delta t(\delta_\xi \vec{F} + \delta_\eta \vec{G} + \delta_\varsigma \vec{H} + \vec{T})_{ijk}^n \qquad (5)$$

The right hand side of this system, the *Residual*, is evaluated using the finite volume formulation of the numerical approximation and then inverted by an effectively explicit sweep through the domain in the positive $\xi-, \eta-, \varsigma-$ directions. The resulting intermediate variables $\Delta\vec{Y}^n$ are then used to start the upper sweep.

2) Upper Sweep

$$\left[I + \mu\Delta t(\delta_\xi^+ A_2 + \delta_\eta^+ B_2 + \delta_\varsigma^+ C_2)\right]\Delta\vec{W}_{ijk}^n = \Delta\vec{Y}_{ijk}^n \qquad (6)$$

An effectively explicit sweep through the domain is also required to solve this system, but in directions opposite to those taken in the lower sweep. The resulting flow field corrections are then used to update the flow field.

$$\vec{W}_{ijk}^{n+1} = \Delta\vec{W}_{ijk}^n + \vec{W}_{ijk}^n$$

The LU factorization requires the solution of two block triangular operators each of which, through back substitution, can be reduced to simple 5x5 matrix systems at every mesh cell. These reduced systems can be written:

1) Lower Sweep

$$\left[I + \mu\Delta t(A_1 + B_1 + C_1)\right]\Delta\vec{Y}_{ijk}^n =$$

$$-\Delta t(\bar{\delta}_\xi \vec{F} + \bar{\delta}_\eta \vec{G} + \bar{\delta}_\varsigma \vec{H} + \vec{T})_{ijk}^n$$

$$+\mu\Delta t(A_1\Delta\vec{Y}_{i-1,j,k}^n + B_1\Delta\vec{Y}_{i,j-1,k}^n + C_1\Delta\vec{Y}_{i,j,k-1}^n) \qquad (7)$$

2) Upper Sweep

$$\left[I - \mu\Delta t(A_2 + B_2 + C_2)\right]\Delta\vec{W}_{ijk}^n = \Delta\vec{Y}_{ijk}^n$$

$$-\mu\Delta t(A_2\Delta\vec{W}_{i+1,j,k}^n + B_2\Delta\vec{W}_{i,j+1,k}^n + C_2\Delta\vec{W}_{i,j,k+1}^n) \qquad (8)$$

and are typically inverted by Gaussian elimination.

5. Diagonal Inversion

The solution of the LU factorization can be made more efficient by diagonally inverting its 5x5 matrix systems. The diagonal inversion is based on the diagonalizing procedure used to classify the Euler equations formally as a hyperbolic system of partial differential equations.

The Euler equations are classified as an hyperbolic system of partial differential equations since a similarity transformation exists such that:

$$Q^{-1}(k_1 A + k_2 B + k_3 C)Q = \Lambda \qquad (9)$$

where Λ is a diagonal matrix whose elements are real, and k_1, k_2, and k_3 are scalar constants. For the specific case of $k_1 = k_2 = k_3 = 1$, a corresponding diagonalizing transformation can be derived from the general similarity transformation developed for the Euler equations in Cartesian coordinates by Warming, Beam, and Hyett[9]. The matrix Q has elements

$$Q_{11} = \hat{l}_1 \qquad Q_{21} = u\hat{l}_1$$
$$Q_{12} = \hat{l}_2 \qquad Q_{22} = u\hat{l}_2 - \rho\hat{l}_3$$
$$Q_{13} = \hat{l}_3 \qquad Q_{23} = u\hat{l}_3 + \rho\hat{l}_2$$
$$Q_{14} = \frac{\rho}{\sqrt{2}c} \qquad Q_{24} = \frac{\rho u}{\sqrt{2}c} + \frac{\rho\hat{l}_1}{\sqrt{2}}$$
$$Q_{15} = \frac{\rho}{\sqrt{2}c} \qquad Q_{25} = \frac{\rho u}{\sqrt{2}c} - \frac{\rho\hat{l}_1}{\sqrt{2}}$$

$$Q_{31} = v\hat{l}_1 + \rho\hat{l}_3 \qquad Q_{41} = w\hat{l}_1 - \rho\hat{l}_2$$
$$Q_{32} = v\hat{l}_2 \qquad Q_{42} = w\hat{l}_2 + \rho\hat{l}_1$$
$$Q_{33} = v\hat{l}_3 - \rho\hat{l}_1 \qquad Q_{43} = w\hat{l}_3$$
$$Q_{34} = \frac{\rho v}{\sqrt{2}c} + \frac{\rho\hat{l}_2}{\sqrt{2}} \qquad Q_{44} = \frac{\rho w}{\sqrt{2}c} + \frac{\rho\hat{l}_3}{\sqrt{2}}$$
$$Q_{35} = \frac{\rho v}{\sqrt{2}c} - \frac{\rho\hat{l}_2}{\sqrt{2}} \qquad Q_{45} = \frac{\rho w}{\sqrt{2}c} - \frac{\rho\hat{l}_3}{\sqrt{2}}$$

$$Q_{51} = \frac{q^2}{2}\hat{l}_1 + \rho v\hat{l}_3 - \rho w\hat{l}_2$$
$$Q_{52} = \frac{q^2}{2}\hat{l}_2 - \rho u\hat{l}_3 + \rho w\hat{l}_1$$
$$Q_{53} = \frac{q^2}{2}\hat{l}_3 + \rho u\hat{l}_2 - \rho v\hat{l}_1$$
$$Q_{54} = \frac{\rho q^2}{2\sqrt{2}c} + \frac{\rho u\hat{l}_1}{\sqrt{2}} + \frac{\rho v\hat{l}_2}{\sqrt{2}} + \frac{\rho w\hat{l}_3}{\sqrt{2}} + \frac{\rho c}{\sqrt{2}(\gamma - 1)}$$
$$Q_{55} = \frac{\rho q^2}{2\sqrt{2}c} - \frac{\rho u\hat{l}_1}{\sqrt{2}} - \frac{\rho v\hat{l}_2}{\sqrt{2}} - \frac{\rho w\hat{l}_3}{\sqrt{2}} + \frac{\rho c}{\sqrt{2}(\gamma - 1)}$$

where

$$q^2 = u^2 + v^2 + w^2$$
$$l_1 = \xi_x + \eta_x + \varsigma_x$$
$$l_2 = \xi_y + \eta_y + \varsigma_y$$
$$l_3 = \xi_z + \eta_z + \varsigma_z$$

$$\hat{l}_1 = \frac{l_1}{\sqrt{l_1^2+l_2^2+l_3^2}} \qquad \hat{l}_2 = \frac{l_2}{\sqrt{l_1^2+l_2^2+l_3^2}}$$
$$\hat{l}_3 = \frac{l_3}{\sqrt{l_1^2+l_2^2+l_3^2}}$$

and c is the local speed of sound.

This diagonalizing transformation can be used to produce the diagonal matrix

$$Q^{-1}(A+B+C)Q = \Lambda$$

which has elements

$$\lambda_{11} = \lambda_{22} = \lambda_{33} = U+V+W$$
$$\lambda_{44} = U+V+W - c\sqrt{l_1^2+l_2^2+l_3^2}$$
$$\lambda_{55} = U+V+W + c\sqrt{l_1^2+l_2^2+l_3^2}$$
$$\lambda_{ij} = 0 \qquad when \qquad i \neq j$$

Applying this local similarity transformation to the lower and upper sweeps produces the following scalar systems:

1) Lower Sweep

$$\left[I + \frac{\mu\Delta t}{2}\Big(\Lambda + \beta(|A|+|B|+|C|)I\Big)\right]Q^{-1}\Delta\vec{Y}^n_{ijk} =$$
$$-\Delta t Q^{-1}\Big((\delta_\xi \vec{F} + \delta_\eta \vec{G} + \delta_\varsigma \vec{H} + \vec{T})^n_{ijk}$$
$$-\mu(A_1\Delta\vec{Y}^n_{i-1,j,k} + B_1\Delta\vec{Y}^n_{i,j-1,k} + C_1\Delta\vec{Y}^n_{i,j,k-1})\Big) \quad (10)$$

2) Upper Sweep

$$\left[I - \frac{\mu\Delta t}{2}\Big(\Lambda - \beta(|A|+|B|+|C|)I\Big)\right]Q^{-1}\Delta\vec{W}^n_{ijk} =$$
$$Q^{-1}\Big(\Delta\vec{Y}^n_{ijk} - \mu\Delta t(A_2\Delta\vec{W}^n_{i+1,j,k}$$
$$+B_2\Delta\vec{W}^n_{i,j+1,k} + C_2\Delta\vec{W}^n_{i,j,k+1})\Big) \quad (11)$$

The lower and upper sweeps have now been transformed into uncoupled systems of scalar equations that retain the stability and time accuracy of the original LU factorization. Unlike in the Diagonalized ADI method, time conservation is not altered by the decoupling process since the similarity transformation is not factored out of the implicit spatial differences. For a time step defined as

$$\Delta t = \frac{Cn}{(|A|+|B|+|C|)}$$

where Cn is the Courant number used in the time marching, the vector components $(m = 1, ..., 5)$ of the lower and upper sweeps can now be written and solved as follows.

1) Lower Sweep

$$(Q^{-1}\Delta\vec{Y}^n_{ijk})_m =$$
$$\frac{\begin{pmatrix} -\Delta t Q^{-1}\Big((\delta_\xi \vec{F} + \delta_\eta \vec{G} + \delta_\varsigma \vec{H} + \vec{T})^n_{ijk} \\ -\mu(A_1\Delta\vec{Y}^n_{i-1,j,k} + B_1\Delta\vec{Y}^n_{i,j-1,k} + C_1\Delta\vec{Y}^n_{i,j,k-1})\Big)\end{pmatrix}_m}{\Big((1+\frac{\mu\beta Cn}{2})I + \frac{\mu\Delta t}{2}\Lambda\Big)_m} \quad (12)$$

2) Upper Sweep

$$(Q^{-1}\Delta\vec{W}^n_{ijk})_m =$$
$$\frac{\begin{pmatrix} Q^{-1}\Big(\Delta\vec{Y}^n_{ijk} - \mu\Delta t(A_2\Delta\vec{W}^n_{i+1,j,k} \\ +B_2\Delta\vec{W}^n_{i,j+1,k} + C_2\Delta\vec{W}^n_{i,j,k+1})\Big)\end{pmatrix}_m}{\Big((1+\frac{\mu\beta Cn}{2})I - \frac{\mu\Delta t}{2}\Lambda\Big)_m} \quad (13)$$

and

$$\vec{W}^{n+1}_{ijk} = Q\Big(Q^{-1}\Delta\vec{W}^n_{ijk}\Big) + \vec{W}^n_{ijk}$$

The solution of these scalar equations requires significantly less computional effort than the original matrix equations. The diagonally inverted lower sweep requires approximately 855 operation counts per mesh cell as opposed to the over 1453 operation counts per mesh cell required by the original lower sweep (a similar reduction is produced in the upper sweep). On a scalar machine, this reduction in operation counts translates into a 30-35 percent reduction in CPU time, but on a vector machine the code can be restructured to produce an even greater increase in efficiency due to increased vectorization. All outward subroutine calls and short inner loops can be eliminated from the lower and upper sweeps, thereby allowing most of the non-recursive parts of the code to vectorize. This decrease in operation counts, together with the partial vectorization of the previously unvectorizable lower and upper sweeps, significantly increases the computational efficiency of the LU factorization.

6. Initial and Boundary Conditions

Initial conditions are needed to start the calculations and are chosen to be a uniform flow field based on an initial guess of the upstream flow.

During the residual calculation, a no-flux condition is enforced at the solid boundaries by setting to zero the contravariant velocity component normal to the boundary. This treatment requires that only pressure needs to be specified along solid boundaries and is obtained from a three-dimensional interpretation of the normal momentum anaylsis developed by Rizzi[10]. For a solid boundary aligned with an $\eta - \varsigma$ plane, the normal momentum analysis requires that the following equation be satisfied.

$$a_1\frac{\partial P}{\partial \xi} + a_2\frac{\partial P}{\partial \eta} + a_3\frac{\partial P}{\partial \varsigma} = b_1 + b_2 \quad (14)$$

where

$$a_1 = (D\xi_x)^2 + (D\xi_y)^2 + (D\xi_z)^2$$
$$a_2 = D^2\xi_x\eta_x + D^2\xi_y\eta_y + D^2\xi_z\eta_z$$
$$a_3 = D^2\xi_x\varsigma_x + D^2\eta_y\varsigma_y + D^2\xi_z\varsigma_z$$
$$b_1 = \rho DV\left(u\frac{\partial(D\xi_x)}{\partial\eta} + v\frac{\partial(D\xi_y)}{\partial\eta} + w\frac{\partial(D\xi_z)}{\partial\eta}\right)$$
$$b_2 = \rho DW\left(u\frac{\partial(D\xi_x)}{\partial\varsigma} + v\frac{\partial(D\xi_y)}{\partial\varsigma} + w\frac{\partial(D\xi_z)}{\partial\varsigma}\right)$$

For a steady subsonic inflow boundary, total pressure, total temperature and two flow angles are specified while a one-dimensional Rieman invariant

$$R = q - \frac{2c}{\gamma - 1}$$

is extrapolated from the interior flow field (similar to the treatment used by Chima[11]).

For steady subsonic outflow, the specified boundary condition is a nonreflective treatment, based on the work of Rudy and Strikwerda[12], that attempts to minimizes unwanted reflected waves from the outflow boundary. Static pressures are obtained from a radial momentum analysis

$$\frac{1}{\rho}\frac{\partial P}{\partial r} = \frac{U_\theta}{r}$$

(r is radial distance and U_θ^2 is tangential velocity) and then coupled with the incoming compatibility relation to produce a nonreflective boundary condition

$$\frac{\partial P}{\partial t} - \frac{\rho \tilde{C}}{\varsigma_z}\frac{\partial w}{\partial t} + \alpha(P - P_{ref}) = 0 \qquad (15)$$

where

$$\tilde{C} = c\sqrt{\varsigma_x^2 + \varsigma_y^2 + \varsigma_z^2}$$

P_{ref} is the static pressure obtained from the radial momentum equation and $\alpha \approx 1$ is a scalar constant.

7. Steady State Calculations

The calculation of steady state solutions is made more efficient by using local time-stepping and the multigrid method.

Local time-stepping is used to optimize the time step throughout the flow field. A locally varing time step size, based on a constant Courant number, is used to create a warped time integration that can accelerate the calculation to a steady state without affecting the steady state solution.

The multigrid method is incorporated into the Diagonally Inverted LU scheme to accelerate the removal of low frequency errors from the flow solution and thus increase the efficiency of the time-marching procedure. Following the work of Jameson[13] and Jameson and Yoon[14], the flow solver is used to smooth out high frequency errors resolvable on any current grid level, while the multigrid method is used to eliminate low frequency errors through a sequence of flow calculations on coarser grids. The multigrid sequencing used is a three-level W- cycle that incorporates a double pass on the coarse grids. Coarse grid boundary conditions are identical to those used on the fine grid with the exception of the inflow/outflow conditions which are updated only on the fine grids. Coarse grid residuals are kept smooth by adding only a constant coefficient second difference artificial dissipation term (the nonlinear blended terms are used only on the fine grid). This treatment attempts to limit the amount of high frequency errors reintroduced into the flow field by the upward interpolation of the coarse grid corrections. The flow solver, in this case the Diagonally Inverted LU scheme, is invoked only once on each grid level and only before transfering the flow field to the next coarser grid. The multigrid cycle defined above requires 41/32 work units of computational effort where work units are normalized by a single flow calculation on the finest grid.

8. Results

Numerical results are presented to illustrate the Diagonally Inverted LU scheme's ability to calculate three-dimensional turbomachinery flows and the convergence acceleration produced by the multigrid method.

All results were calculated on H-type grids consisting of 64x16x16 mesh cells in the throughflow, blade-to-blade, and radial directions, respectively. The grids were generated using a modified version of the GRAPE code[15] originally developed by Sorenson[16]. All calculations were performed on a CRAY X-MP, where a calculation consisting of 250 work units required approximately 0.9 million words of memory and 10 minutes of CPU time. This result is in contrast to the original LU scheme for which a comparable calculation required over 42 minutes of CPU time. The diagonal inversion has significantly reduced the computational effort required to solve the LU factorization and has therefore transformed it into a more efficient numerical scheme.

The first test case was a stationary (rot.= 0) cascade passage whose blade shapes were derived from the NACA 0012 airfoil. The three-dimensional grid shown in Figure 1 is based on a 36 blade axial rotor with a 0.825 hub-to-tip radius ratio and a 14.3 tip radius-to-blade chord ratio. Flow conditions of 0.66 exit static-to-inflow total pressure ratio (on the hub, P.R.= 0.66) and axial relative inflow (angy = 90, angz = 0) were used to produce a transonic flow through the turbomachinery passage. This case is used to illustrate the numerical scheme's ability to calculate transonic flows and the convergence acceleration produced by the multigrid method.

Convergence histories for calculations on a single grid and with the multigrid method are shown in Figures 2 and 3. Both calculations were run at a Courant number of 10 and with local time-stepping. The logarithm of the flow field error and a measure of the developing supersonic region are plotted versus the work units required to produce these results. The flow field error is characterized by the logarithm of the rms rate of change of the continuity equation on the finest grid (scaled by its initial value), while the developing supersonic region (a measure of global convergence) is represented by the number of grid cells (NSUP) found to contain supersonic flow (scaled by its final value). The calculation performed on a single grid (Figure 2) shows a reduction in the average residual of almost 3.5 orders of magnitude after 800 work units. The supersonic pocket has not yet converged to its final value after 800 work units, which suggests that the flow field error should be further reduced (Res1 is the calculation's starting residual while Res2 is the final residual). The multigrid method is incorporated into the time-stepping technique to produce a satisfactory reduction in the flow field error without having to resort to a large number of time steps. Figure 3 shows the convergence history of the three-level W-cycle, where a fully developed supersonic region was produced in approximately 150 work units and 4.5 order of magnitude drop in the flow field error was produced within 250 work units. The use of the multigrid method has clearly accelerated the convergence rate of this steady state calculation.

The blade surface and blade-to-blade Mach number distributions at mid-span are shown in Figures 4 and 5 to illustrate the accuracy and the shock-capturing capabilities of the Diagonally Inverted LU scheme. Near the blades the flow is accelerated to a peak Mach number of 1.28 and then shocked down to subsonic flow through a shock that is captured over three mesh cells.

The second test case is the Annular Cascade designed and extensively tested at NASA Lewis[17,18]. The computational grid shown in Figure 6 is based on the full annular ring of 36 core turbine stator vanes. The geometry is a 38.10 mm high untwisted blade of constant profile with an axial chord of 38.23 mm. The stator has a tip diameter of 508 mm and a 0.85 hub-to-tip radius ratio.

Test conditions of ambient axial inflow and a 0.65 hub-static to inlet-total pressure ratio produce a flow field with mean radius inlet and exit critical velocity ratios of 0.231 and 0.778 respectively. To match the upstream flow conditions (an inflow Mach number of 0.211), the Euler calculations were run with a 0.66 hub-static to inlet-total pressure ratio.

Figure 7 shows the convergence history of the Annular Cascade calculation (with multigrid) where a drop of 5 orders of magnitude in the flow field error was produced within 250 work units.

The resulting flow field is fully subsonic and compared with experimental data at three spanwise positions. Figures 8, 9, and 10 compare the calculated blade surface static pressure distributions at 13.3, 50, and 86.7 percent span with the experimental data produced by Goldman and Seasholtz[18]. The trailing edge flow is not well captured due to the bluntness of the blade's trailing edge but the overall flow field agrees well with the experimental data. Figure 11 shows the blade-to-blade Mach number distribution at the mid-span position where the flow is turned approximately 67 degrees from the axial inflow. Figure 12 shows the critical velocity ratio distribution along the 50 percent radial span plane at the 150 percent axial chord location. The Euler equations cannot capture the viscous wake effects but the overall flow field does seem to be adequately modelled. The computational results compare well with the experimental data and demonstrate the scheme's ability to calculate complex flows accurately.

9. Concluding Remarks

A new Diagonally Inverted LU Implicit scheme has been developed for the three-dimensional unsteady Euler equations. The matrix systems that are normally inverted in the LU scheme are treated by local diagonalizing transformations that decouple them, without a loss of time conservation, into systems of scalar equations. This decoupling significantly reduces the computational effort required to solve the LU approximation and therefore transforms it into a more efficient method of numerically solving the unsteady three-dimensional Euler equations. Results illustrate the scheme's ability to calculate accurately three-dimensional flows through turbomachines and the convergence acceleration produced when coupled with the multigrid method.

10. References

[1] Chaussee, D.S., and Pulliam, T.H., "Two-Dimensional Inlet Simulation using a Diagonal Implicit Algorithm," *AIAA Journal*, Vol. 19, No. 2, pp. 153-159, February 1981.

[2] Obayashi, S., and Fujii, K., "Computation of Three-Dimensional Viscous Transonic Flow with the LU Factored Scheme," AIAA Paper 85-1510, 7th Computational Fluid Dynamics Conference, Cincinnati, Ohio, July 1985.

[3] Yoon, S., and Jameson, A., "An LU-SSOR Scheme for the Euler and Navier-Stokes Equations," AIAA Paper 87-0600, 25th Aerospace Sciences Meeting, Reno, Nevada, January 1987.

[4] Yokota, J.W., and Caughey, D.A., "An L-U Implicit Multigrid Algorithm for the Three-Dimensional Euler Equations," AIAA Paper 87-0453, 25th Aerospace Sciences Meeting, Reno, Nevada, January 1987.

[5] Jameson, A., and Turkel, E., "Implicit Schemes and LU Decompositions," *Mathematics of Computation*, Vol. 37, No. 156, pp. 385-397, October 1981.

[6] Buratynski, E.K., and Caughey, D.A., "An Implicit LU Scheme for the Euler Equations applied to Arbitrary Cascades," *AIAA Journal*, Vol. 24, No.1, pp.39-46, January 1986.

[7] Jameson, A., "Transonic Flow Calculations for Aircraft," in *Numerical Methods in Fluid Dynamics Lecture Notes in Mathematics*, Vol. 1127, Brezzi,F., ed., pp. 156-242, Springer-Verlag, New York, 1985.

[8] Pulliam, T.H., "Artificial Dissipation Models for the Euler Equations," AIAA Paper 85-0438 , 23rd Aerospace Sciences Meeting, Reno, Nevada, January 1985.

[9] Warming, R.F., Beam, R.M., and Hyett, B.J., "Diagonalization and Simultaneous Symmetrization of the Gas-Dynamic Matrices," *Mathematics of Computation*, Vol. 29, pp. 1037-1045, October 1975.

[10] Rizzi, A., "Numerical Implementation of the Solid-Body Boundary Conditions for the Euler Equations," Zeitschrift fuer Angewandte Mathematik und Mechanik, Vol. 58, No. 7, pp. 301-304, July 1978.

[11] Chima, R.V., "Analysis of Inviscid and Viscous Flows in Cascades with an Explicit Multiple-Grid Algorithm," NASA TM 83636, 1984.

[12] Rudy, D.H., and Strikwerda, J.C., "A Nonreflecting Outflow Boundary Condition for Subsonic Navier-Stokes Calculations," *Journal of Computational Physics*, Vol. 36, No.1, pp. 55-70, June 1980.

[13] Jameson, A., "Solution of the Euler Equations for Two-Dimensional Transonic Flow by a Multigrid Method," MAE Report 1613, Princeton University, 1983.

[14] Jameson, A., and Yoon, S., "Multigrid Solution of the Euler Equations Using Implicit Schemes," AIAA Paper 85-0293, Reno, Nevada, January 1985.

[15] Smith, W.A., "Modified Grape Code," Sibley School of Mechanical and Aerospace Engineering FDA Report 87-10, Cornell University, June 1987.

[16] Sorenson, R., "A Computer Program to Generate Two Dimensional Grids About Airfoils and Other Shapes by the use of the Poisson's Equation," NASA TM 81198, 1980.

[17] Goldman, L.J., and McLallin, K.L., "Cold-Air Annular Cascade Investigation of Aerodynamic Performance of Core-Engine-Cooled Turbine Vanes. I: Solid-Vane Performance and Facility Description," NASA TM X-3224, 1975.

[18] Goldman, L.J., and Seasholtz, R.G., "Laser Anemometer Measurements in a Annular Cascade of Core Turbine Vanes and Comparison with Theory," NASA TP-2018, 1982.

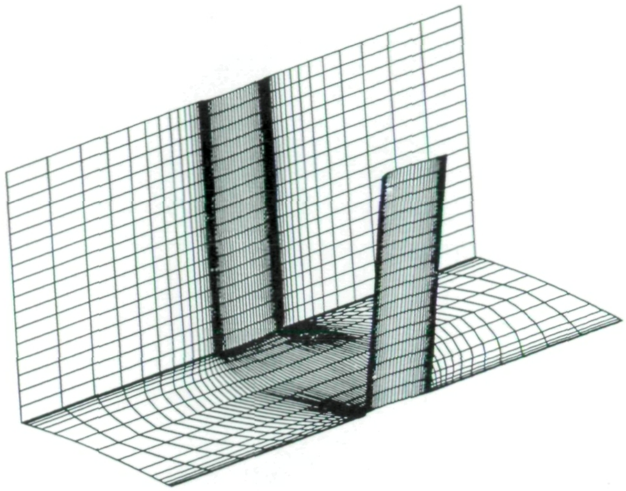


Figure - 1. Computational grid for the NACA 0012 Cascade.

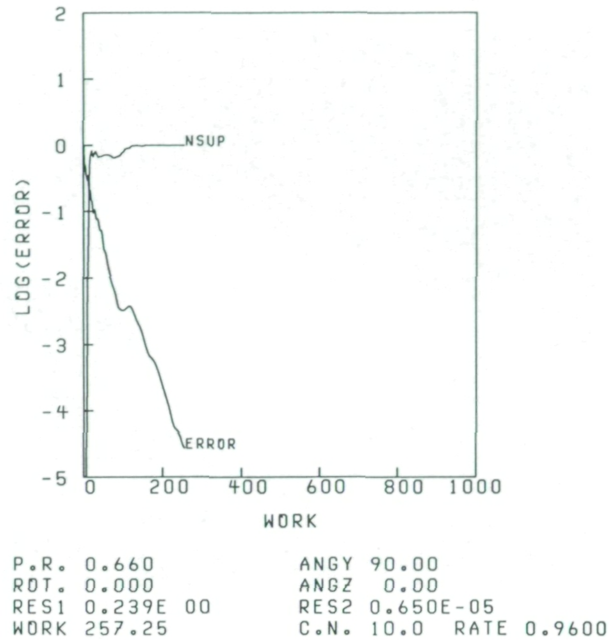


Figure - 3. Convergence history for the multigrid calculation.

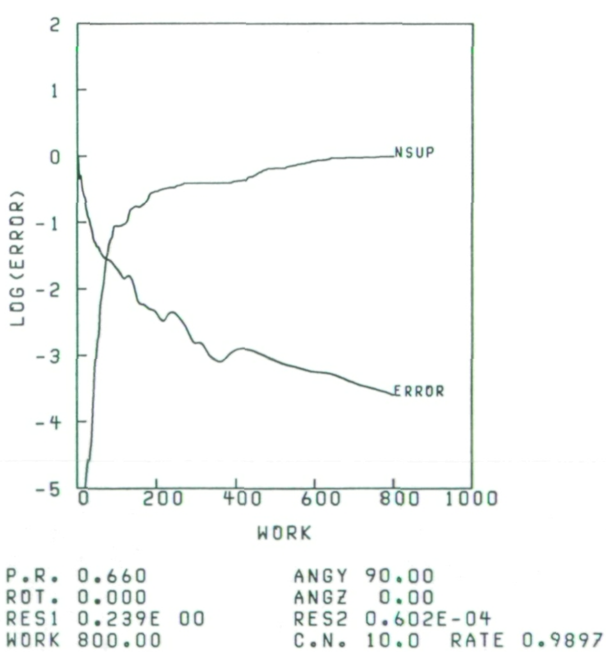


Figure - 2. Convergence history for the single grid calculation.

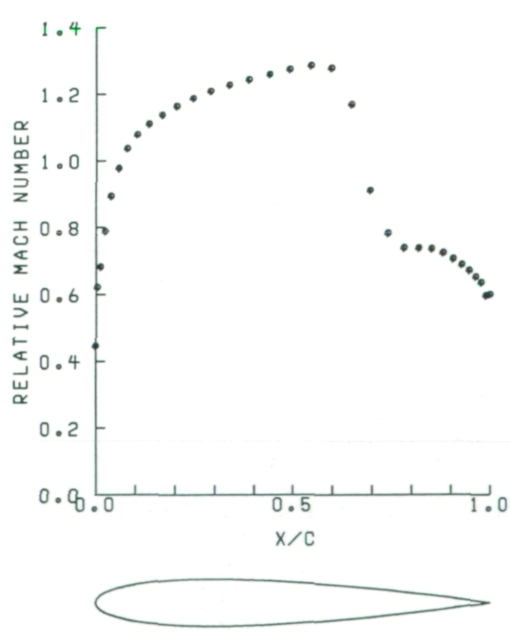


Figure - 4. Mach number blade distribution at 50 percent span.

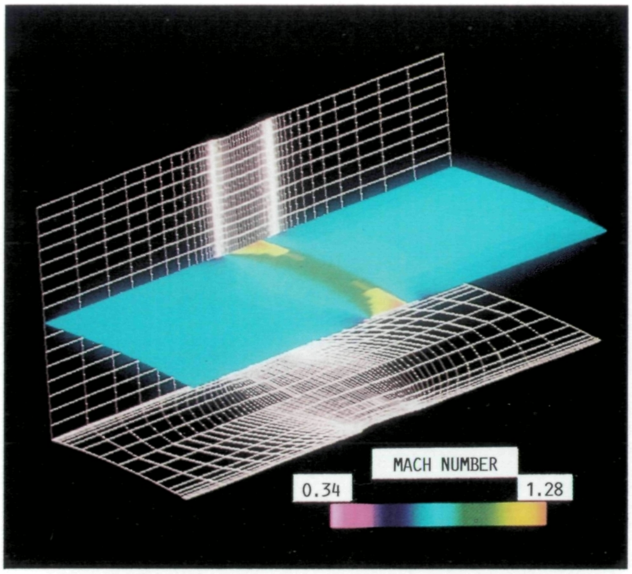


Figure - 5. Mach number distribution at 50 percent span.

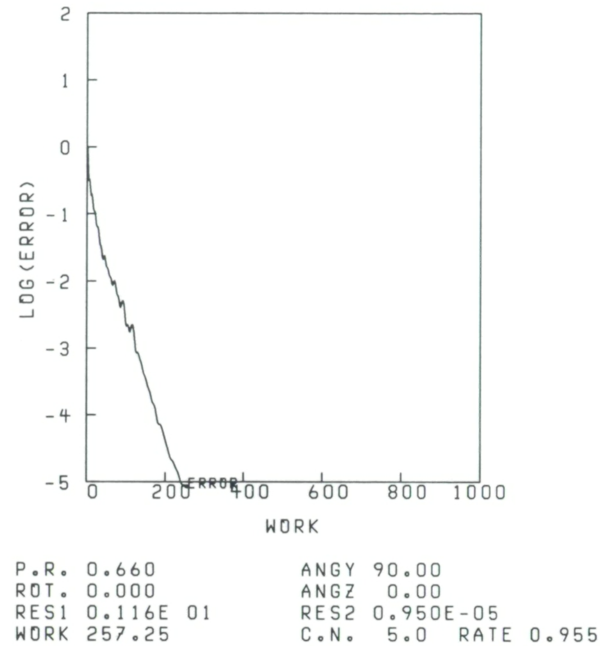


Figure - 7. Convergence history for the multigrid calculation.

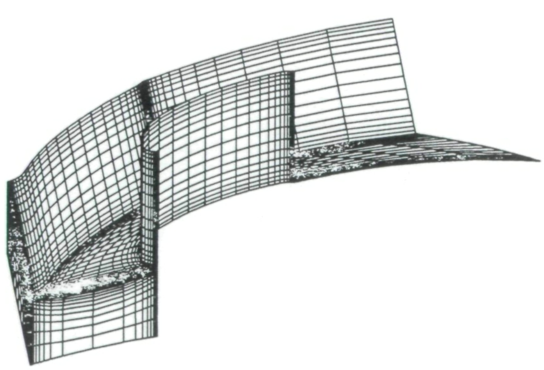


Figure - 6. Computational grid for the Annular Cascade.

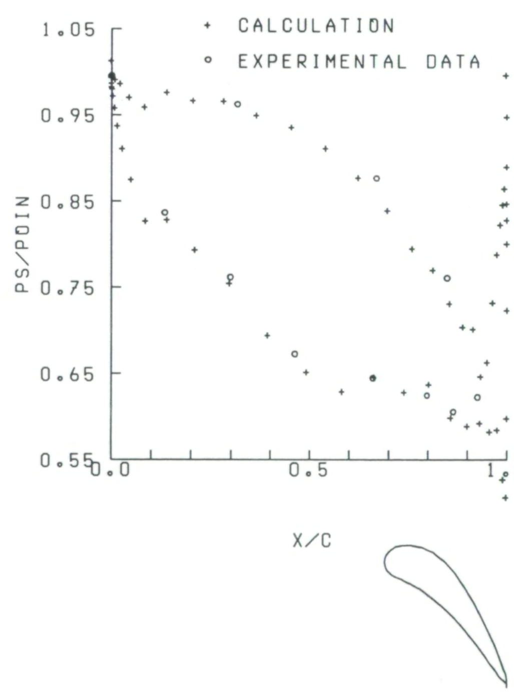


Figure - 8. Static pressure blade distribution at 13.3 percent span.

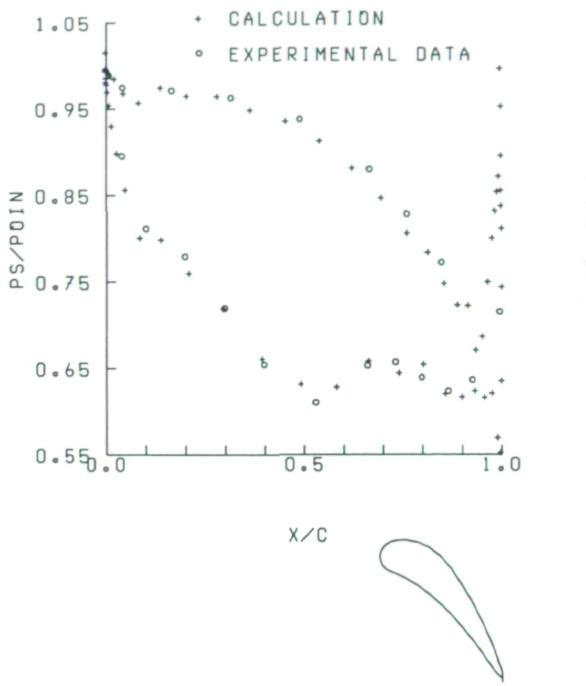


Figure - 9. Static pressure blade distribution at 50 percent span.

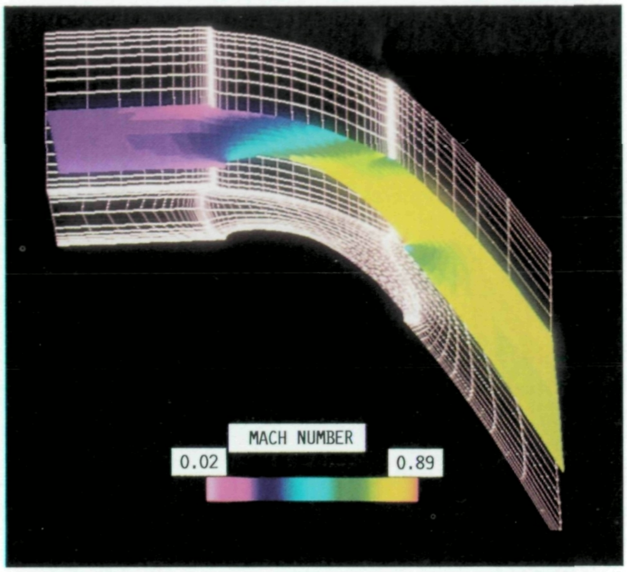


Figure - 11. Mach number distribution at 50 percent span.

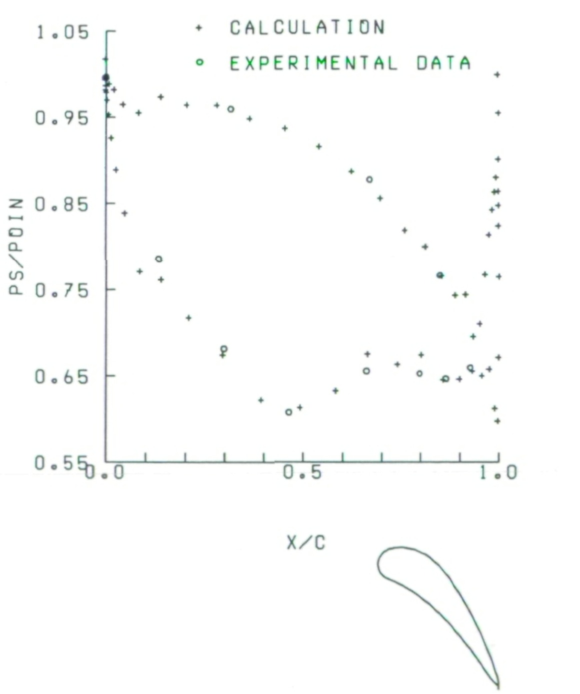


Figure - 10. Static pressure blade distribution at 86.7 percent span.

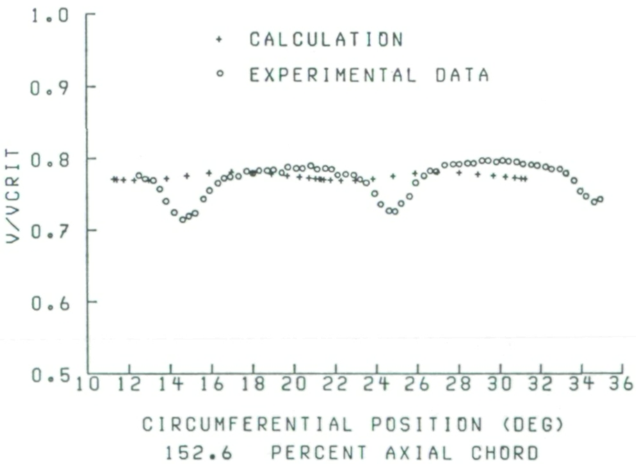


Figure - 12. Critical velocity ratio at 50 percent span.

Report Documentation Page

NASA
National Aeronautics and Space Administration

1. Report No. NASA TM-100911 AIAA-88-3567	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle: A Diagonally Inverted LU Implicit Multigrid Scheme		5. Report Date
		6. Performing Organization Code
7. Author(s): Jeffrey W. Yokota, David A. Caughey, and Rodrick V. Chima		8. Performing Organization Report No. E-4163
		10. Work Unit No. 505-62-21
9. Performing Organization Name and Address: National Aeronautics and Space Administration, Lewis Research Center, Cleveland, Ohio 44135-3191		11. Contract or Grant No.
		13. Type of Report and Period Covered: Technical Memorandum
12. Sponsoring Agency Name and Address: National Aeronautics and Space Administration, Washington, D.C. 20546-0001		14. Sponsoring Agency Code

15. Supplementary Notes

Prepared for the First National Fluid Dynamics Congress cosponsored by the AIAA, ASME, ASCE, SIAM, and APS, Cincinnati, Ohio, July 24-28, 1988. Jeffrey W. Yokota, Sverdrup Technology, Inc., NASA Lewis Research Center Group, Cleveland, Ohio 44135; David A. Caughey, Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, New York 14853; Rodrick V. Chima, NASA Lewis Research Center.

16. Abstract

A new Diagonally Inverted LU Implicit scheme is developed within the framework of the multigrid method for the three-dimensional unsteady Euler equations. The matrix systems that are to be inverted in the LU scheme are treated by local diagonalizing transformations that decouple them into systems of scalar equations. Unlike the Diagonalized ADI method, the time accuracy of the LU scheme is not reduced since the diagonalizing procedure does not destroy time conservation. Even more importantly, this diagonalization significantly reduces the computational effort required to solve the LU approximation and therefore transforms it into a more efficient method of numerically solving the three-dimensional Euler equations.

17. Key Words (Suggested by Author(s)): Turbomachinery; Multigrid; LU implicit; Diagonally inverted; CFD	18. Distribution Statement: Unclassified – Unlimited; Subject Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No of pages	22. Price*
Unclassified	Unclassified		

NASA FORM 1626 OCT 86

*For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA TM-100911 AIAA-88-3567	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A Diagonally Inverted LU Implicit Multigrid Scheme		5. Report Date
		6. Performing Organization Code
7. Author(s) Jeffrey W. Yokota, David A. Caughey, and Rodrick V. Chima		8. Performing Organization Report No. E-4163
		10. Work Unit No. 505-62-21
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546-0001		14. Sponsoring Agency Code

15. Supplementary Notes

Prepared for the First National Fluid Dynamics Congress cosponsored by the AIAA, ASME, ASCE, SIAM, and APS, Cincinnati, Ohio, July 24-28, 1988. Jeffrey W. Yokota, Sverdrup Technology, Inc., NASA Lewis Research Center Group, Cleveland, Ohio 44135; David A. Caughey, Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, New York 14853; Rodrick V. Chima, NASA Lewis Research Center.

16. Abstract

A new Diagonally Inverted LU Implicit scheme is developed within the framework of the multigrid method for the three-dimensional unsteady Euler equations. The matrix systems that are to be inverted in the LU scheme are treated by local diagonalizing transformations that decouple them into systems of scalar equations. Unlike the Diagonalized ADI method, the time accuracy of the LU scheme is not reduced since the diagonalizing procedure does not destroy time conservation. Even more importantly, this diagonalization significantly reduces the computational effort required to solve the LU approximation and therefore transforms it into a more efficient method of numerically solving the three-dimensional Euler equations.

17. Key Words (Suggested by Author(s)) Turbomachinery; Multigrid; LU implicit; Diagonally inverted; CFD	18. Distribution Statement Unclassified – Unlimited Subject Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No of pages	22. Price*
Unclassified	Unclassified	10	A02

NASA FORM 1626 OCT 86

*For sale by the National Technical Information Service, Springfield, Virginia 22161

LANGLEY GRANT
IN-03-CR
149247
P-101

PRINCETON UNIVERSITY
Department of Mechanical and Aerospace Engineering
Princeton, New Jersey

PR-150-6549-1

PROGRESS REPORT ON INTELLIGENT GUIDANCE AND CONTROL FOR WIND SHEAR ENCOUNTER

Six-Month Progress Report for
NASA Grant No.NAG-1-834
Princeton Account No. 150-6549

July 21, 1988

Prepared for:

LANGLEY RESEARCH CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Hampton, Virginia

Submitted by:

Robert F. Stengel
Robert F. Stengel
Professor
Principal Investigator

(NASA-CR-183060) INTELLIGENT GUIDANCE AND CONTROL FOR WIND SHEAR ENCOUNTER Biannual Progress Report (Princeton Univ.) 101 p CSCL 01C

N88-26347

Unclas
G3/03 0149247

Summary

A study of intelligent guidance and control concepts for protecting against the adverse effects of wind shear during aircraft takeoffs and landings has been begun. The principal objectives are to develop methods for assessing the likelihood of wind shear encounter, for deciding what flight path to pursue, and for using the aircraft's full potential for combating wind shear. This study requires the definition of both deterministic and statistical techniques for fusing internal and external information, for making "go/no-go" decisions, and for generating commands to the aircraft's cockpit displays and autopilot for both manually controlled and automatic flight.

The program has begun with the development of a real-time expert system for pilot aiding that is based on the results of the FAA Windshear Training Aids Program. A two-volume manual that presents an overview, pilot guide, training program, and substantiating data provides guidelines for this initial development. The Expert System to Avoid Wind Shear (ESAWS) currently contains over 140 rules and is coded in the LISP programming language for implementation on a Symbolics 3670 LISP Machine.

This six-month progress report includes a brief introduction to ESAWS, as well as numerous appendices that describe the logic and code of the program in considerable detail. This development is especially fluid, and these materials are presented as evidence of progress made to date rather than as documentation of a working system; consequently, the logic is subject to further change, and no guarantee is made that the submitted coding is "bug-free" or even representative of ESAWS's final structure. Program logic is being developed by Professor Stengel and graduate student Alexander Stratton; Mr. Stratton has been responsible for all program coding and has authored a considerable portion of this progress report.

Table of Contents

1. LISP Implementation of Wind Shear Expert System

Wind Shear Expert System software has been written in Common LISP for a Symbolics 3670 LISP machine, using elementary list operations. Execution of the system and input/output operations are carried out in the standard LISP environment, using a "LISP Listener". Currently, the user acts as both flight crew and simulation environment; additions planned for the system, including a FORTRAN aircraft simulation, will add realism for system development and demonstration and will complete the model of flight crew decision-making and control presented in the FAA Windshear Training Aid.

Software for the Wind Shear Expert System consists of parameter definitions (Appendix A), rule bases (Appendix B), and functional LISP procedures (Appendices C to F). Rule bases are defined as LISP variables using the function "*defvar*". A rule base is simply a list of related rules (see Appendix B for English translations of rules). In the code, each rule is a list containing seven elements. The first element of a rule is the rule's name. Two other elements, called the"if-part" or *premise* and the "then-part" or *consequent*, are executable LISP statements that are evaluated if the rule is tried. Two of the elements are character strings that give an English translation of the rule; these strings are typed to a LISP Listener if the premise is evaluated and does not return "*NIL*".

The two remaining elements of a rule are lists of parameters. One list consists of the variables whose values are set if the consequent is evaluated. This list is a "flag" for a goal-directed search procedure. The other parameter list consists of variables that must have values for the premise to be evaluated; this list is a "flag" for a forward-chaining procedure and is not presently used. Editing features of the Symbolics machine enable nested lists such as the rule bases to be easily entered into files in a neat format that simplifies identification and later modification. Grouping the rules into separate rule bases breaks down the expert system functions into sub-functions in a natural manner.

Expert system parameters store data for the expert system, and are objects of search procedures (see Appendix A). Parameters have been defined using "*defvar*", and are bound using "*setq*" in rules and functional procedures. Some parameters are additionally given property values, to expand them to vectors. Some parameters have been grouped to aid variable binding. With hindsight, this has worked well for internal parameters that are given values during search, but it is cumbersome for representing other parameters external to the system. Redefinition of external data, perhaps using "*defflavor*" and "*make-instance*", may have advantages and is being considered.

Functional LISP procedures retrieve rule elements, evaluate rules, conduct goal-directed searches, and bind variables. They are defined using "*defun*" and are called from a LISP Listener or from within the rule bases. The goal-directed search procedure "*get-value-of*" is currently the only search procedure being used, although other procedures have been written. "*Get-value-of*" is called with a goal parameter and a rule base as arguments. It searches the rule base for rules that have a "flag" indicating they can set the value of the goal parameter and tries them usingthe procedure "tryrule". "*Get-value-of*" is called to begin execution of the expert system and is called frequently within the rule base, resulting in backward-chaining. The development of LISP procedures is essentially complete; however, FORTRAN procedures for simulating, estimating, and controlling aircraft and systems must be implemented. The Symbolics LISP machine permits interaction of LISP and FORTRAN procedures, making the demonstration of FAA Windshear Training Aid procedures and other algorithms possible.

The expert system may be called by a LISP Listener or by a FORTRAN program; the former method has been used up until now. The expert system presently sends documentation to a LISP Listener while it is operating. It prompts the user for inputs through a LISP Listener; the user can give additional inputs by temporarily halting execution of the expert system. This interim arrangement has been adequate for development thus far, but it does not adequately demonstrate the interaction of the expert system with its operating environment.

To adequately demonstrate the system and aid further development, a simulation environment must be implemented that includes aircraft, on-board systems, ground-based information sources, and "mother nature". Modelling events outside of the system adds autonomy and realism to the simulation. To better understand the interaction of the expert system with its environment, a multi-window display must be developed that gives information about several aspects of the system and its environment simultaneously. The Symbolics machine has programs that are intended to streamline the development of multi-window displays; our intention is to quickly develop a simple means of presenting a variety of information.

2. Search Procedures for Wind Shear Expert System

The Wind Shear Expert System, in its current state, is principally a model of pilot and crew decision-making and control, as described in the FAA Windshear Training Aid. The four primary functions of the system are:

1. MONITORING -- Observe sensors, receive reports, alerts, warnings

2. ASSESSMENT -- Detect wind shear encounters, determine if there are signs of wind shear, and if it is safe to continue

3. PLANNING -- Determine what actions and precautions are taken

4. ACTION -- In automatic mode, execute standard, recovery, and go-around procedures; in semi-automatic mode, issue commands to flight directors

These functions are performed as side effects of a goal-directed search for parameter values in a set of rules. A rule, in the context of this report, is an IF-THEN statement; a list of the 141 rules implemented on a Symbolics 3670 LISP machine is enclosed (see Appendix B). A search procedure is simply a process of "trying" rules in a selective fashion. "Trying" a rule means that the premise of a rule is examined to determine its truth or falsehood, and the consequent is examined if the premise is true. Often the consequent is a direction to perform some action or procedure. Typically, the value of a parameter is set in the consequent. The search procedure is given a goal parameter, it finds rules that have consequents that set the value of the goal parameter, and it tries them sequentially, stopping when a rule is tried whose premise is true. If another parameter, whose value has not been determined, is encountered in a rule's premise, that parameter becomes the new goal parameter and a new search commences for the value of that parameter. If none of the rules capable of setting a goal parameter have premises that are true, the user is asked to give a value for the parameter in question. This enables the expert system to be tested with an incomplete rule base.

To demonstrate the actions of the search procedure, consider the following example. The expert system is invoked periodically from a simulation by directing the search procedure to get a value for the parameter "*RBC-search-complete*". Turning to the section "Parameters for Wind Shear Expert System", we find that "*RBC-search-complete*" is set in the EXECUTIVE rule base. Now turning to Appendix B, we find two rules that can set this parameter:

"IF the mission phase is not landed,
THEN we must complete executive procedures,
and the rule-based controller search is not complete.

IF the mission phase is landed,
THEN the rule-based controller search is complete."

Trying the first rule, we must consider the truth or falsehood of the expression, "the mission phase is not landed". To know this, we must find out what the mission phase is. Turning to the list of parameters, we find a parameter "*mission-phase*" that is set in the MISSION PHASE rule base. The rules within the MISSION PHASE rule base are tried next. When the mission phase has been determined, we return to considering the truth or falsehood of the first premise. If, for example, we determined that the mission phase was not "*landed*", we would then consider the consequent of the first rule, "we must complete executive procedures, and the rule-based controller search is not complete." The first part of the consequent is actually a call to the search procedure (rules were translated into English to facilitate readability; regrettably, this can make it more difficult to follow the search process), to get a value for the parameter "*executive-search-complete*". Once this search is complete, the parameter "*RBC-search-complete*" is set to NIL and the top-level search is finished.

3. Plans for Improvement of Wind Shear Expert System

The Wind Shear Expert System has developed primarily from the top level downward, branching out as the number of rules increases. This approach, one of the advantages of the expert system paradigm for control system development, tends to leave open-ended parameters on the lowest level. The material enclosed does not represent a complete rule-based controller and should be viewed as an interim development. Currently, there are a few parameters that do not have any rules or procedures to give them values. Additionally, there are procedures called by the expert system that are not yet implemented as software. Briefly, a list of major items still needing attention includes:

- Simulation environment
- Improved user interface
- Improved relevance information and testing
- Control system architecture
- Target trajectory generation
- Prediction procedures for aircraft and weather
- Rules to interface with new procedures

Addition of the above items should complete an expert system model of the FAA Windshear Training Aid and provide the basis for further development of a rule-based pilot aid for wind shear survival.

Appendix A

Parameters for Wind Shear Expert System

Below is a list of the current parameters contained in the wind shear expert system. Parameters have been classified according to how they are given values. These classifications are:

INTERNAL (I) -	Parameter is internal to the expert system and gets its value from a goal-directed search procedure.
PRESET (Pr) -	Parameter is set by flight crew, through the pilot interface.
STATE VARIABLE (S) -	Parameter is a state estimate, set by a state estimator.
OUTSIDE (O) -	Parameter's value is given to the program by an outside interface.

1. Internal Parameters

PARAMETER NAME	TYPE	VALUES (INTERPRETED)
EXECUTIVE RULEBASE sets:		
RBC-search-complete	I	T (we should keep searching) NIL (we should stop searching)
executive-procedures-complete	I	T (procedures are not complete)

MISSION PHASE RULEBASE sets:

mission-phase	I	PREFLIGHT-DECISION stage TAKEOFF (pilot committed) CLIMBOUT (off the ground) CRUISE (takeoff completed) PRELANDING-DECISION stage APPROACH (pilot committed) FLARE (below flare height) LANDING (on runway) LANDED (operations completed)
current-airport	Pr/I	[any airport name] (nearest airport)
are-we-committed	Pr/I	T (flight crew committed to takeoff/land) NIL (flight crew not committed)

WINDSHEAR RULEBASE sets:

microburst-encounter	I	T (we are having a microburst encounter) NIL (no microburst encounter)

DETECTION RULEBASE sets:

detection-complete	I	T (search of detection rule base completed)

DEVIATION RULEBASE sets:

aircraft-response	I	[list] contains one or more of: MICROBURST (significant deviations): ABOVE-TARGET-AIRSPEED BELOW-TARGET-AIRSPEED ABOVE-TARGET-VERTICAL-SPEED BELOW-TARGET-VERTICAL-SPEED ABOVE-TARGET-PITCH-ATTITUDE BELOW-TARGET-PITCH-ATTITUDE ABOVE-TARGET-GLIDESLOPE-DEV. BELOW-TARGET-GLIDESLOPE-DEV. ABOVE-TARGET-THROTTLE-POS. BELOW-TARGET-THROTTLE-POS.
target-airspeed	I	[numerical] (knots)
airspeed-deviation	I	[numerical] (deviation from target, knots)
target-vertical-speed	I	[numerical] (ft/min)
vertical-speed-deviation	I	[numerical] (deviation from target, ft/min)
target-pitch-attitude	I	[numerical] (deg)
attitude-deviation	I	[numerical] (deviation from target, deg)
target-glideslope-displacement	I	[numerical] (dots)

glideslope-deviation	I	[numerical] (deviation from target, dots)
throttle-deviation	I	[numerical] (deviation from target, %)

COMMUNICATION RULEBASE sets:

new-information-received	I	T (new information is received) NIL (no new info received)
incident-reported	I	T (if there was an incident, it's reported)
tower-informed-goa	I	T (tower informed of intent to go-around) NIL (tower not yet informed of intent)
tower-informed-delay	I	T (tower informed of intent to delay) NIL (tower not yet informed)

ACTION RULEBASE sets:

actions-taken	I	T (actions have been taken)

STANDARD PROCEDURES RULEBASE sets:

standard-procedures-complete	I	T (standard procedures are now complete)
configuration-set-for-takeoff	I	T (configuration is set for takeoff) NIL (configuration not set for takeoff)
configuration-set-for-landing	I	T (configuration is set for landing) NIL (configuration is not set for landing)

glideslope-established	Pr/I	T (glideslope has been established) NIL (glideslope is not yet established)

RECOVERY PROCEDURES RULEBASE sets:

recovery-procedures-complete	I	T (recovery procedures are now complete)

GO-AROUND PROCEDURES RULEBASE sets:

go-around-procedures-complete	I	T (go-around procedures are complete)

DELAY PROCEDURES RULEBASE sets:

delay-procedures-complete	I	T (delay procedures are complete)

PLANNING RULEBASE sets:

recommended-procedures	I	T (mission plan is now updated)

RUNWAY SELECTION RULEBASE sets:

recommended-runway	I	[runway id] (new runway selection)

SPEED SELECTION RULEBASE sets:

recommended-Vr	I	[numerical] (precautionary rotation speed)
recommended-Va	I	[numerical] (precautionary approach speed)

FLAP SELECTION RULEBASE sets:

recommended-flaps	I	[numerical] (precautionary flap setting)

ASSESSMENT RULEBASE sets:

recommended-action	I	CONTINUE operations as planned CONSIDER-PRECAUTIONS DELAY-ALTER-ABORT EXECUTE-RECOVERY procedures

RISK RULEBASE sets:

risk	I	[numerical] (overall risk factor for wind shear - values correspond to: 0 - no probability of wind shear 1 - low probability of wind shear 2 - medium probability of shear 3 or greater - high probability

WEATHER RISK RULEBASE sets:

weather	I	[numerical] (risk of wind shear derived from weather reports,e.g., SIGMETs, same severity scale as "risk")
suspected-runways	I	[list of runway id's] (runways reported or observed to have conditions suggesting wind shear)

LOCALIZED WINDS RULEBASE sets:

localized-strong-winds	I	[numerical] values correspond to: 0 - no indications of wind shear 1 - indications, off the flight path 2 - indications, convecting onto path 3 - indications of wind shear on path

HEAVY PRECIPITATION RULEBASE sets:

heavy-precipitation	I	[numerical] (same scale as localized-strong winds)

RAINSHOWER RULEBASE sets:

rainshower	I	[numerical] (same scale as localized-strong winds)

LIGHTNING RULEBASE sets:

lightning	I	[numerical] (same scale as localized-strong winds)

VIRGA RULEBASE sets:

virga	I	[numerical] (same scale as localized-strong winds)

TURBULENCE RULEBASE sets:

turbulence	I	[numerical] (moderate or greater - same scale as localized-strong-winds)

2. Preset Parameters

origin-airport	Pr	[any airport name] (origin airport)
taxi-speed-limit	Pr	[numerical] (taxi speed limit, knots)
destination-airport	Pr	[any airport name] (next destination)
flare-agl	Pr	[numerical] (flare altitude, ft)
Vr	Pr	[numerical] (standard rotation speed for this configuration, knots)
Va	Pr	[numerical] (standard approach speed for this configuration, knots)
critical-agl	Pr	[numerical] (height below which ground contact is major concern, ft)
aircraft-type	Pr	[any type in PWG] (type of aircraft)
V1	Pr	[numerical] (abort groundspeed, knots)
suitable-runways	Pr	[list of runways] (suitable runways)
Vr-field-length-limit	Pr	[numerical] (field length limit rotation speed for this configuration, knots)

3. State Variables

agl	S	[numerical] (alt. above ground, ft)

airspeed	S	[numerical] (current airspeed, ft)
vertical-speed	S	[numerical] (ft/min)
pitch-attitude	S	[numerical] (deg)
glideslope-displacement	S	[numerical] (dots)
groundspeed	S	[numerical] (groundspeed, knots)
runway-remaining	S	[numerical] (current est. of available runway, ft)

4. Outside Parameters

wind shear-alert	O	T (on-board wind shear detection sys. alert) NIL (no on-board wind shear det. sys. alert)
traffic	O	T (threatening traffic ahead) NIL (no threatening traffic ahead)
pirep*	O	[numerical] (amount of speed lost/gained by some other flight crew, knots)
llwas*	O	[numerical] (amount of speed differential that triggered some llwas alert, knots)
forecast*	O	[list] (forecast of convective weather)
temp/dewpt-spread	O	[numerical] (degrees F)

ATIS*	O	TURBULENCE noted in report RAINSHOWERS noted in report LIGHTNING noted in report VIRGA noted in report HEAVY-PRECIPITATION noted LLWS (low-level wind shear noted)
ASWW*	O	TURBULENCE noted in report RAINSHOWERS noted in report LIGHTNING noted in report VIRGA noted in report HEAVY-PRECIPITATION noted LLWS (low-level wind shear noted)
SIGMET*	O	TURBULENCE noted in report RAINSHOWERS noted in report LIGHTNING noted in report VIRGA noted in report HEAVY-PRECIPITATION noted LLWS (low-level wind shear noted)
onboard-radar*	O	TURBULENCE observed HEAVY-PRECIPITATION observed
tower-report*	O	TURBULENCE noted by tower RAINSHOWERS noted by tower LIGHTNING noted by tower VIRGA noted by tower HEAVY-PRECIPITATION noted WINDSHEAR noted by tower
TDWR*	O	TURBULENCE noted by TDWR HEAVY-PRECIPITATION noted WINDSHEAR noted by TDWR
wind-profiler*	O	TURBULENCE detected by profiler WINDSHEAR detected by profiler

forward-looking*	O	TURBULENCE detected RAINSHOWERS detected LIGHTNING detected VIRGA detected HEAVY-PRECIPITATION detected WINDSHEAR detected
observation	O	[numerical] (phenomenon seen by flight crew on flight path, values correspond to: 0 - no suspicious phenomenon seen 1 - localized strong winds seen 2 - heavy precipitation seen 3 - rain showers seen 4 - lightning seen 5 - virga seen
LSS*	O	LIGHTNING detected by lightning sensor

* The parameter also has properties indicating airports and runways to which it pertains.

Appendix B

Rule Bases for Wind Shear Expert System

EXECUTIVE RULEBASE

IF the mission phase is not landed,
THEN we must complete executive procedures,
and the rule-based controller search is not complete.

IF the mission phase is landed,
THEN the rule-based controller search is complete.

IF the mission phase is preflight decision stage or prelanding decision stage,
THEN we must determine recommended procedures,
and executive procedures are complete.

IF the mission phase is takeoff,
or the mission phase is climbout,
or the mission phase is approach,
or the mission phase is flare,
or the mission phase is landing rollout,
THEN we must take control actions,
and executive procedures are complete.

IF the mission phase is takeoff complete,
THEN the current airport is now the destination airport,
and the pilot is not committed to land,
and executive procedures are complete.

MISSION PHASE RULEBASE

IF the current airport is the origin airport,
and our altitude(ft) is 0 feet,
and the pilot is not committed to takeoff,
THEN the mission phase is preflight decision stage.

IF the current airport is the origin airport,
    and our altitude (ft) is 0 feet,
    and the pilot is committed to takeoff,
THEN the mission phase is takeoff.

IF the current airport is the origin airport,
    and our altitude (ft) is 0 feet,
    and our airspeed (knots) is greater than the taxi speed limit (knots),
    and the flight crew has not committed to takeoff,
THEN the flight crew has passively committed to takeoff,
    and the mission phase is takeoff.

IF the current airport is the origin airport,
    and our altitude (ft) is greater than 0 feet,
    and our altitude (ft) is less than 1000 feet,
THEN the mission phase is climbout.

IF the current airport is the origin airport,
    and our altitude (ft) is greater than 1000 feet,
THEN the mission phase is takeoff complete.

IF the current airport is the destination airport,
    and our altitude (ft) is greater than 1000 feet,
    and the flight crew is not committed to land,
THEN the mission phase is prelanding decision stage.

IF the current airport is the destination airport,
    and our altitude (ft) is below 1000 feet.,
    and we are not flight crew is not committed to land,
THEN the flight crew has passively committed to land,
    and the mission phase is approach.

IF the current airport is the destination airport,
    and our altitude (ft) is less than 1000 feet,
    and our altitude (ft) is above flare altitude (ft),
    and the flight crew is committed to land,
THEN the mission phase is approach.

IF the current airport is the destination airport,
    and our altitude (ft) is below flare altitude (ft),
    and the flight crew is committed to land,
THEN the mission phase is flare.

IF the current airport is the destination airport,
and our altitude (ft) is 0 feet,
and our airspeed (knots) is above the taxi speed limit (knots),
THEN the mission phase is landing roll.

IF the current airport is the destination airport,
and our altitude (ft) is 0 feet,
and our airspeed (knots) is below the taxi speed limit (knots),
THEN the mission phase is landed.

WINDSHEAR RULEBASE

IF the search of the detection rulebase is complete,
and the aircraft trajectory is deviating significantly from the target trajectory,
THEN there is a microburst encounter.

IF a wind shear alert is sounded by an on-board wind shear detection system,
THEN there is a microburst encounter.

IF a microburst encounter has already begun,
and the aircraft trajectory is deviating somewhat from the target trajectory,
THEN there is a microburst encounter.

IF a microburst encounter has not begun,
and the aircraft trajectory is not deviating significantly from the target trajectory,
THEN there is no microburst encounter.

DETECTION RULEBASE

IF the mission phase is approach or prelanding decision stage,
THEN airspeed deviations must be monitored,
and vertical speed deviations must be monitored,
and pitch attitude deviations must be monitored,
and glideslope displacement deviations must be monitored,
and throttle position deviations must be monitored,
and the search of the detection rulebase is complete.

IF the mission phase is neither approach nor prelanding decision stage,
THEN airspeed deviations must be monitored,
and vertical speed deviations must be monitored,
and pitch attitude deviations must be monitored
and the search of the detection rulebase is complete.

DEVIATION RULEBASE

IF airspeed (knots) is below target airspeed (knots) by more than 15 knots,
THEN the aircraft is deviating significantly from the target trajectory.

IF airspeed (knots) is above target airspeed (knots) by more than 15 knots,
THEN the aircraft is deviating significantly from the target trajectory.

IF airspeed (knots) is below target airspeed (knots) by between 5 and 15 knots,
THEN the aircraft is deviating somewhat from the target trajectory,
and the type of deviation includes "below target airspeed".

IF airspeed (knots) is above target airspeed (knots) by between 5 and 15 knots,
THEN the aircraft is deviating somewhat from the target trajectory,
and the type of deviation includes "above target airspeed".

IF vertical speed (ft/min) is below target vertical speed (ft/min) by more than 500 ft/min,
THEN the aircraft is deviating significantly from the target trajectory.

IF vertical speed (ft/min) is above target vertical speed (ft/min) by more than 500 ft./min.,
THEN the aircraft is deviating significantly from the target trajectory.

IF vertical speed (ft/min) is below target vertical speed (ft/min) by between 200 and 500 ft./min.,
THEN the aircraft is deviating somewhat from the target trajectory,
and the type of deviation includes "below target vertical speed".

IF vertical speed (ft/min) is above target vertical speed (ft/min) by between 200 and 500 ft./min.,
THEN the aircraft is deviating somewhat from the target trajectory,
and the type of deviation includes "above target vertical speed".

IF pitch attitude (deg) is below target pitch attitude (deg) by more than 5 degrees,
THEN the aircraft is deviating significantly from the target trajectory.

IF pitch attitude (deg) is above target pitch attitude (deg) by more than 5 degrees,
THEN the aircraft is deviating significantly from the target trajectory.

IF pitch attitude (deg) is below target pitch attitude (deg) by between 2 and 5 degrees,
THEN the aircraft is deviating somewhat from the target trajectory, and the type of deviation includes "below target pitch attitude".

IF pitch attitude (deg) is above target pitch attitude (deg) by between 2 and 5 degrees,
THEN the aircraft is deviating somewhat from the target trajectory, and the type of deviation includes "above target pitch attitude".

IF glideslope displacement (dots) is below target glideslope displacement (dots) by more than 1 dot,
THEN the aircraft is deviating significantly from the target trajectory.

IF glideslope displacement (dots) is above target glideslope displacement (dots) by more than 1 dot,
THEN the aircraft is deviating significantly from the target trajectory.

IF glideslope displacement (dots) is below target glideslope displacement (dots) by between 0.4 and 1 dot,
THEN the aircraft is deviating somewhat from the target trajectory, and the type of deviation includes "below target glideslope displacement".

IF glideslope displacement (dots) is above target glideslope displacement (dots) by between 0.4 and 1 dot,
THEN the aircraft is deviating somewhat from the target trajectory, and the type of deviation includes "below target glideslope displacement".

COMMUNICATION RULEBASE

IF the outside interface indicates new information,
THEN new information has been received.

IF the outside interface does not indicate new information,
THEN no new information has been received.

IF there was a microburst encounter,
and there is no longer a microburst encounter,
THEN the incident must be recorded now.

IF the above premise is not true,
THEN there is no need to report an incident.

ACTION RULEBASE

IF we are having a microburst encounter,
and guidance mode is automatic,
THEN recovery procedures must be completed,
and control actions are taken.

IF we are not having a microburst encounter,
and guidance mode is automatic,
and recommended procedures have been determined,
and standard procedures are recommended,
THEN standard procedures must be completed,
and control actions are taken.

IF we are not having a microburst encounter,
and guidance mode is automatic,
and go-around procedures are recommended,
THEN go-around procedures must be completed,
and control actions are taken.

IF we are not having a microburst encounter,
and delaying procedures are recommended,
THEN delaying procedures must be completed,
and control actions are taken.

IF we are having a microburst encounter,
and guidance mode is semi-automatic,
THEN semi-automatic recovery procedures must be completed,
and control actions are taken.

IF we are not having a microburst encounter,
and guidance mode is semi-automatic,
and recommended procedures have been determined,
and standard procedures are recommended,
THEN semi-automatic standard procedures must be completed,
and control actions are taken.

IF we are not having a microburst encounter,
and guidance mode is semi-automatic,
and go-around procedures are recommended,
THEN semi-automatic go-around procedures must be completed,
and control actions are taken.

STANDARD PROCEDURES RULEBASE

IF the mission phase is takeoff,
and the configuration has not been set for the takeoff roll,
THEN the configuration must be set for the takeoff roll,
and standard procedures are complete.

IF the mission phase is takeoff,
and the airspeed (knots) is below Vr (knots),
THEN takeoff control laws must be executed,
and standard procedures are complete.

IF the mission phase is takeoff,
and the airspeed (knots) is above Vr (knots),
THEN rotation control laws must be executed,
and standard procedures are complete.

IF the mission phase is climbout,
THEN climbout control laws must be executed,
and standard procedures are complete.

IF the mission phase is approach,
and the configuration has not been set for landing,
THEN the configuration must be set for landing,
and standard procedures are complete.

IF the mission phase is approach,
and the glideslope has not been established,
THEN we must complete procedures to establish the glideslope,
and standard procedures are complete.

IF the mission phase is approach,
and the glideslope has been established,
THEN approach control laws must be executed,
and standard procedures are complete.

IF the mission phase is flare,
THEN flare control laws must be executed,
and standard procedures are complete.

IF the mission phase is landing roll,
THEN landing control laws must be executed,
and standard procedures are complete.

RECOVERY PROCEDURES RULEBASE

IF the altitude (ft) is above critical (ft),
and the aircraft type is an L-1011,
and the mission phase is takeoff,
THEN controls must be configured for a recovery,
and the desired thrust setting is maximum rated,
and the desired pitch attitude (deg) is 17.5 degrees,
and recovery procedures are complete.

IF the altitude is above critical,
and the aircraft is not an L-1011,
or the aircraft is an L-1011 and the mission phase is approach,
THEN controls must be configured for a recovery,
and the desired thrust setting is maximum rated,
and the desired pitch attitude (deg) is 15 degrees,
and recovery procedures are complete.

IF the altitude is below critical,
and the mission phase is not takeoff or landed,
THEN controls must be configured for a recovery,
and the desired thrust setting is overboost,
and the desired pitch attitude (deg) is 20 degrees,
and recovery procedures are complete.

IF the mission phase is takeoff,
and the groundspeed (knots) is less than V1 (knots),
THEN controls must be configured for takeoff abort,
and the desired thrust setting is reverse,
and recovery procedures are complete.

IF the mission phase is takeoff,
and the groundspeed (knots) is greater than V1 (knots),
and the airspeed (knots) is less than Vr (knots),
and the runway remaining (ft) is greater than 2000 feet,
THEN controls must be configured for takeoff,
and the desired thrust setting is maximum rated,
and recovery procedures are complete.

IF the mission phase is takeoff,
and the airspeed (knots) is greater than Vr (knots),
THEN controls must be configured for a recovery,
and the desired thrust setting is maximum rated,
and the desired pitch attitude (deg) is 15 degrees,
and recovery procedures are complete.

IF the mission phase is takeoff,
and the runway remaining (ft) is less than 2000 feet,
THEN controls must be configured for a recovery,
and the desired thrust setting is maximum rated,
and the desired pitch attitude (deg) is 15 degrees,
and recovery procedures are complete.

GO-AROUND PROCEDURES RULEBASE

IF the tower has not been informed of our intent to go-around,
and there is no traffic in our way,
THEN go-around control laws must be executed,
and the tower must be informed of our intent to go-around,
and go-around procedures are complete.

IF the tower has not been informed of our intent to go-around,
and there is traffic in our way,
and our altitude is above critical,
THEN approach control laws must be executed,
and the tower must be informed of our intent to go-around,
and go-around procedures are complete.

IF the tower has not been informed of our intent to go-around,
and there is traffic in our way,
and our altitude is below critical,
THEN go-around control laws must be executed,
and the tower must be informed of our intent to go-around,
and go-around procedures are complete.

IF the tower has been informed of our intent to go-around,
THEN go-around control laws are in effect,
and go-around procedures are complete.

DELAY PROCEDURES RULEBASE

IF the tower has not been informed of our intent to delay,
and our ETA at the airport (min) is less than 20 minutes,
THEN the tower must be informed of our intent to delay,
and delay procedures are complete.

IF the tower has been informed of our intent to delay,
THEN delay procedures are complete.

PLANNING RULEBASE

IF the recommended action is to delay, alter, or abort,
and the mission phase is preflight decision stage,
or the mission phase is prelanding decision stage,
THEN delaying procedures are recommended.

IF the recommended action is to delay, alter, or abort,
and the mission phase is takeoff,
or the mission phase is climbout,
or the mission phase is flare,
or the mission phase is landing roll,
THEN standard procedures are recommended.

IF the recommended action is to delay, alter, or abort,
and the mission phase is approach,
and the altitude (ft) is above decision height (ft),
THEN go-around procedures are recommended.

IF the recommended action is to delay, alter, or abort,
and the mission phase is approach,
and the altitude (ft) is below decision height (ft),
THEN standard procedures are recommended.

IF the recommended action is to take precautions,
and precautions have already been taken, or the flight crew has rejected precautions,
THEN standard procedures are recommended.

IF the recommended action is to take precautions,
and the mission phase is preflight decision stage,
THEN we must determine the best runway,
and we must determine the best flap setting (deg),
and we must determine the best Vr (knots),
and the precautions must be accepted or rejected by the flight crew.

IF the recommended action is to take precautions,
and the mission phase is prelanding decision stage,
THEN we must stabilize approach at 1000 feet,
and we must determine the best flap setting (deg),
and we must determine the best Va (knots),
and the precautions must be accepted or rejected by the flight crew.

IF the recommended action is to take precautions,
and the mission phase is not preflight or prelanding decision stage,
THEN standard procedures are recommended.

IF the recommended action is to continue,
and the mission phase is approach,
and altitude (ft) is below critical altitude (ft),
and the glideslope has not been established,
THEN go-around procedures are recommended.

IF the recommended action is to continue,
and the mission phase is approach,
and the altitude (ft) is above the decision altitude (ft),
and the weighted norm of the 2-sigma touchdown dispersion (ft) is greater than the dispersion tolerance (ft),
THEN go-around procedures are recommended.

IF the recommended action is to continue,
and the above premises are not true,
THEN standard procedures are recommended.

RUNWAY RULEBASE

IF some of the suitable runways are suspected of wind shear,
THEN the best runway is the longest of the suitable runways not suspected.

IF none of the suitable runways are suspected of wind shear,
THEN the best runway is the longest of the suitable runways.

SPEED RULEBASE

IF field-length-limit Vr (knots) is greater than the standard Vr (knots),
and is less than 20 knots greater than the standard Vr (knots),
THEN the best Vr (knots) is the field-length-limit Vr (knots).

IF field-length-limit Vr (knots) is more than 20 knots greater than the standard Vr (knots),
THEN the best Vr (knots) is 20 knots plus the standard Vr (knots).

IF this rule is fired,
THEN the best Va (knots) is 20 knots plus the standard Va (knots).

FLAP RULEBASE

IF the mission phase is preflight decision stage,
and the aircraft type is a B727,
THEN the best flap setting (deg) is 15 degrees.

(12 more rules like the above, for different A/C configurations)

ASSESSMENT RULEBASE

IF there is no new information received,
or new information has been received,
and all of the rules in the risk rulebase have been tried,
and the overall risk factor is 0 (none),
THEN the recommended action is to continue.

IF the overall risk factor is 3 or greater (high),
THEN the recommended action is to delay, alter, or abort.

IF the overall risk factor is 2 (medium),
THEN the recommended action is to consider precautions.

IF the overall risk factor is 1 (low),
THEN the recommended action is to continue.

RISK RULEBASE

IF this rule is fired,
THEN the overall risk factor is set to 0
and the list of causes is set to nil.

IF there is an indication of convective weather on our flight path,
THEN all of the rules in the weather risk rulebase must be tried,
and the weather risk factor is added to the overall risk factor,
and weather is added to the list of causes.

IF there is a pilot report of speed loss (knots) less than 15 knots (but nonzero),
and this report is relevant to our current flight path,
THEN the overall risk factor is increased by 2,
and the current runway is added to the list of suspected runways,
and the pirep is added to the list of causes.

IF there is a pilot report of speed loss (knots) greater than 15 knots,
and this report is relevant to our current flight path,
THEN the overall risk factor is increased by 3,
and the current runway is added to the list of suspected runways,
and the pirep is added to the list of causes.

IF there is a LLWAS alert with a speed differential (knots) less than 20 knots,
and this alert is relevant to our current flight path,
THEN the overall risk factor is increased by 2,
and the current runway is added to the list of suspected runways,
and the LLWAS alert is added to the list of causes.

IF there is a LLWAS alert with a speed differential (knots) greater than 20 knots,
and this alert is relevant to our current flight path,
THEN the overall risk factor is increased by 3,
and the current runway is added to the list of suspected runways,
and the LLWAS alert is added to the list of causes.

IF there is a pilot report of speed loss (knots) less than 15 knots (but nonzero),
and this report is near but not on our current flight path,
THEN the overall risk factor is increased by 1,
and the current runway is added to the list of suspected runways,
and the pirep is added to the list of causes.

IF there is a pilot report of speed loss (knots) greater than 15 knots,
and this report is near but not on our current flight path,
THEN the overall risk factor is increased by 2,
and the current runway is added to the list of suspected runways,
and the pirep is added to the list of causes.

IF there is a LLWAS alert with a speed differential (knots) less than 20 knots,
and this alert is near but not on our current flight path,
THEN the overall risk factor is increased by 1,
and the current runway is added to the list of suspected runways,
and the LLWAS alert is added to the list of causes.

IF there is a LLWAS alert with a speed differential (knots) greater than 20 knots,
and this alert is near but not on our current flight path,
THEN the overall risk factor is increased by 2,
and the current runway is added to the list of suspected runways,
and the LLWAS alert is added to the list of causes.

IF there is a forecast of convective weather on our flight path,
THEN the overall risk factor is increased by 1,
and forecast is added to the list of causes.

WEATHER RISK RULEBASE

IF this rule is fired,
THEN the weather risk factor is reset to zero.

IF there are indications of localized strong winds on our flight path,
THEN the weather risk factor is increased by 3,
and the affected runways are added to the list of suspected runways.

IF there are indications of heavy precipitation on our flight path,
THEN the weather risk factor is increased by 3,
and the affected runways are added to the list of suspected runways.

IF there are indications of a rain shower on our flight path,
THEN the weather risk factor is increased by 2,
and the affected runways are added to the list of suspected runways.

IF there are indications of lightning on our flight path,
THEN the weather risk factor is increased by 2,
and the affected runways are added to the list of suspected runways.

IF there are indications of virga on our flight path,
THEN the weather risk factor is increased by 2,
and the affected runways are added to the list of suspected runways.

IF there are indications of moderate or greater turbulence on our flight path,
THEN the weather risk factor is increased by 2,
and the affected runways are added to the list of suspected runways.

IF the ATIS report indicates a temperature/dew point spread (^{0} F) greater than 30 ^{0}F,
THEN the weather risk factor is increased by 2,
and the affected runways are added to the list of suspected runways.

IF there are indications of localized strong winds near but not on our flight path,
THEN the weather risk factor is increased by 2,
and the affected runways are added to the list of suspected runways.

IF there are indications of heavy precipitation near but not on our flight path,
THEN the weather risk factor is increased by 2,
and the affected runways are added to the list of suspected runways.

IF there are indications of rain showers near but not on our flight path,
THEN the weather risk factor is increased by 1,
and the affected runways are added to the list of suspected runways.

IF there are indications of lightning near but not on our flight path,
THEN the weather risk factor is increased by 1,
and the affected runways are added to the list of suspected runways.

IF there are indications of virga near but not on our flight path,
THEN the weather risk factor is increased by 1,
and the affected runways are added to the list of suspected runways.

IF there are indications of moderate or greater turbulence near our flight path,
THEN the weather risk factor is increased by 1,
and the affected runways are added to the list of suspected runways.

LOCALIZED WINDS RULEBASE

IF ATIS information indicates low-level wind shear,
and the wind shear is on our flight path,
or an Aviation Severe Weather Watch indicates low-level wind shear,
and the wind shear is on our flight path,
or a SIGMET indicates low-level wind shear,
and the wind shear is on our flight path,
or a tower report indicates low-level wind shear,
and the wind shear is on our flight path,
or a Terminal Doppler Weather Radar indicates low-level wind shear,
and the wind shear is on our flight path,

or a wind profiler indicates low-level wind shear,
and the wind shear is on our flight path,
or a forward-looking system indicates low-level wind shear,
and the wind shear is on our flight path,
or a flight crew observation indicates low-level wind shear on our flight path,
THEN there are indications of localized strong winds on our flight path.

IF ATIS information indicates low-level wind shear,
and the wind shear is near but not on on our flight path,
or an Aviation Severe Weather Watch indicates low-level wind shear,
and the wind shear is near but not on our flight path,
or a SIGMET indicates low-level wind shear,
and the wind shear is near but not on our flight path,
or a tower report indicates low-level wind shear,
and the wind shear is near but not on our flight path,
or a Terminal Doppler Weather Radar indicates low-level wind shear,
and the wind shear is near but not on our flight path,
or a wind profiler indicates low-level wind shear,
and the wind shear is near but not on our flight path,
or a forward-looking system indicates low-level wind shear,
and the wind shear is near but not on our flight path,
or a flight crew observation indicates low-level wind shear near our flight path,
THEN there are indications of localized strong winds near our flight path.

HEAVY PRECIPITATION RULEBASE

IF ATIS information indicates heavy precipitation,
and the heavy precipitation is on our flight path,
or an Aviation Severe Weather Watch indicates heavy precipitation,
and the heavy precipitation is on our flight path,
or a SIGMET indicates heavy precipitation,
and the heavy precipitation is on our flight path,
or the on-board radar indicates heavy precipitation,
and the heavy precipitation is on our flight path,
or a tower report indicates heavy precipitation,
and the heavy precipitation is on our flight path,
or a Terminal Doppler Weather Radar indicates heavy precipitation,
and the heavy precipitation is on our flight path,
or a forward-looking system indicates heavy precipitation,
and the heavy precipitation is on our flight path,
or a flight crew observation indicates heavy precipitation on our flight path,
THEN there are indications of heavy precipitation on our flight path.

IF ATIS information indicates heavy precipitation,
and the heavy precipitation is near but not on on our flight path,
or an Aviation Severe Weather Watch indicates heavy precipitation,
and the heavy precipitation is near but not on our flight path,
or a SIGMET indicates heavy precipitation,
and the heavy precipitation is near but not on our flight path,
or the on-board radar indicates heavy precipitation,
and the heavy precipitation is near but not on our flight path,
or a tower report indicates heavy precipitation,
and the heavy precipitation is near but not on our flight path,
or a Terminal Doppler Weather Radar indicates heavy precipitation,
and the heavy precipitation is near but not on our flight path,
or a forward-looking system indicates heavy precipitation,
and the heavy precipitation is near but not on our flight path,
or a flight crew observation indicates heavy precipitation near our flight path,
THEN there are indications of heavy precipitation near our flight path.

RAINSHOWER RULEBASE

IF ATIS information indicates a rain shower,
and the rain shower is on our flight path,
or an Aviation Severe Weather Watch indicates a rain shower,
and the rain shower is on our flight path,
or a SIGMET indicates a rain shower,
and the rain shower is on our flight path,
or the on-board radar indicates a rain shower,
and the rain shower is on our flight path,
or a tower report indicates a rain shower,
and the rain shower is on our flight path,
or a Terminal Doppler Weather Radar indicates a rain shower,
and the rain shower is on our flight path,
or a forward-looking system indicates a rain shower,
and the rain shower is on our flight path,
or a flight crew observation indicates a rain shower on our flight path,
THEN there are indications of a rain shower on our flight path.

IF ATIS information indicates a rain shower,
and the rain shower is near but not on on our flight path,
or an Aviation Severe Weather Watch indicates a rain shower,
and the rain shower is near but not on our flight path,
or a SIGMET indicates a rain shower,

and the rain shower is near but not on our flight path,
or the on-board radar indicates a rain shower,
and the rain shower is near but not on our flight path,
or a tower report indicates a rain shower,
and the rain shower is near but not on our flight path,
or a Terminal Doppler Weather Radar indicates a rain shower,
and the rain shower is near but not on our flight path,
or a forward-looking system indicates a rain shower,
and the rain shower is near but not on our flight path,
or a flight crew observation indicates a rain shower near our flight path,
THEN there are indications of a rain shower near our flight path.

LIGHTNING RULEBASE

IF ATIS information indicates lightning,
and the lightning is on our flight path,
or an Aviation Severe Weather Watch indicates lightning,
and the lightning is on our flight path,
or a SIGMET indicates lightning,
and the lightning is on our flight path,
or the Lightning Sensor System indicates lightning,
and the lightning is on our flight path,
or a tower report indicates lightning,
and the lightning is on our flight path,
or a forward-looking system indicates lightning,
and the lightning is on our flight path,
or a flight crew observation indicates lightning on our flight path,
THEN there are indications of lightning on our flight path.

IF ATIS information indicates lightning,
and the lightning is near but not on on our flight path,
or an Aviation Severe Weather Watch indicates lightning,
and the lightning is near but not on our flight path,
or a SIGMET indicates lightning,
and the lightning is near but not on our flight path,
or the Lightning Sensor System indicates lightning,
and the lightning is near but not on our flight path,
or a tower report indicates lightning,
and the lightning is near but not on our flight path,
or a forward-looking system indicates lightning,
and the lightning is near but not on our flight path,
or a flight crew observation indicates lightning near our flight path,

THEN there are indications of lightning near our flight path.

VIRGA RULEBASE

IF ATIS information indicates virga,
and the virga is on our flight path,
or an Aviation Severe Weather Watch indicates virga,
and the virga is on our flight path,
or a SIGMET indicates virga,
and the virga is on our flight path,
or a tower report indicates virga,
and the virga is on our flight path,
or a forward-looking system indicates virga,
and the virga is on our flight path,
or a flight crew observation indicates virga on our flight path,
THEN there are indications of virga on our flight path.

IF ATIS information indicates virga,
and the virga is near but not on on our flight path,
or an Aviation Severe Weather Watch indicates virga,
and the virga is near but not on our flight path,
or a SIGMET indicates virga,
and the virga is near but not on our flight path,
or a tower report indicates virga,
and the virga is near but not on our flight path,
or a forward-looking system indicates virga,
and the virga is near but not on our flight path,
or a flight crew observation indicates virga near our flight path,
THEN there are indications of virga near our flight path.

TURBULENCE RULEBASE

IF ATIS information indicates moderate or greater turbulence,
and the turbulence is on our flight path,
or an Aviation Severe Weather Watch indicates moderate or greater turbulence,
and the turbulence is on our flight path,
or a SIGMET indicates moderate or greater turbulence,
and the turbulence is on our flight path,
or the on-board radar indicates turbulence,
and the turbulence is on our flight path,
or a tower report indicates turbulence,

and the turbulence is on our flight path,
or a Terminal Doppler Weather Radar indicates moderate or greater turbulence,
and the turbulence is on our flight path,
or a wind profiler indicates moderate or greater turbulence,
and the turbulence is on our flight path,
or a forward-looking system indicates moderate or greater turbulence,
and the turbulence is on our flight path,
or a flight crew observation indicates moderate or greater turbulence on our flight path,
THEN there are indications of moderate or greater turbulence on our flight path.

IF ATIS information indicates moderate or greater turbulence,
and the turbulence is near but not on on our flight path,
or an Aviation Severe Weather Watch indicates moderate or greater turbulence,
and the turbulence is near but not on our flight path,
or a SIGMET indicates moderate or greater turbulence,
and the turbulence is near but not on our flight path,
or the on-board radar indicates turbulence,
and the turbulence is near but not on our flight path,
or a tower report indicates turbulence,
and the turbulence is near but not on our flight path,
or a Terminal Doppler Weather Radar indicates moderate or greater turbulence,
and the turbulence is near but not on our flight path,
or a wind profiler indicates moderate or greater turbulence,
and the turbulence is near but not on our flight path,
or a forward-looking system indicates moderate or greater turbulence,
and the turbulence is near but not on our flight path,
or a flight crew observation indicates turbulence near our flight path,
THEN there are indications of moderate or greater turbulence near our flight path.

Appendix C

LISP Code for Rule Base Executive Functions

```lisp
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER; Base: 10; -*-
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;                                                         ;;;;;
;;;;;                          EXEC                           ;;;;;
;;;;;        RULE BASE FOR WINDSHEAR RULE-BASED CONTROL       ;;;;;
;;;;;                                                         ;;;;;
;;;;;      Started by Alex Stratton on March 17, 1988         ;;;;;
;;;;;       Modified from EXEC4 starting July 6,1988          ;;;;;
;;;;;                                                         ;;;;;
;;;;;       This file contains rules for windshear RBC.       ;;;;;
;;;;;   simple search inference engine that determines        ;;;;;
;;;;;   values of variables.                                  ;;;;;
;;;;;                                                         ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;
;;; define variables
;;;
;;; Variable EXEC-VARIABLES contains a list of the variables defined
;;; in this file.
;;;
(defvar exec-variables nil)
(setq exec-variables '(executive-procedures-complete windshear-alert exec-rules mphase-rules))
;;;
;;; variable RBC-FUNCTIONS-COMPLETE is T when we are landed, nil when we aren't
;;;
(defvar rbc-functions-complete)
;;;
;;; variable EXECUTIVE-PROCEDURES-COMPLETE is set when the executive functions are complete.
;;;
(defvar executive-procedures-complete nil)
;;;
;;; variable ARE-WE-COMMITTED is set to T by the pilot
;;;
(defvar are-we-committed nil)
;;;
;;; variable MICROBURST-ENCOUNTER can have values t or nil
;;;
(defvar microburst-encounter nil)
;;;
;;; variable NEW-INFORMATION-RECEIVED can be t or nil
;;;
(defvar new-information-received nil)
;;;
;;; variable RECOMMENDED-ACTION can be continue, take-precautions,
;;; avoid-windshear and execute-recovery
;;;
(defvar recommended-action 'continue)
;;;
(defvar recommended-procedures)
;;;
;;; variable INCIDENT-REPORTED can have values t and nil
;;;
(defvar incident-reported nil)
;;;
;;;   Rules to use in testing hypotheses
;;;
;;;   New format is:
;;;     (rulename if-doc
;;;               if-part   if-vars
;;;               then-doc
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
;;;              then-part then-vars)
;;;
(defvar exec-rules nil)
(setq exec-rules '(
               (exec1 "EXECUTIVE: Terminal operations are not complete, "
                      (and (get-value-of 'mission-phase mphase-rules)
                           (neq mission-phase 'landed))          'mission-phase
                      "so we will complete executive procedures. "
                      (and (get-value-of 'executive-procedures-complete exec-rules)
                           (setq rbc-functions-complete nil))         'rbc-functions-complete)

               (exec2 "EXECUTIVE: We have completed terminal operations, "
                      (eq mission-phase 'landed)                  'mission-phase
                      "so we should stop searching."
                      (setq rbc-functions-complete t)                 'rbc-functions-complete)

               (exec3 "EXECUTIVE: We are approaching a decision point, and have not encountered a microburst, "
                      (or (eq mission-phase 'preflight-decision)
                          (and (eq mission-phase 'prelanding-decision)
                               (get-value-of 'microburst-encounter windshear-rules)
                               (neq microburst-detected t)))          'mission-phase
                      "so we must determine the recommended procedures."
                      (and (get-value-of 'recommended-procedures planning-rules)
                           (get-value-of 'incident-reported comm-rules)) 'executive-procedures-complete)

               (exec4 "EXECUTIVE: We are completing a terminal operation, "
                      (or (eq mission-phase 'takeoff)
                          (eq mission-phase 'climbout)

                          (eq mission-phase 'approach)
                          (eq mission-phase 'flare)
                          (eq mission-phase 'landing-rollout))        'mission-phase
                      "so we must take the proper actions to complete the operation."
                      (and (get-value-of 'actions-taken action-rules)
                           (get-value-of 'incident-reported comm-rules)) 'executive-procedures-complete)

               (exec5 "EXECUTIVE: The mission phase is takeoff complete, "
                      (eq mission-phase 'takeoff-complete)            'mission-phase
                      "so we must now consider landing operations."
                      (and (setq current-airport destination-airport)
                           (setq are-we-committed nil))               'executive-procedures-complete)
               ))
;;;;;;;;;;;;;;;;;;;; mphase-rules ;;;;;;;;;;;;;;;;;;;;;;;;;
;     These rules determine the mission-phase
;
(defvar mphase-rules)
(setq mphase-rules    '(
                  (mphase1     "          MONITOR: We are near the origin airport, on the ground, not committed to takeoff, "
                               (and (eq current-airport origin-airport)
                                    (eq agl 0)
                                    (eq are-we-committed nil))          '(current-airport origin-airport)
                               "          so the mission phase is preflight decision stage."
                               (setq mission-phase 'preflight-decision)       'mission-phase)

                  (mphase2     "          MONITOR: We are near the origin airport, on the ground, committed to takeoff, "
                               (and (eq current-airport origin-airport)
                                    (eq agl 0)
                                    (eq are-we-committed t))            '(current-airport origin-airport)
                               "          so the mission phase is takeoff. "
                               (setq mission-phase 'takeoff)                  'mission-phase)
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
(mphase3   "        MONITOR: We are near the origin airport, are above taxi speed limit, not committed, "
           (and (eq current-airport destination-airport)
                (> airspeed taxi-speed-limit)
                (eq are-we-committed nil))                '(current-airport destination-airport)
           "        so we are passively committed to takeoff, and the mission phase is takeoff."
           (and (setq are-we-committed t)
                (setq mission-phase 'takeoff))           'mission-phase)

(mphase4   "        MONITOR: We are near the origin airport, and in the air,"
           (and (eq current-airport origin-airport)
                (> agl 0)
                (< agl 1000))                             '(current-airport origin-airport)
           "        so the mission phase is climbout. "
           (setq mission-phase 'climbout)                 'mission-phase)

(mphase5   "        MONITOR: We are above 1000 feet AGL, "
           (and (eq current-airport origin-airport)
                (> agl 1000))                             '(current-airport origin-airport)
           "        so the mission-phase is takeoff complete."
           (and (setq mission-phase 'takeoff-complete))   'mission-phase)

(mphase6   "        MONITOR: We are above decision altitude, not committed to land, "
           (and (eq current-airport destination-airport)
                (> agl decision-agl)
                (eq are-we-committed nil))                '(current-airport destination-airport)
           "        so the mission-phase is the prelanding decision stage."
           (setq mission-phase 'prelanding-decision)      'mission-phase)

(mphase7   "        MONITOR: We have selected the destination, are below 1000 feet, not committed, "
           (and (eq current-airport destination-airport)
                (< agl 1000)
                (eq are-we-committed nil))                '(current-airport destination-airport)
           "        so we are passively committed to land, and the mission phase is approach."
           (and (setq are-we-committed t)
                (setq mission-phase 'approach))           'mission-phase)

(mphase8   "        MONITOR: We are near the destination airport, above flare height, committed to land, "
           (and (eq current-airport destination-airport)
                (> agl flare-agl)
                (eq are-we-committed t))                  '(current-airport origin-airport)
           "        so the mission phase is approach."
           (setq mission-phase 'approach)                 'mission-phase)

(mphase9   "        MONITOR: We are near the destination airport, below flare height, committed to land, "
           (and (eq current-airport destination-airport)
                (< agl flare-agl)
                (eq are-we-committed t))                  '(current-airport origin-airport)
           "        so the mission phase is flare."
           (setq mission-phase 'flare)                    'mission-phase)

(mphase10  "        MONITOR: We are near the destination airport, on the ground, above taxi speed limit, "
           (and (eq current-airport destination-airport)
                (> airspeed taxi-speed-limit)
                (eq agl 0))                               '(current-airport origin-airport)
           "        so the mission phase is landing rollout."
           (setq mission-phase 'landing-roll)            'mission-phase)

(mphase11  "        MONITOR: We are near the destination airport, on the ground, below taxi speed limit, "
           (and (eq current-airport destination-airport)
```

```
          (eq agl 0)
          (< airspeed taxi-speed-limit))          '(current-airport origin-airport)
     "          so the mission phase is landed."
     (setq mission-phase 'landed)                  'mission-phase)
))
```

ORIGINAL PAGE IS
OF POOR QUALITY

Appendix D

LISP Code for Rules of Rule-Based Controller

```lisp
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER; Base: 10; -*-
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;                                                           ;;;;;
;;;;;                            RULES                          ;;;;;
;;;;;                 RULE BASE FOR WINDSHEAR RBC               ;;;;;
;;;;;                                                           ;;;;;
;;;;;        Started by Alex Stratton on March 17, 1988         ;;;;;
;;;;;       Modified from RULES4 starting July 6, 1988          ;;;;;
;;;;;                                                           ;;;;;
;;;;;     This file contains rules for rule-based control.      ;;;;;
;;;;;     Currently, format is consistent with RBC-FUNCTIONS.   ;;;;;
;;;;;                                                           ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;
;;;;;;;;;;;;; variable definitions ::::::::::::::
;;;
;;; define variables
;;;
;;;
(defvar airspeed 150)
(defvar target-v1)
(defvar target-vr)
(defvar target-va)
(defvar vr-field-length-limit 180)
(defvar taxi-speed-limit 50)
;;;
(defvar runway-remaining 1000)
;;;
(defvar agl 300)
(defvar critical-agl 200)
(defvar decision-agl 200)
(defvar flare-agl 200)
;;
(defvar current-airport)
(defvar current-runway 2)
(defvar always-bother-the-pilot t)
;;;
(defvar aircraft-type)
(defvar recommended-flaps)
(defvar recommended-throttle-setting)
(defvar recommended-runway)
(defvar recommended-vr)
(defvar recommended-stabilized-approach-altitude)
(defvar recommended-va)
;;;
;;;
(defvar aircraft-response nil)
;;;
;;; variable WINDSHEAR-ALERT can have values t, nil, and warning.
;;;
(defvar windshear-alert nil)
;;;
;;; Boolean variable PIREP has properties speed-change, airport
;;;
(defvar pirep nil)
;;;
;;; Boolean variable TOWER-REPORT has properties winds
;;;
(defvar tower-report nil)
;;;
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
;;; Boolean variable LLWAS has properties speed, airport
;;;
(defvar llwas)
;;;
;;; Variable ONBOARD-RADAR can have values NIL, TURBULENCE, and HEAVY-PRECIPITATION
;;;
(defvar onboard-radar nil)
;;;
;;; Boolean variable OBSERVATION has properties skies, winds
;;;
(defvar observation nil)
;;;
;;; Boolean variable WEATHER-REPORT has properties turbulence, temp-sprd, dwpt-sprd
;;;
(defvar weather-report nil)
;;;
;;; Variable MISSION-PHASE can have values TAKE-OFF and APPROACH
;;;
(defvar mission-phase)
;;;
;;; Variable ORIGIN-AIRPORT
;;;
(defvar origin-airport nil)
;;;
;;; Variable DESTINATION-AIRPORT
;;;
(defvar destination-airport nil)
;;;
;;; Variable SUSPECTED-RUNWAYS
;;;
(defvar suspected-runways)
;;;
;;;   Rules to use in testing hypotheses
;;;
;;;   New format is:
;;;     (rulename if-doc
;;;               if-part   if-vars
;;;               then-doc
;;;               then-part then-vars)
;;;
;;;
;;;;;;;;;;;;;;;;;;;;;; windshear-rules ;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;     these rules determine whether we are in windshear
;;;         or not.
;;;
;;; variable EXTERNAL-WINDSHEAR-VARIABLES contains a list of the variables
;;; that are assumed to have values before the windshear-rules are tried
;;;
(defvar external-windshear-variables nil)
(setq external-windshear-variables '(aircraft-response system-status))
;;;
(defvar windshear-rules nil)
(setq windshear-rules '(
                        (windshear1  "          MONITOR: Aircraft trajectory is deviating significantly from target trajectory!  "
                                     (and (get-value-of 'detection-completed detection-rules)
                                          (member 'microburst aircraft-response))        '(aircraft-response system-status)
                                     "          ******* MICROBURST ENCOUNTERED  *********"
                                     (setq microburst-encounter t)          'microburst-encounter)

                        (windshear2  "          MONITOR: Onboard windshear detection system alert! "
```

ORIGINAL PAGE IS OF POOR QUALITY

```
                                    (and (ask-lisp-listener-about 'windshear-alert)
                                         (eq windshear-alert t))           'windshear-alert
                                    "          ******* MICROBURST ENCOUNTERED  *********"
                                    (setq microburst-encounter t)          'microburst-encounter)

                (windshear3  "            MONITOR: A microburst encounter has begun, and there are still deviations from targets, "
                                    (and (eq microburst-encounter t)
                                         (or (member 'above-target-airspeed aircraft-response)
                                             (member 'below-target-airspeed aircraft-response)
                                             (member 'above-target-vertical-speed aircraft-response)
                                             (member 'below-target-vertical-speed aircraft-response)
                                             (member 'above-target-attitude aircraft-response)
                                             (member 'below-target-attitude aircraft-response)
                                             (member 'above-target-glideslope-displacement aircraft-response)
                                             (member 'below-target-glideslope-displacement aircraft-response)
                                             (member 'above-target-throttle aircraft-response)
                                             (member 'below-target-throttle aircraft-response))) 'aircraft-response
                                    "          ******* MICROBURST ENCOUNTERED  ********"
                                    (setq microburst-encounter t)          'microburst-encounter)

                (windshear4  "            MONITOR: Aircraft trajectory is not deviating from target trajectory, "
                                    (and (not (member 'microburst aircraft-response))
                                         (neq microburst-encounter t))       'aircraft-response
                                    "          so there is no microburst encounter at this time."
                                    (setq microburst-encounter nil)        'microburst-encounter)
                ))
;;;
;;;;;;;;;;;;;;;;;;;;;; detection-rules ;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;    these rules direct a search for response deviations
;;;
(defvar detection-rules)
(setq detection-rules '(
                (det1   "              MONITOR: The mission phase is approach or prelanding decision stage, "
                          (or (eq mission-phase 'approach)
                              (eq mission-phase 'prelanding-decision))    'mission-phase
                          "              so we will compute deviations from all target values."
                          (and (get-value-of 'airspeed-deviation deviation-rules)
                               (get-value-of 'vertical-speed-deviation deviation-rules)
                               (get-value-of 'attitude-deviation deviation-rules)
                               (get-value-of 'glideslope-deviation deviation-rules)
                          ;     (get-value-of 'throttle-deviation deviation-rules)
                               ) 'detection-completed)

                (det2   "              MONITOR: The mission phase is not approach or prelanding decision stage, "
                          (and (neq mission-phase 'prelanding-decision)
                               (neq mission-phase 'approach))          'mission-phase
                          "              so we will compute deviations from target speeds and attitude only."
                          (and (get-value-of 'airspeed-deviation deviation-rules)
                               (get-value-of 'vertical-speed-deviation deviation-rules)
                               (get-value-of 'attitude-deviation deviation-rules)) 'detection-completed)
                ))
;;;
;;;;;;;;;;;;;;;;;;;;;; deviation-rules ;;;;;;;;;;;;;;;;;;;;;;;;;
;;;    these rules direct detection of abnormal aircraft response on an established approach
;;;
(defvar deviation-rules)
(setq deviation-rules '(
                (dev1   "                   MONITOR: Airspeed is below target by over 15 knots, "
                          (and (> target-airspeed airspeed)
                               (> (- target-airspeed airspeed) 15))   'target-airspeed
```

```
             "                   and this is a strong indication of a microburst encounter."
             (setq aircraft-response '(microburst))  'airspeed-deviation)

     (dev2   "                   MONITOR: Airspeed is above target by over 15 knots, "
             (and (> airspeed target-airspeed)
                  (> (- airspeed target-airspeed) 15))  'target-airspeed
             "                   and this is a strong indication of a microburst encounter."
             (setq aircraft-response '(microburst))  'airspeed-deviation)

     (dev3   "                   MONITOR: Airspeed is below target between 5 and 15 knots, "
             (and (> target-airspeed airspeed)
                  (< (- target-airspeed airspeed) 15)
                  (> (- target-airspeed airspeed) 5))   'target-airspeed
             "                   and this is an indication of a windshear encounter."
             (setq aircraft-response '(above-target-airspeed))  'airspeed-deviation)

     (dev4   "                   MONITOR: Airspeed is above target between 5 and 15 knots, "
             (and (> airspeed target-airspeed)
                  (< (- airspeed target-airspeed) 15)
                  (> (- airspeed target-airspeed) 5))      'target-airspeed
             "                   and this is an indication of a windshear encounter."
             (setq aircraft-response '(above-target-airspeed))  'airspeed-deviation)

     (dev5   "                   MONITOR: Vertical speed is under target by over 500 ft/min, "
             (and (> target-vertical-speed vertical-speed)
                  (> (- target-vertical-speed vertical-speed) 500))   'target-vertical-speed
             "                   and this is an indication of a microburst encounter."
             (setq aircraft-response (cons 'microburst aircraft-response)) 'vertical-speed-deviation)

     (dev6   "                   MONITOR: Vertical speed is over target by over 500 ft/min, "
             (and (> vertical-speed target-vertical-speed)
                  (> (- vertical-speed target-vertical-speed) 500))   'target-vertical-speed
             "                   and this is an indication of a microburst encounter."
             (setq aircraft-response (cons 'microburst aircraft-response)) 'vertical-speed-deviation)

     (dev7   "                   MONITOR: Vertical speed is under target by between 200 and 500 ft/min, "
             (and (> target-vertical-speed vertical-speed)
                  (< (- target-vertical-speed vertical-speed) 500)
                  (> (- target-vertical-speed vertical-speed) 200))   'target-vertical-speed
             "                   and this is an indication of a windshear encounter."
           (setq aircraft-response (cons 'below-target-vertical-speed aircraft-response)) 'vertical-speed-deviation)

     (dev8   "                   MONITOR: Vertical speed is over target by betweeen 200 and 500 ft/min, "
             (and (> vertical-speed target-vertical-speed)
                  (< (- vertical-speed target-vertical-speed) 500)
                  (> (- vertical-speed target-vertical-speed) 200))   'target-vertical-speed
             "                   and this is an indication of a windshear encounter."
           (setq aircraft-response (cons 'above-target-vertical-speed aircraft-response)) 'vertical-speed-deviation)

     (dev9   "                   MONITOR: Pitch attitude is under target by over 5 degrees, "
             (and (> target-pitch-attitude pitch-attitude)
                  (> (- target-vertical-speed vertical-speed) 5))   'target-pitch-attitude
             "                   and this is an indication of a microburst encounter."
             (setq aircraft-response (cons 'microburst aircraft-response)) 'attitude-deviation)

     (dev10  "                   MONITOR: Pitch attitude is over target by over 5 degrees, "
             (and (> pitch-attitude target-pitch-attitude)
                  (> (- pitch-attitude target-pitch-attitude) 5))   'target-pitch-attitude
             "                   and this is an indication of a microburst encounter."
             (setq aircraft-response (cons 'microburst aircraft-response)) 'attitude-deviation)
```

```
(dev11  "                    MONITOR: Pitch attitude is under target by between 2 and 5 degrees, "
        (and (> target-pitch-attitude pitch-attitude)
             (< (- target-pitch-attitude pitch-attitude) 5)
             (> (- target-pitch-attitude pitch-attitude) 2))   'target-pitch-attitude
        "                    and this is an indication of a windshear encounter."
        (setq aircraft-response (cons 'below-target-pitch-attitude aircraft-response)) 'attitude-deviation)

(dev12  "                    MONITOR: Pitch attitude is over target by betweeen 2 and 5 degrees, "
        (and (> pitch-attitude target-pitch-attitude)
             (< (- pitch-attitude target-pitch-attitude) 5)
             (> (- pitch-attitude target-pitch-attitude) 2))   'target-pitch-attitude
        "                    and this is an indication of a windshear encounter."
        (setq aircraft-response (cons 'above-target-pitch-attitude aircraft-response)) 'attitude-deviation)

(dev13  "                    MONITOR: Glideslope displacement is under target by over 1 dot, "
        (and (> target-glideslope-displacement glideslope-displacement)
             (> (- target-glideslope-displacement glideslope-displacement) 1)) 'target-glideslope-displacement
        "                    and this is an indication of a microburst encounter."
        (setq aircraft-response (cons 'microburst aircraft-response)) 'glideslope-deviation)

(dev14  "                    MONITOR: Glideslope displacement is over target by over 1 dot, "
        (and (> glideslope-displacement target-glideslope-displacement)
             (> (- glideslope-displacement target-glideslope-displacement) 1)) 'target-glideslope-displacement
        "                    and this is an indication of a microburst encounter."
        (setq aircraft-response (cons 'microburst aircraft-response)) 'glideslope-deviation)

(dev15  "                    MONITOR: Glideslope displacement is under target by between .4 and 1 dot, "
        (and (> target-glideslope-displacement glideslope-displacement)
             (< (- target-glideslope-displacement glideslope-displacement) 1)
             (> (- target-glideslope-displacement glideslope-displacement) .4)) 'target-glideslope-displacement
        "                    and this is an indication of a windshear encounter."
  (setq aircraft-response (cons 'below-target-glideslope-displacement aircraft-response)) 'glideslope-deviation)

(dev16  "                    MONITOR: Glideslope displacement is over target by betweeen .4 and 1 dot, "
        (and (> glideslope-displacement target-glideslope-displacement)
             (< (- glideslope-displacement target-glideslope-displacement) 1)
             (> (- glideslope-displacement target-glideslope-displacement) .4)) 'target-glideslope-displacement
        "                    and this is an indication of a windshear encounter."
  (setq aircraft-response (cons 'above-target-glideslope-displacement aircraft-response)) 'glideslope-deviation)

(dev17  "                    MONITOR: Throttle setting is under target by over   percent for      , "
        (and (> target-throttle throttle)
             (> (- target-throttle throttle) 1)) 'target-throttle
        "                    and this is an indication of a microburst encounter."
        (setq aircraft-response (cons 'microburst aircraft-response)) 'throttle-deviation)

(dev18  "                    MONITOR: Throttle setting is over target by over   percent for      , "
        (and (> throttle target-throttle)
             (> (- throttle target-throttle) 1)) 'target-throttle
        "                    and this is an indication of a microburst encounter."
        (setq aircraft-response (cons 'microburst aircraft-response)) 'throttle-deviation)

(dev19  "                    MONITOR: Throttle setting is under target by between   and   percent for      , "
        (and (> target-throttle throttle)
             (< (- target-throttle throttle)  )
             (> (- target-throttle throttle)  )) 'target-throttle
        "                    and this is an indication of a windshear encounter."
  (setq aircraft-response (cons 'below-target-throttle aircraft-response)) 'throttle-deviation)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
;                    (dev20  "                          MONITOR: Throttle setting is over target by betweeen    and   percent for      , "
;                           (and (> throttle target-throttle)
;                                (< (- throttle target-throttle)  )
;                                (> (- throttle target-throttle)  )) 'target-glideslope-displacement
;                           "                          and this is an indication of a windshear encounter."
;                     (setq aircraft-response (cons 'above-target-throttle aircraft-response)) 'throttle-deviation)
                     ))
;;;
;;;;;;;;;;;;;;;;;;;;;;  comm-rules  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;    these rules direct communications procedures
;;;
;;;
;;; variable COMM-VARIABLES contains the external variables received by communications
;;;
(defvar comm-variables nil)
(setq comm-variables '(pirep tower-report llwas onboard-radar weather-report))
;;;
;;; variable MESSAGES sets up the COMM-VARIABLES
;;;
(defvar messages)
(setq messages '((pirep 0 nil nil)
                 (tower-report nil nil)
                 (llwas 0 nil nil)
                 (onboard-radar nil nil)
                 (weather-report nil nil nil)))
;;;
;;; variable PRESET-VARIABLES contains variables preset once per situation
;;;
(defvar preset-variables)
(setq preset-variables '(always-bother-the-pilot current-airport current-runway critical-agl decision-agl flare-agl aircraft-type
                        vr-field-length-limit vr v1 origin-airport destination-airport current-runway old-action))
;;;
;;; variable PRESET-MESSAGES sets up the PRESET-VARIABLES
;;;
(defvar preset-messages)
(setq preset-messages '((always-bother-the-pilot t nil nil)
                        (current-airport Dallas nil nil)
                        (current-runway runway1 nil nil)
                        (critical-agl 200 nil nil)
                        (decision-agl 200 nil nil)
                        (flare-agl 200 nil nil)
                        (aircraft-type MD-80 nil nil)
                        (vr-field-length-limit 120 nil nil)
                        (vr 110 nil nil)
                        (v1 80 nil nil)
                        (origin-airport Dallas nil nil)
                        (destination-airport Denver nil nil)
                        (current-runway 26 nil nil)
                        (old-action continue)))
;;;
;;; variable STATE-VARIABLES contains state variables that connot be set by MONITOR rules
;;;
(defvar state-variables)
(setq state-variables '(agl aircraft-response airspeed runway-remaining))
;;;
;;; variable CONTROL-VARIABLES contains controls that the RBC can tweak
;;;
(defvar control-variables)
(setq control-variables '(elevator thrust brakes))
;;;
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
;;; variable COMM-VARIABLE-PROPERTIES contains a list of property values that must be assigned for the COMM-VARIABLES
;;;
(setq comm-variable-properties '((pirep (airport runways))
                                 (tower-report (airport runways))
                                 (llwas (airport runways))
                                 (onboard-radar (airport runways))
                                 (weather-report (airport runways temp-dewpt-sprd))
                                 ))
;;;
;;;
;;;
;;;
(defvar comm-rules nil)
(setq comm-rules '(
                (comm1 "                 INTERFACE: An important message was received, "
                       (bind-variables comm-variables messages)     'messages
                       "                 so there is new information to be assessed."
                       (setq new-information-received t)     'new-information-received)

                (comm2 "                 INTERFACE: No important messages were received, "
                       't     ()
                       "                 so there is no new information to assess."
                       (setq new-information-received nil)   'new-information-received)

                (comm3 "                 INTERFACE: An assessment of weather on our intended flight path is required. "
                       't     ()
"               ***********************************************
                       *       ATTENTION           ATTENTION
                       *  CONVECTIVE   WEATHER  INFORMATION  REQUEST
                       *       ATTENTION           ATTENTION
                       *
                       *  PLEASE INDICATE IF CONVECTIVE WEATHER IS
                       *  OBSERVED ON OUR INTENDED FLIGHT PATH:
                       *    0. Convective weather not observed
                       *    1. With localized strong winds - HIGH risk
                       *    2. With heavy precipitation -- HIGH risk
                       *    3. With a rainshower -- MEDIUM risk
                       *    4. With lightning -- MEDIUM risk
                       *    5. With virga -- MEDIUM risk
                       *  ENTER 0 THROUGH 5 TO INDICATE CHOICE
                       ***********************************************"
                       (bind-variables '(observation) 'lisp-listener)   'observation)

                (comm4 "                 INTERFACE: Windshear has been encountered, "
                       (and (eq (get 'microburst-encounter 'last-value) t)
                            (eq microburst-encounter nil))    'microburst-encounter
                       "                 so a PIREP is being sent out."
                       (setq incident-reported t)           'incident-reported)

                (comm5 "                 INTERFACE: There are no incidents to report, "
                       't  ()
                       "                 so no PIREP will be sent."
                       (setq incident-reported nil)          'incident-reported)
                ))
;;;
;;;;;;;;;;;;;;;;;;;;;; action-rules ;;;;;;;;;;;;;;;;;;;;;;;;;
;;;       these rules determine the proper top level control actions
;;;
(defvar action-rules)
(setq action-rules '(
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
(action1   "     ACTION: We are in a microburst !!!! "
           (and (get-value-of 'microburst-encounter windshear-rules)
                (eq microburst-encounter t)
                (eq guidance-mode 'automatic))      'microburst-encounter
           "     Executing recovery procedures !!! "
           (get-value-of 'recovery-procedures-complete recovery-rules)  'actions-taken)

(action2   "     ACTION: We are not in a microburst, and standard procedures are recommended, "
           (and (neq microburst-encounter t)
                (eq guidance-mode 'automatic)
                (get-value-of 'recommended-procedures planning-rules)
                (eq recommended-procedures 'standard))       '(microburst-encounter recommended-procedures)
           "     so we must execute standard procedures as planned."
           (get-value-of 'standard-procedures-complete sop-rules)  'actions-taken)

(action3   "     ACTION: We are not in a microburst, but go-around procedures are recommended, "
           (and (neq microburst-encounter t)
                (eq guidance-mode 'automatic)
                (eq recommended-procedures 'go-around))    '(microburst-encounter recommended-procedures)
           "     so we must execute go-around procedures as planned."
           (get-value-of 'go-around-procedures-complete go-around-rules) 'actions-taken)

(action4   "     ACTION: We are not in a microburst, but delaying procedures are recommended, "
           (and (neq microburst-encounter t)
                (eq recommended-procedures 'delay))  '(microburst-encounter recommended-procedures)
           "     so we must execute delay procedures as recommended."
           (get-value-of 'delay-procedures-complete delay-rules) 'actions-taken)

(action5   "     ACTION: We are in a microburst !!!! "
           (and (eq microburst-encounter t)
                (eq guidance-mode 'semi-automatic))     'microburst-encounter
           "     Executing semi-automatic recovery procedures !!! "
           (get-value-of 'semi-recovery-procedures-complete semi-recovery-rules)  'actions-taken)

(action6   "     ACTION: We are not in a microburst, and standard procedures are recommended, "
           (and (neq microburst-encounter t)
                (eq guidance-mode 'semi-automatic)
                (get-value-of 'recommended-procedures planning-rules)
                (eq recommended-procedures 'standard))  'microburst-encounter
           "     so we must execute semi-automatic standard operations as planned."
           (get-value-of 'semi-standard-procedures-complete semi-sop-rules) 'actions-taken)

(action7   "     ACTION: We are not in a microburst, but go-around procedures are recommended, "
           (and (neq microburst-encounter t)
                (eq guidance-mode 'semi-automatic)
                (eq recommended-procedures 'go-around))    '(microburst-encounter recommended-procedures)
           "     so we must execute semi-automatic go-around procedures as planned."
           (get-value-of 'semi-go-around-procedures-complete semi-go-around-rules) 'actions-taken)
))
;;;
;;;;;;;;;;;;;;;;;;;;; sop-rules ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;      these rules direct standard operating procedures
;;;
(defvar sop-rules)
(setq sop-rules '(
          (sop1   "          ACTION: The mission phase is takeoff, and the controls have not been set for takeoff roll, "
                  (and (eq mission-phase 'takeoff)
                       (neq configuration-set-for-takeoff t))          '(mission-phase configuration-set-for-takeoff)
                  "        so the control laws must be set for takeoff roll at this time."
                  (get-value-of 'configuration-set-for-takeoff config-rules)  'standard-procedures-complete)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
(sop2   "          ACTION: On takeoff, groundspeed is below v1, "
        (and (eq mission-phase 'takeoff)
             (< groundspeed v1))            '(mission-phase v1)
        "          so takeoff roll control laws remain in operation."
        (f77:execute ftn-user:takeoff-cl)   'standard-procedures-complete)

(sop3   "          ACTION: On takeoff, groundspeed is above v1, airspeed is below vr, "
        (and (eq mission-phase 'takeoff)
             (> groundspeed v1)
             (< airspeed vr))               '(mission-phase v1)
        "          so takeoff roll control laws remain in operation."
        (f77:execute ftn-user:takeoff-cl)   'standard-procedures-complete)

(sop4   "          ACTION: On takeoff, airspeed is above vr, "
        (and (eq mission-phase 'takeoff)
             (> airspeed vr))               '(mission-phase vr)
        "          so rotation control laws are in effect. "
        (f77:execute ftn-user:rotation-cl)  'standard-procedures-complete)

(sop5   "          ACTION: We are climbing out, "
        (eq mission-phase 'climbout)        'mission-phase
        "          so climbout control laws are in effect."
        (f77:execute ftn-user:climbout-cl)  'standard-procedures-complete)

(sop6   "          ACTION: We are on approach, and controls are not set for landing, "
        (and (eq mission-phase 'approach)
             (neq configuration-set-for-landing t)) '(mission-phase configuration-set-for-landing)
        "          so the configuration must be set for landing."
        (get-value-of 'configuration-set-for-landing config-rules) 'standard-procedures-complete)

(sop7   "          ACTION: We are on approach, have not established a glideslope, and altitude is above critical, "
        (and (eq mission-phase 'approach)
             (neq glideslope-established t)
             (> agl critical-agl))          'glideslope-established
        "          so we must continue to establish the glideslope."
        (get-value-of 'glideslope-established glideslope-rules) 'standard-procedures-complete)

(sop8   "          ACTION: We are on approach, and have established the proper glideslope, "
        (and (eq mission-phase 'approach)
             (eq glideslope-established t))    'glideslope-established
        "          so approach control laws are in effect."
        (f77:execute ftn-user:approach-cl)     'standard-procedures-complete)

(sop9   "          ACTION We are on flare, "
        (eq mission-phase 'flare)  'mission-phase
        "          so flare control laws are in effect."
        (f77:execute ftn-user:flare-cl)        'standard-procedures-complete)

(sop10  "          ACTION: We are on landing roll, "
        (eq mission-phase 'landing-roll)   'mission-phase
        "          so landing control laws are in effect."
        (f77:execute ftn-user:landing-cl)      'standard-procedures-complete)
))
;;;;;;;;;;;;;;;;; recovery-rules ;;;;;;;;;;;;;;;;
;;;;;; these rules execute recovery procedures
;;;
(defvar recovery-rules)
(setq recovery-rules  '(
                (recvry1     "          ACTION: In microburst, AGL is above critical, aircraft type is L-1011, "
```

```
                (and (> agl critical-agl)
                     (eq aircraft-type 'L-1011)
                     (eq mission-phase 'climbout))                'agl
               !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
                 !!!  APPLY  FULL  THRUST IMMEDIATELY  !!!
                 !!!PULL TO 17.5 DEGREES PITCH ATTITUDE !!!
                 !!!        RESPECT STICK SHAKER        !!!
                 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                (f77:execute ftn-user:recover)                'recovery-procedures-complete)

  (recvry2      "        ACTION: In micorburst, AGL is above critical, "
                (and (> agl critical-agl)
                     (or (neq aircraft-type 'L-1011)
                         (and (eq aircraft-type 'L-1011)
                              (or (eq mission-phase 'approach)
                                  (eq mission-phase 'flare))))) 'agl
               !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
                 !!!  APPLY  FULL  THRUST IMMEDIATELY  !!!
                 !!! PULL TO 15 DEGREES PITCH ATTITUDE  !!!
                 !!!        RESPECT STICK SHAKER        !!!
                 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                (f77:execute ftn-user:recover)               'recovery-procedures-complete)

 (recvry3       "        ACTION:  IN MICROBURST! AGL IS BELOW CRITICAL !!!!!!!!"
                (and (neq mission-phase 'takeoff)
                     (neq mission-phase 'landing-roll)
                     (< agl critical-agl))                   'agl
               !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
                 !!!  APPLY OVERBOOST IMMEDIATELY       !!!
                 !!! PULL TO 20 DEGREES PITCH ATTITUDE  !!!
                 !!!      AVOID GROUND CONTACT          !!!
                 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                (f77:execute ftn-user:recover)              'recovery-procedures-complete)

 (recvry4       "        ACTION: In microburst, on takeoff, below V1 "
                (and (eq mission-phase 'takeoff)
                     (< airspeed v1))          'agl
               !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
                 !!!      AIRSPEED IS BELOW V1          !!!
                 !!!     ABORT TAKEOFF IMMEDIATELY      !!!
                 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                (f77:execute ftn-user:abortto)             'recovery-procedures-complete)

 (recvry5       "        ACTION: In microburst, on takeoff, too fast to abort and too slow to rotate "
                (and (eq mission-phase 'takeoff)
                     (> airspeed v1)
                     (< airspeed vr)
                     (> runway-remaining 2000))          'agl
               !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
                 !!!     AIRSPEED IS ABOVE V1           !!!
                 !!!  OVER 2000 FT RUNWAY REMAINING     !!!
                 !!!  APPLY FULL THRUST IMMEDIATELY     !!!
                 !!!      DO NOT ABORT TAKEOFF          !!!
                 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                (f77:execute ftn-user:emgyto)            'recovery-procedures-complete)

 (recvry6       "        ACTION: In microburst, on takeoff, fast enough to rotate "
                (and (or (eq mission-phase 'takeoff)
                         (eq mission-phase 'climbout))
                     (> airspeed vr))                    'agl
```

```
                                  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
                                     !!!    AIRSPEED IS ABOVE VR            !!!
                                     !!!  APPLY FULL THRUST IMMEDIATELY     !!!
                                     !!!  PULL TO 15 DEGREES PITCH ATTITUDE !!!
                                     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                                   (f77:execute ftn-user:emgyto)              'recovery-procedures-complete)

                 (recvry7      "         ACTION: In microburst, on takeoff with less than 2000 ft of runway remaining "
                                   (and (eq microburst-encounter t)
                                        (eq mission-phase 'takeoff)
                                        (< runway-remaining 2000))             'agl
                                  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
                                     !!! LESS THAN 2000 FEET OF RUNWAY LEFT !!!
                                     !!!  APPLY FULL THRUST IMMEDIATELY     !!!
                                     !!!  PULL TO 15 DEGREES PITCH ATTITUDE !!!
                                     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                                   (f77:execute ftn-user:emgyto)              'recovery-procedures-complete)
                 ))
;;;
;;;;;;;;;;;;;;;;;;;; go-around-rules ;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;      these rules execute go-around procedures
;;;
(defvar go-around-rules)
(setq go-around-rules '(
                   (goar1  "         ACTION: Tower not informed of our intent to go-around, no traffic in the way, "
                           (and (eq tower-informed-goa nil)
                                (eq traffic nil))        '(tower-informed-goa traffic)
                           "         so we must request clearance for a go-around while executing go-around control laws."
                           (and (f77:execute ftn-user:go-around-cl)
                                (get-value-of 'tower-informed-goa comm-rules))  'go-around-procedures-complete)

                   (goar2  "         ACTION: Tower not informed of our intentions, traffic in the way, altitude above critical, "
                           (and (eq tower-informed-goa nil)
                                (eq traffic t)
                                (> agl critical-agl))        '(tower-informed-goa traffic)
                           "         so we must request clearance for a go-around while executing approach control laws."
                           (and (f77:execute ftn-user:approach-cl)
                                (get-value-of 'tower-informed-goa comm-rules))  'go-around-procedures-complete)

                   (goar3  "         ACTION: Tower not informed of our intentions, traffic in the way, ALTITUDE BELOW CRITICAL! "
                           (and (eq tower-informed-goa nil)
                                (eq traffic t)
                                (< agl critical-agl))        '(tower-informed-goa traffic)
                           "         so we must request clearance for a go-around while executing go-around control laws."
                           (and (f77:execute ftn-user:go-around-cl)
                                (get-value-of 'tower-informed-goa comm-rules))  'go-around-procedures-complete)

                   (goar4  "         ACTION: Tower has been informed we are going-around, "
                           (eq tower-informed-goa t)  'tower-informed-goa
                           "         so go-around control laws are in effect."
                           (f77:execute ftn-user:go-around-cl)      'go-around-procedures-complete)
                   ))
;;;
;;;;;;;;;;;;;;;;;;;; delay-rules ;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;      these rules execute delay procedures
;;;
(defvar delay-rules)
(setq delay-rules '(
                (delay1  "         ACTION: Decision stage, tower has not been informed of our intention to delay, "
                         (eq tower-informed-delay nil)  'tower-informed-delay
```

```
                  "        so the tower will be informed of our intent to delay takeoff."
                  (get-value-of 'tower-informed-delay comm-rules) 'delay-procedures-complete)

             (delay2  "        ACTION: Prelanding decision stage, tower is informed of our intent to delay, "
                  (eq tower-informed-delay t)     'tower-informed-delay
                  "        so delay procedures are complete."
                  't                                                'delay-procedures-complete)
             ))
;;;
;;;;;;;;;;;;;;;;;;;;;; planning-rules  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;     these rules determine the recommended procedures
;;;
;;;
(defvar planning-rules nil)
(setq planning-rules '(
              (mission1     "     PLANNING: Recommended action is delay, alter, abort, and we are in a decision stage, "
                       (and (get-value-of 'recommended-action assessment-rules)
                            (eq recommended-action 'delay-alter-abort)
                            (or (eq mission-phase 'preflight-decision)
                                (eq mission-phase 'prelanding-decision)))  'recommended-action
                       "     so delaying procedures are recommended."
                       (setq recommended-procedures 'delay)          'recommended-procedures)

              (mission2     "     PLANNING: Recommended action is delay, alter, abort, but we are taking off or landing, "
                       (and (eq recommended-action 'delay-alter-abort)
                            (or (eq mission-phase 'takeoff)
                                (eq mission-phase 'climbout)
                                (eq mission-phase 'flare)
                                (eq mission-phase 'landing-roll)))       'recommended-action
                       "      so standard procedures are recommended."
                       (setq recommended-procedures 'standard)         'recommended-procedures)

              (mission3     "     PLANNING: Recommended action is delay, alter, abort, on approach, above decision height, "
                       (and (eq recommended-action 'delay-alter-abort)
                            (eq mission-phase 'approach)
                            (> agl decision-agl))                      'recommended-action
                       "     so go-around procedures are recommended."
                       (setq recommended-procedures 'go-around)        'recommended-procedures)

              (mission4     "     PLANNING: Recommended action is delay, alter, abort, on approach, below decision-height, "
                       (and (eq recommended-action 'delay-alter-abort)
                            (eq mission-phase 'approach)
                            (< agl decision-agl))                      'recommended-action
                       "     so standard procedures are recommended."
                       (setq recommended-procedures 'standard)         'recommended-procedures)

              (mission5     "     PLANNING: Recommended action take precautions, but precautions were accepted or refused,"
                       (and (eq recommended-action 'consider-precautions)
                            (eq precautions-taken t))                  '(recommended-action precautions-taken)
                       "     so standard procedures are recommended."
                       (setq recommended-procedures 'standard)      'recommended-procedures)

              (mission6     "     PLANNING: We should consider takeoff precautions "
                       (and (eq recommended-action 'consider-precautions)
                            (eq mission-phase 'preflight-decision))      'recommended-action
                       "     so the the recommended runway, flap setting, and rotation speed will be found."
                       (and (setq recommended-throttle-setting 'max-rated)
                            (get-value-of 'recommended-runway runway-rules)
                            (get-value-of 'recommended-flaps flap-rules)
                            (get-value-of 'recommended-vr speed-rules)
```

```
                (get-value-of 'precautions-taken 'precaution-rules))    'recommended-procedures)

        (mission7     "     PLANNING: We should consider approach precautions "
                (and (eq recommended-action 'consider-precautions)
                     (eq mission-phase 'prelanding-decision))       'recommended-action
                "     so the recommended runway, flap setting, and approach speed will be found."
                (and (setq 'recommended-stabilized-approach-altitude 1000)
                     (get-value-of 'recommended-runway runway-rules)
                     (get-value-of 'recommended-flaps flap-rules)
                     (get-value-of 'recommended-va speed-rules)
                     (get-value-of 'precautions-taken precaution-rules))    'recommended-procedures)

        (mission8     "     PLANNING: Recommended action is take precautions, but we are not in a decision stage, "
                (and (eq recommended-action 'consider-rpecautions)
                     (neq mission-phase 'preflight-decision))     'recommended-action
                "     so standard procedures are recommended."
                (setq recommended-procedures 'standard)      'recommended-procedures)

        (mission9     "     PLANNING: We have not established glideslope and ALTITUDE IS BELOW CRITICAL!  "
                (and (eq mission-phase 'approach)
                     (< agl critical-agl)
                     (neq glideslope-established t))    '(recommended-action glideslope-established)
                "     so go-around procedures are recommended."
                (setq recommended-procedures 'go-around)   'recommended-procedures)

        (mission10    "     PLANNING: Uncertainty in touchdown point is High! Still enough altitude to go-around, "
                (and (eq mission-phase 'approach)
                     (> agl decision-agl)
                     (> touchdown-uncertainty uncertainty-tolerance)) '(touchdown-uncertainty uncertainty-tolerance)
                "     so go-around procedures are recommended."
                (setq recommended-procedures 'go-around)   'recommended-procedures)

        (mission11    "     PLANNING: It is safe to continue at this time, "
                (eq recommended-action 'continue)        'recommended-action
                "     so standard operating procedures are now in effect."
                't            'recommended-procedures)
        ))
;
;;;;;;;;;;;;;;;;;;;;; runway-rules ;;;;;;;;;;;;;;;;;;;;;;;;;;
;     These rules set the recommended runway and suspected runway. Need to write rules to determine suspected runways.
;
(defvar runway-rules)
(setq runway-rules '(
        (run1   "          PLANNING: Some runways are suspected, "
                (neq suspected-runways nil)      'suspected-runways
                "          so the recommended runway is the longest of the other suitable runways."
                (setq recommended-runway 'longest-suitable-not-suspected) 'recommended-runway)

        (run2   "          PLANNING: No runways are suspected, "
                (eq suspected-runways nil)       'suspected-runways
                "          so the recommended runway is the longest suitable runway."
                (setq recommended-runway 'longest-suitable)      'recommended-runway)
        ))
;;;;;;;;;;;;;;;;;;;;; speed-rules ;;;;;;;;;;;;;;;;;;;;;;;;;;
;     These rules set recommended approach and rotation speed.
;
(defvar speed-rules)
(setq speed-rules '(
        (speed1 "        PLANNING: Field length limit Vr is 0 to 20 knots greater than actual gross weight Vr, "
                (and (> vr-field-length-limit target-vr)
```

```
                    (< (- vr-field-length-limit target-vr) 20))   '(standard-vr vr-field-length-limit)
                "        so the recommended Vr is now the field length limit Vr. Don't reset speed bugs, however."
                (setq recommended-vr vr-field-length-limit)          'recommended-vr)

        (speed2 "        PLANNING: Field length limit Vr is over 20 knots greater than the actual gross wieght Vr, "
                (> (- vr-field-length-limit vr) 20)     '(standard-vr vr-field-length-limit)
                "        so the recommended Vr is now the standard Vr + 20 knots. Don't reset speed bugs, however."
                (setq recommended-vr (+ 20 target-vr))    'recommended-vr)

        (speed3 "        PLANNING: Increased approach speed could be desireable, "
                't    ()
                "        if available landing field length permits, approach speed could be increased up to 20 knots."
                (setq recommended-va '0-to-20-over-standard) 'recommended-approach-speed)
        ))
;;;;;;;;;;;;;;;;;;;; flap-rules ;;;;;;;;;;;;;;;;;;;;;;;;;;;
;     These rules set RECOMMENDED-FLAPS to the value given in
;     the FAA guidelines, for takeoff or approach.
;
(defvar flap-rules)
(setq flap-rules '(
        (flap1  "        PLANNING: Mission phase is preflight, "
                (and (eq mission-phase 'preflight-decision)
                     (eq aircraft-type 'B727))            'aircraft-type
                "        so the recommended flap setting is 15 degrees."
                (setq recommended-flaps 15)          'recommended-flaps)

        (flap2  "        PLANNING: Mission phase is preflight, "
                (and (eq mission-phase 'preflight-decision)
                     (or (eq aircraft-type 'MD-80)
                         (eq aircraft-type 'B737)))      'aircraft-type
                "        so the recommended flap setting is 5 to 15 degrees. "
                (setq recommended-flaps '5-to-15)    'recommended-flaps)

        (flap3  "        PLANNING: Mission phase is preflight, "
                (and (eq mission-phase 'preflight-decision)
                     (or (eq aircraft-type 'B747)
                         (eq aircraft-type 'B757)
                         (eq aircraft-type 'B767)))     'aircraft-type
                "        so the recommended flap setting is 20 degrees. "
                (setq recommended-flaps 20)         'recommended-flaps)

        (flap4  "        PLANNING: Mission phase is preflight, "
                (and (eq mission-phase 'preflight-decision)
                     (eq aircraft-type 'DC-9-10))      'aircraft-type
                "        so the recommended flap setting is 10 or 20 degrees."
                (setq recommended-flaps '10-or-20)     'recommended-flaps)

        (flap5  "        PLANNING: Mission phase is preflight, "
                (and (eq mission-phase 'preflight-decision)
                     (or (eq aircraft-type 'DC-9-20)
                         (eq aircraft-type 'DC-9-30)
                         (eq aircraft-type 'DC-9-40)
                         (eq aircraft-type 'DC-9-50)))    'aircraft-type
                "        so the recommended flap setting is 5 or 15 degrees. "
                (setq recommended-flaps '5-or-15)       'recommended-flaps)

        (flap6  "        PLANNING: Mission phase is preflight, "
                (and (eq mission-phase 'preflight-decision)
                     (eq aircraft-type 'DC-10))          'aircraft-type
                "        so the recommended flap setting is 5 to 20 degrees."
```

```
                (setq recommended-flaps '5-to-20)    'recommended-flaps)

(flap7          "          PLANNING: Mission phase is preflight, "
                (and (eq mission-phase 'preflight-decision)
                     (eq aircraft-type 'L-1011))          'aircraft-type
                "          so the recommended flap setting is 10 to 22 degrees."
                (setq recommended-flaps '10-to-22)    'recommended-flaps)

(flap8          "          PLANNING: Mission phase is prelanding, "
                (and (eq mission-phase 'preflight-decision)
                     (or (eq aircraft-type 'B727)
                         (eq aircraft-type 'B737)
                         (eq aircraft-type 'B757)
                         (eq aircraft-type 'B767)))         'aircraft-type
                "          so the recommended flap setting is 30 degrees."
                (setq recommended-flaps 30)    'recommended-flaps)

(flap9          "          PLANNING: Mission phase is prelanding, "
                (and (eq mission-phase 'prelanding-decision)
                     (eq aircraft-type 'B747))          'aircraft-type
                "          so the recommended flap setting is 25 or 30 degrees."
                (setq recommended-flaps '25-or-30)    'recommended-flaps)

(flap10          "          PLANNING: Mission phase is prelanding, "
                (and (eq mission-phase 'prelanding-decision)
                     (eq aircraft-type 'MD-80))          'aircraft-type
                "          so the recommended flap setting is 28 degrees."
                (setq recommended-flaps 28)    'recommended-flaps)

(flap11         "          PLANNING: Mission phase is prelanding, "
                (and (eq mission-phase 'landing-decision)
                     (or (eq aircraft-type 'DC-9-10)
                         (eq aircraft-type 'DC-9-20)
                         (eq aircraft-type 'DC-9-30)
                         (eq aircraft-type 'DC-9-40)
                         (eq aircraft-type 'DC-9-50)))             'aircraft-type
                "          so the recommended flap setting is the minimum authorized flap setting for this model."
                (setq recommended-flaps 'minimum-authorized-for-this-model)    'recommended-flaps)

(flap12         "          PLANNING: Mission phase is prelanding, "
                (and (eq mission-phase 'prelanding-decision)
                     (eq aircraft-type 'DC-10))          'aircraft-type
                "          so the recommended flap setting is 35 degrees."
                (setq recommended-flaps 35)          'recommended-flaps)

(flap13         "          PLANNING: Mission phase is prelanding, "
                (and (eq mission-phase 'prelanding-decision)
                     (eq aircraft-type 'L-1011))          'aircraft-type
                "          so the recommended flap setting is 33 degrees."
                (setq recommended-flaps 33)          'recommended-flaps)
))
```

Appendix E

LISP Code for Rules of Wind Shear Assessment

```lisp
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER; Base: 10; -*-
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;                                                          ;;;;;
;;;;;                       ASSESSMENT                         ;;;;;
;;;;;         RULE BASE FOR WINDSHEAR RISK ASSESSMENT          ;;;;;
;;;;;                                                          ;;;;;
;;;;;       Started by Alex Stratton on March 12, 1988         ;;;;;
;;;;;   Modified from ASSESSMENT4 beginning July 6, 1988       ;;;;;
;;;;;                                                          ;;;;;
;;;;;        This file contains windshear avoidance rules.     ;;;;;
;;;;;    Currently, format is consistent with the functions    ;;;;;
;;;;;    in file RBC-FUNCTIONS.LISP.                           ;;;;;
;;;;;                                                          ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;
;;; define variables
;;;
;;; Variable ASSESSMENT-VARIABLES contains a list of the variables defined
;;; in this file.
;;;
(defvar assessment-variables nil)
(setq assessment-variables '(weather risk obs-risk external-assessment-variables
                             weather-rules assessment-rules risk-rules observation-rules))
;;;
;;; variable EXTERNAL-ASSESSMENT-VARIABLES contains a list of the variables which
;;;  are assumed to be already set when the ASSESSMENT-RULES are evaluated.
;;;
(defvar external-assessment-variables nil)
(setq external-assessment-variables '(windshear-rules comm-rules pirep current-airport llwas onboard-radar tower-report
                                      current-runway weather-report))
;;;
;;; Variable RISK is a numerical risk factor. Its values correspond to
;;;             0 - no risk
;;;             1 - low risk
;;;             2 - medium risk
;;;     3 or more - high risk
;;;
;;; RISK has the properties: cause
;;;
(defvar risk nil)
(setq risk 'unknown)
;;;
;;; Variables WEATHER and OBS-RISK are numerical severity factors. Their values correspond to
;;;             0 - clear weather
;;;             1 - some indication of a microburst
;;;     2 or more - high indication of a microburst
;;;
(defvar weather nil)
(setq weather 'unknown)
(defvar obs-risk 'unknown)
;;;
;;;   Rules to use in testing hypotheses
;;;
;;;   New format is:
;;;     (rulename if-doc
;;;               if-part   if-vars
;;;               then-doc
;;;               then-part then-vars)
;;;
(defvar weather-rules nil)
```

```
(setq weather-rules '(
                  (weather1   "           ASSESSMENT: Resetting weather to 0 "
                              't             ()
                              "             ... "
                              (setq weather 0)    'weather)

                  (weather2   "           ASSESSMENT: The onboard radar indicates moderate turbulence on our current flight path, "
                              (and (eq onboard-radar 'turbulence)
                                   (neq (is-it-relevant 'onboard-radar current-runway) nil))  'onboard-radar
                              "           and turbulence is an indication of a microburst."
                              (and (setq weather (+ 2 weather))
                                   (setq suspected-runways (append (get 'onboard-radar 'runways) suspected-runways)))   'weather)

                  (weather3   "           ASSESSMENT: A weather report indicates moderate turbulence on our current flight path, "
                              (and (eq weather-report 'turbulence)
                                   (neq (is-it-relevant 'weather-report current-runway) nil))  'weather-report
                              "           and turbulence is an indication of a microburst."
                              (and (setq weather (+ 2 weather))
                                   (setq suspected-runways (append (get 'weather-report 'runways) suspected-runways)))   'weather)

                  (weather4   "           ASSESSMENT: A weather report indicates a rainshower on our current flight path, "
                              (and (eq weather-report 'rainshower)
                                   (neq (is-it-relevant 'weather-report current-runway) nil))  'weather-report
                              "           and turbulence is an indication of a microburst."
                              (and (setq weather (+ 2 weather))
                                   (setq suspected-runways (append (get 'weather-report 'runways) suspected-runways)))   'weather)

                  (weather5   "           ASSESSMENT: A weather report indicates heavy precipitation on our current flight path, "
                              (and (eq weather-report 'heavy-precipitation)
                                   (neq (is-it-relevant 'weather-report current-runway) nil))  'weather-report
                              "           and turbulence is a significant indication of a microburst."
                              (and (setq weather (+ 3 weather))
                                   (setq suspected-runways (append (get 'weather-report 'runways) suspected-runways)))   'weather)

                  (weather6   "           ASSESSMENT: A weather report indicates loow-level windshear on our current flight path, "
                              (and (eq weather-report 'llws)
                                   (neq (is-it-relevant 'weather-report current-runway) nil))  'weather-report
                              "           and turbulence is a significant indication of a microburst."
                              (and (setq weather (+ 3 weather))
                                   (setq suspected-runways (append (get 'weather-report 'runways) suspected-runways)))   'weather)

                  (weather7   "           ASSESSMENT: There is a relevant tower report of turbulence, "
                              (and (eq tower-report 'turbulence)
                                   (neq (is-it-relevant 'tower-report current-runway) nil))    'tower-report
                              "           and this is an indication of a microburst."
                              (and (setq weather (+ 2 weather))
                                   (setq suspected-runways (append (get 'tower-report 'runways) suspected-runways)))  'weather)

                  (weather8   "           ASSESSMENT: There is a relevant tower report of a rainshower, "
                              (and (eq tower-report 'rainshower)
                                   (neq (is-it-relevant 'tower-report current-runway) nil))    'tower-report
                              "           and this is an indication of a microburst."
                              (and (setq weather (+ 2 weather))
                                   (setq suspected-runways (append (get 'tower-report 'runways) suspected-runways)))  'weather)

                  (weather9   "           ASSESSMENT: There is a relevant tower report of heavy precipitation, "
                              (and (eq tower-report 'blowing-dust)
                                   (neq (is-it-relevant 'tower-report current-runway) nil))    'tower-report
                              "           and this is a significant indication of a microburst."
                              (and (setq weather (+ 3 weather))
```

```
                    (setq suspected-runways (append (get 'tower-report 'runways) suspected-runways)))  'weather)

        (weather10 "            ASSESSMENT: There is a relevant tower report of localized strong winds, "
                   (and (eq tower-report 'blowing-dust)
                        (neq (is-it-relevant 'tower-report current-runway) nil))    'tower-report
                   "            and this is a significant indication of a microburst."
                   (and (setq weather (+ 3 weather))
                        (setq suspected-runways (append (get 'tower-report 'runways) suspected-runways)))  'weather)

        (weather11 "            ASSESSMENT: There is a relevant tower report of localized strong winds, "
                   (and (eq tower-report 'rings-of-dust)
                        (neq (is-it-relevant 'tower-report current-runway) nil))    'tower-report
                   "            and this is a significant indication of a microburst."
                   (and (setq weather (+ 3 weather))
                        (setq suspected-runways (append (get 'tower-report 'runways) suspected-runways)))  'weather)

        (weather12 "            ASSESSMENT: There is a relevant tower report of localized strong winds, "
                   (and (eq tower-report 'tornado-like-features)
                        (neq (is-it-relevant 'tower-report current-runway) nil))    'tower-report
                   "            and this is a significant indication of a microburst."
                   (and (setq weather (+ 3 weather))
                        (setq suspected-runways (append (get 'tower-report 'runways) suspected-runways)))  'weather)

        (weather13 "            ASSESSMENT: The onboard radar indicates heavy precipitation on our current flight path, "
                   (and (eq onboard-radar 'heavy-precipitation)
                        (neq (is-it-relevant 'onboard-radar current-runway) nil))   'onboard-radar
                   "            and this is a significant indication of a microburst."
                   (and (setq weather (+ 3 weather))
                        (setq suspected-runways (append (get 'onboard-radar 'runways) suspected-runways)))  'weather)

        (weather14 "ASSESSMENT: The temperature/dewpoint spread is between 30 and 50 degrees F, "
                   (and (> temp-dewpt-sprd 30)
                        (< temp-dewpt-sprd 50))              'weather-report
                   "and this is an indication of a microburst."
                   (setq weather (+ 2 weather))          'weather)
        ))
;
(setq assessment-rules '(
        (action1   "     ASSESSMENT: There is no computed risk of windshear, "
                   (or (and (get-value-of 'new-information-received comm-rules) (eq new-information-recieved nil))
                       (and (setq risk 'unknown) (eq (return-value-of 'risk risk-rules) 0))) 'new-information-received
                   "     so IT IS SAFE TO CONTINUE."
                   (setq recommended-action 'continue)              'recommended-action)

        (action2   "     ASSESSMENT: Risk level is now high!  "
                   (> (return-value-of 'risk risk-rules) 2)         ()
                   "     WE SHOULD ALTER OUR MISSION PLAN IMMEDIATELY."
                   (setq recommended-action 'delay-alter-abort)         'recommended-action)

        (action3   "     ASSESSMENT: Risk level is now medium, "
                   (eq risk 2)                                 'risk
                   "     so WE SHOULD CONSIDER PRECAUTIONS."
                   (setq recommended-action 'consider-precautions) 'recommended-action)

        (action4   "     ASSESSMENT: Risk level is now low, "
                   (eq risk 1)                                 'risk
                   "     so we should continue, with caution."
                   (setq recommended-action 'continue) 'recommended-action)
        ))
;
```

```
; variable RISK-RULES contains rules about windshear
;
(defvar risk-rules nil)
(setq risk-rules '(
                (risk1  "        ASSESSMENT: Setting value of risk to 0 "
                        't                              ()
                        "         ... "
                        (and (setq risk '0)
                             (putt 'risk () 'cause)
                             (setq suspected-runways '())
                             (setq weather 'unknown))                'risk)

                (risk2  "        ASSESSMENT: There is a relevant pilot report that indicates moderate airspeed change, "
                        (and (> pirep 0)
                             (< pirep 15)
                             (neq (is-it-relevant 'pirep current-runway) nil))  'pirep
                        "        so the risk level is now increased."
                        (and (setq risk (+ 1 risk))
                             (put 'risk '(pirep) 'cause)
                             (setq suspected-runways (append (get 'pirep 'runways) suspected-runways)))    'risk)

                (risk3  "        ASSESSMENT: There is a relevant LLWAS alert that indicates moderate wind shifts, "
                        (and (> llwas 0)
                             (< llwas 20)
                             (neq (is-it-relevant 'llwas current-runway) nil))  'llwas
                        "        so the risk level is now increased."
                        (and (setq risk (+ 1 risk))
                             (put 'risk (cons 'llwas (get 'risk 'cause)) 'cause)
                             (setq suspected-runways (append (get 'llwas 'runways) suspected-runways)))        'risk)

                (risk4  "        ASSESSMENT: There is a relevant pilot report that indicates significant airspeed change, "
                        (and (> pirep 15)
                             (neq (is-it-relevant 'pirep current-runway) nil))  'pirep
                        "        so the risk level is now significantly increased."
                        (and (setq risk (+ 2 risk))
                             (put 'risk (cons 'pirep (get 'risk 'cause)) 'cause)
                             (setq suspected-runways (append (get 'llwas 'runways) suspected-runways)))        'risk)

                (risk5  "        ASSESSMENT: There is a relevant LLWAS alert that indicates significant wind shifts, "
                        (and (> llwas 20)
                             (neq (is-it-relevant 'llwas current-runway) nil))  'llwas
                        "        so the risk level is now significantly increased."
                        (and (setq risk (+ 2 risk))
                             (put 'risk (cons 'llwas (get 'risk 'cause)) 'cause)
                             (setq suspected-runways (append (get 'llwas 'runways) suspected-runways)))        'risk)

                (risk6  "        ASSESSMENT: There is a relevant indication of hazardous weather, "
                        (neq (return-value-of 'weather weather-rules) 0)  'weather
                        "        so the risk level is now increased."
                        (and (setq risk (+ weather risk))
                             (put 'risk (cons 'weather (get 'risk 'cause)) 'cause))       'risk)

                (risk7  "        ASSESSMENT: There is an observation of hazardous weather on our current flight path, "
                        (and (or (neq risk 0) (eq always-bother-the-pilot t))
                             (setq obs-risk 'unknown)
                             (neq (return-value-of 'obs-risk observation-rules) 0))  'weather
                        "        so the risk level is now increased."
                        (and (setq risk (+ obs-risk risk))
                             (put 'risk (cons 'observation (get 'risk 'cause)) 'cause))       'risk)
                ))
```

```
;
; variable observation-rules contains rules relevant to onboard observation
;
(defvar observation-rules)
(setq observation-rules '(
                     (obs1  "             ASSESSMENT: Resetting obs-risk to 0 "
                            't               ()
                            "               ... "
                            (and (setq obs-risk 0)
                                 (setq observation 'unknown))       'obs-risk)

                     (obs2  "             ASSESSMENT: There is an observation of localized strong winds on our flight path, "
                            (eq (return-value-of 'observation comm-rules) 1)            'observation
                            "             and this is a significant indication of a microburst."
                            (and (setq obs-risk (+ 2 obs-risk))
                                 (setq suspected-runways (cons current-runway suspected-runways)))   'obs-risk)

                     (obs3  "             ASSESSMENT: There is an observation of heavy precipitation on our flight path, "
                            (eq (return-value-of 'observation comm-rules) 2)          'observation
                            "             and this is a significant indication of a microburst."
                            (and (setq obs-risk (+ 2 obs-risk))
                                 (setq suspected-runways (cons current-runway suspected-runways)))   'obs-risk)

                     (obs4  "             ASSESSMENT: There is an observation of a rainshower on our flight path, "
                            (eq (return-value-of 'observation comm-rules) 3)           'observation
                            "             and virga is an indication of a microburst."
                            (and (setq obs-risk (+ 1 obs-risk))
                                 (setq suspected-runways (cons current-runway suspected-runways)))   'obs-risk)

                     (obs5  "             ASSESSMENT: There is an observation of lightning on our flight path, "
                            (eq (return-value-of 'observation comm-rules) 4)            'observation
                            "             and this is a indication of a microburst."
                            (and (setq obs-risk (+ 1 obs-risk))
                                 (setq suspected-runways (cons current-runway suspected-runways)))   'obs-risk)

                     (obs6  "             ASSESSMENT: There is an observation of virga on our flight path, "
                            (eq (return-value-of 'observation comm-rules) 5)            'observation
                            "             and this is a significant indication of a microburst."
                            (and (setq obs-risk (+ 1 obs-risk))
                                 (setq suspected-runways (cons current-runway suspected-runways)))   'obs-risk)
                     ))
;
```

Appendix F

LISP Code for Rule Base Control Functions

```lisp
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: USER; Base: 10; -*-
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;                                                                ;;;;;
;;;;;                        RBC FUNCTIONS                           ;;;;;
;;;;;             FUNCTIONS FOR RULE-BASED CONTROL                   ;;;;;
;;;;;                                                                ;;;;;
;;;;;               Coded 1988   by A. Stratton                      ;;;;;
;;;;;                                                                ;;;;;
;;;;;                                                                ;;;;;
;;;;;      This program contains functions that will aid in          ;;;;;
;;;;;    the evaluation of variables using a rule base and given     ;;;;;
;;;;;    facts. The user is prompted for information by the         ;;;;;
;;;;;    program, which attempts by backward chaining to find        ;;;;;
;;;;;    the value of the key variable. Rules are fired as side      ;;;;;
;;;;;    effects of the search for a value for the variable.         ;;;;;
;;;;;                                                                ;;;;;
;;;;; (tryrule rule) -- rule firing                                  ;;;;;
;;;;;    This function evaluates a rule's premise, printing          ;;;;;
;;;;;    the rule's documentation and evaluating the consequent      ;;;;;
;;;;;    if the premise evaluates non-NIL.                           ;;;;;
;;;;;                                                                ;;;;;
;;;;; (get-value-of 'key-variable-name rules) -- simple search       ;;;;;
;;;;;    This function searches a set of rules, trying to set the    ;;;;;
;;;;;    value of the key variable using function TRYRULE.           ;;;;;
;;;;;                                                                ;;;;;
;;;;; (ask-lisp-listener-about var) -- interactive variable bind      ;;;;;
;;;;;    This function queries the lisp listener to give a value     ;;;;;
;;;;;    for VAR. Properties can also be set.                        ;;;;;
;;;;;                                                                ;;;;;
;;;;; (set-variables variables rules) -- search on a variable set    ;;;;;
;;;;;    This function involkes GET-VALUE-OF on a set of variables   ;;;;;
;;;;;    to set their value. If a GET-VALUE-OF is unsuccessful,      ;;;;;
;;;;;    SET-VARIABLES prints out a warning message.                 ;;;;;
;;;;;                                                                ;;;;;
;;;;; (fire-rules rules) -- multiple rule firing                      ;;;;;
;;;;;    This function fires the rules in the rulelist using         ;;;;;
;;;;;    TRYRULE. If any of the rules fire, FIRE-RULES returns T.    ;;;;;
;;;;;                                                                ;;;;;
;;;;; (fire-relevant-rules 'variable rules) -- all relevant rules    ;;;;;
;;;;;    This function, a cross between GET-VALUE-OF and FIRE-       ;;;;;
;;;;;    RULES, fires all rules that could set the value of the      ;;;;;
;;;;;    VARIABLE that are contained in RULES.                       ;;;;;
;;;;;                                                                ;;;;;
;;;;; (return-value-of 'variable rules) -- fire rules, return value;;;;;
;;;;;    If the value of the variable is not UNKNOWN, this           ;;;;;
;;;;;    function returns the variable's value. If the value of      ;;;;;
;;;;;    the variable is equal to UNKNOWN, RETURN-VALUE-OF gets      ;;;;;
;;;;;    the value of the variable by applying FIRE-RELEVANT-RULES   ;;;;;
;;;;;    to the rules RULES. In any case, RETURN-VALUE-OF            ;;;;;
;;;;;    returns the value of the variable upon exit.                ;;;;;
;;;;;                                                                ;;;;;
;;;;; (get-value-return 'variable rules) -- get and return a value   ;;;;;
;;;;;    If the value of the variable is not UNKNOWN, this           ;;;;;
;;;;;    function returns the variable's value. If the value of      ;;;;;
;;;;;    the variable is UNKNOWN, GET-VALUE-RETURN get the value     ;;;;;
;;;;;    of the variable by applying GET-VALUE-OF to the rules. It   ;;;;;
;;;;;    returns the value of the variable on exit.                  ;;;;;
;;;;;                                                                ;;;;;
;;;;; ($eq 'key-variable-name 'value rules) -- get-value-return+EQ   ;;;;;
;;;;;    This function combines the normal LISP function EQ with     ;;;;;
```

```
;;;;;    the function GET-VALUE-RETURN described above.           ;;;;;
;;;;;                                                              ;;;;;
;;;;; (bind-variables) -- setup function for comm-rules            ;;;;;
;;;;;    This function binds variables for the rulebase.          ;;;;;
;;;;;                                                              ;;;;;
;;;;; (set-up) -- sets up the current rulebase (as of 7/88)        ;;;;;
;;;;;    This function sets up the rulebase and fires it once.    ;;;;;
;;;;;    The inputs to the simulation are set by SEARCH-MESSAGES  ;;;;;
;;;;;    and the rulebase is searched for completion of executive ;;;;;
;;;;;    tasks.                                                   ;;;;;
;;;;;                                                              ;;;;;
;;;;; (run-rules) -- fires the current rule-base                   ;;;;;
;;;;;    This function calls the rule-base for a single search.   ;;;;;
;;;;;    This is to be used primarily from a FORTRAN simulation.  ;;;;;
;;;;;                                                              ;;;;;
;;;;; (simulate) -- execute the FORTRAN simulation                 ;;;;;
;;;;;                                                              ;;;;;
;;;;; (fetch 'property a-list) -- returns a-list values            ;;;;;
;;;;;    This function takes an A-LIST, which looks like:         ;;;;;
;;;;;       ((property1 value1) .... (propertyn valuen))          ;;;;;
;;;;;    It searches the A-LIST for the key PROPERTY. It returns  ;;;;;
;;;;;    the value of the A-LIST associated with the PROPERTY. If ;;;;;
;;;;;    it can't find the PROPERTY, it returns NIL.              ;;;;;
;;;;;                                                              ;;;;;
;;;;; (put 'vname 'pvalue 'pname) -- Z-LISP function PUTPROP       ;;;;;
;;;;;    This makes it a bit easier to call putrop, that's all.   ;;;;;
;;;;;                                                              ;;;;;
;;;;; (make-rulesprop)                                             ;;;;;
;;;;;                                                              ;;;;;
;;;;; (is-it-relevant 'alert 'runway 'flight-path) -- relevance    ;;;;;
;;;;;    This function is called to determine the relevance of the ;;;;;
;;;;;    ALERT given to the windshear expert system. Relevance is ;;;;;
;;;;;    assessed by first comparing the AIRPORT property of ALERT ;;;;;
;;;;;    to CURRENT-AIRPORT, and comparing the RUNWAYS property of ;;;;;
;;;;;    ALERT to RUNWAY, or examining the AREA property          ;;;;;
;;;;;    of ALERT. IS-IT-RELEVANT returns T if FLIGHT-PATH passes ;;;;;
;;;;;    through AREA (as determined by FLIGHT-PATH-IN below), or ;;;;;
;;;;;    if the RUNWAYS property contains RUNWAY. If quantities   ;;;;;
;;;;;    needed to assess relevance aren't known, IS-IT-RELEVANT  ;;;;;
;;;;;    returns UNKNOWN. IS-IT-RELEVANT only returns NIL if the  ;;;;;
;;;;;    alert is definitely not relevant.                        ;;;;;
;;;;;                                                              ;;;;;
;;;;; (flight-path-in 'area) -- determine if FLIGHT-PATH is in AREA;;;;;
;;;;;         not yet written                                      ;;;;;
;;;;;                                                              ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;
;;;
;;;
;;;  define variables
;;;
(defvar flag nil)
;;;
;;;
;;;   Define the rule base using a RULES variable.
;;;
;;;   Curent format is:
;;;    (defvar myrules nil)
;;;    (setq myrules '((rulename if-doc if-part if-vars then-doc then-part then-vars)
;;;                                       :
```

```
;;;                    (lastrule if-doc if-part if-vars then-doc then-part then-vars)))
;;;
;;;      Now, 'ifpart' and 'thenpart' must be valid LISP expressions.
;;;    'Ifpart' must be an expression that evaluates to something or NIL. The 'thenpart'
;;;    cannot be a nested expression; its second element is expected to be a
;;;    variable. If the 'ifpart' is evaluated non-NIL (this actually happens in function
;;;    TRYRULE), then the 'thenpart' is evaulated. The presumption of this
;;;    program is that variables (or at least one of them) are given
;;;    values in the 'thenpart' of a rule if their values are unknown. Otherwise,
;;;    the program will query the user for the values of some of the variables.
;;;      'If-doc' and 'then-doc' are documentation lines - they should be character
;;;    strings, suitable for printing, and set in double quotes. As they are printed
;;;    only when the if-part evaluates to non-NIL, the statements should reflect the
;;;    fact that the rule has successfully fired.
;;;      'If-vars' should contain a list of INTERNAL variables that must have values in order for
;;;    the rule to fire (if-part to evaluate non-NIL). 'Then-vars' should list the variable,
;;;    if any, that will be set if the then-part is evaluated. If-vars is not being used yet.
;
;
;
;  Functions IF-PART and THEN-PART return the if-part and the then-part,
;  respectively, of the argument RULE.
;
(defun if-part (rule)
      (third rule))
(defun then-part (rule)
      (sixth rule))
;
; Functions IF-DOC and THEN-DOC return the comments for the if and
; then conditions, which are then printed out by TRYRULE.
;
(defun if-doc (rule)
      (second rule))
(defun then-doc (rule)
      (fifth rule))
;
;  Function IF-VARS returns the internal variable that must have values
;  if the if-part of the rule is to fire.
;  Function THEN-VARS returns the variables set by RULE.
;
(defun if-vars (rule)
      (fourth rule))
(defun then-vars (rule)
      (seventh rule))
;
; Function TRYRULE evaluates its argument RULE. It tests the 'ifpart'
; of the rule; if it is true, the 'thenpart' is evaluated, and the
; documentation is printed out to the user. If the
; 'ifpart' is not true, TRYRULE returns NIL.
;
(defun tryrule (rule)
      (prog (flag)
      (setq flag nil)
      (cond ((eval (if-part rule)) (setq flag t)))
      (cond ((eq flag t) (and (zl:cursorpos 'a)
                              (princ (if-doc rule))
                              (zl:cursorpos 'a)
                              (zl:cursorpos 'f)
                              (zl:cursorpos 'f)
                              (zl:cursorpos 'f)
```

```lisp
                                (zl:cursorpos 'f)
                                (zl:cursorpos 'f)
                                (zl:cursorpos 'f)
                                (zl:cursorpos 'f)
                                (zl:cursorpos 'f)
                                (zl:cursorpos 'f)
                                (zl:cursorpos 'f)
                                (princ (then-doc rule))
                                (eval (then-part rule)))))
       (cond ((eq flag t) (return t)))))
;
; Function GET-VALUE-OF does a simple goal-directed search. The strategy used by
; GET-VALUE-OF is as follows:
;          -search each rule's then-vars to see if they can update
;           VAR's value.
;          -call TRYRULE to try to fire the relevant rules. If a
;           rule fires (TRYRULE returns T), GET-VALUE-OF returns T.
;           If none of the rules fire, GET-VALUE-OF returns NIL.
;
(defun get-value-of (var rules)
       (prog (oldval)
             (setq oldval (eval var))
             (cond ((null rules) (and (ask-lisp-listener-about var) (go endpt)))
                   ((and (member var (then-vars (car rules)))
                         (tryrule (car rules))) (go endpt))
                   (t (get-value-of var (cdr rules))))
             endpt
;            (print oldval)
;            (print (eval var))
             (cond ((neq (eval var) oldval) (put var oldval 'last-value))))
       't)
;
; Function ASK-LISP-LISTENER-ABOUT queries the lisp listener to get
; a value for VAR
;
(defun ask-lisp-listener-about (var)
       (zl:cursorpos 'a)
       (princ "INFERENCE ENGINE: Please give me a value for")
       (print var)
       (princ "=")
       (set var (read))
       't)
;
; Function SET-VARIABLES involkes GET-VALUE-OF for a set of variables.
; If GET-VALUE-OF is unsuccessful, SET-VARIABLES prints out a warning
; message.
;
(defun set-variables (vars rules)
       (cond ((null vars) t)
             ((get-value-of (car vars) rules) (set-variables (cdr vars) rules))
             (t (and (zl:cursorpos 'a)
                     (princ "couldn't get any value of ")
                     (princ (car vars))
                     (set-variables (cdr vars) rules)))))
;
; Function FIRE-RULES fires several rules as described above.
;
(defun fire-rules (rules)
       (do ((remaining-rules rules) (flag nil)) ((null remaining-rules) flag)
           (cond ((tryrule (car remaining-rules)) (setq flag t)))
```

```
          (setq remaining-rules (cdr remaining-rules))))
;
; Function FIRE-RELEVANT-RULES fires rules that can set VARIABLE as described above.
;
(defun fire-relevant-rules (var rules)
       (cond ((null rules) nil)
             ((and (member var (then-vars (car rules))) (tryrule (car rules)))
              (fire-relevant-rules var (cdr rules)))
             (t (fire-relevant-rules var (cdr rules)))))


;
; Function RETURN-VALUE-OF returns the value of VARIABLE, as written above.
; If the FIRE-RELEVANT-RULES is unsuccessful,
; RETURN-VALUE-OF still returns the variable's value (i.e., UNKNOWN)
;
(defun return-value-of (variable rules)
       (cond ((neq (eval variable) 'unknown) (eval variable))
             ((fire-relevant-rules variable rules) (eval variable))
             (t (eval variable))))    ; this line fires if FIRE-RELEVANT-RULES fails
;
; Function GET-VALUE-RETURN returns the value of a variable, getting
; a value for it if it is UNKNOWN using GET-VALUE-OF. If GET-VALUE-OF
; is unsuccessful, GET-VALUE-RETURN still returns the variables value.
;
 (defun get-value-return (variable rules)
       (cond ((neq (eval variable) 'unknown) (eval variable))
             ((get-value-of variable rules) (eval variable))
             (t (eval variable))))    ;this line fires if GET-VALUE-OF fails
;
; Function $EQ combines GET-VALUE-RETURN and EQ
;
(defun $eq (var val rules)
       (get-value-return var rules)
       (eq var val))
;
; Function BIND-VARIABLES searches a list of MESSAGES
;  to set values for VARIABLES
;
; Format for defining the MESSAGES is:
;      (setq my-messages '((varname1 varval1)
;                                   :
;                          (varname1 varval1)))
; BIND-VARIABLES binds the value 'varvali' to the variable 'varnamei'
;
(defun bind-variables (variables messages)
       (do ((var (car variables) (car variables)) (messwipe) (flag)) ((null variables) flag)
           (setq variables (cdr variables))
           (setq messwipe messages)
           (cond ((eq messages 'lisp-listener) (and (ask-lisp-listener-about var) (go loop))))
           (do ((message (car messwipe) (car messwipe))) ((null messwipe))
               (setq messwipe (cdr messwipe))
;              (print var)
;              (print message)
;              (print variables)
;              (print variables)
               (cond ((eq (car message) var) (setq flag t)
                                              (set var (second message))
                                              (putt var (third message) 'airport)
                                              (putt var (fourth message) 'runways)
                                              (go loop))
```

```
                    ((null messwipe) (ask-lisp-listener-about var))))
;
;interactive variable bind, doesn't seem to give the right result here
;
;           (zl:cursorpos 'a)
;           (princ "COMMUNICATIONS: Please give me a value for")
;           (print var)
;           (princ "=")
;           (set var (read))
            loop))
;
;  Function SET-UP sets up a simulation of the current rulebase
;
(defun set-up ()
       (zl:cursorpos 'a)
       (princ "*******************************************")
       (zl:cursorpos 'a)
       (princ "    SIMULATION OF THE CURRENT RULEBASE     ")
       (zl:cursorpos 'a)
       (princ " ")
       (zl:cursorpos 'a)
       (princ "  COMM-VARIABLES have been set to:")
       (zl:cursorpos 'a)
       (print comm-variables)
       (zl:cursorpos 'a)
       (princ " ... and the value of MESSAGES is:")
       (zl:cursorpos 'a)
       (print messages)
       (zl:cursorpos 'a)
       (princ "  PRESET-VARIABLES have been set to:")
       (zl:cursorpos 'a)
       (print preset-variables)
       (zl:cursorpos 'a)
       (princ " ... and the value of PRESET-MESSAGES is:")
       (zl:cursorpos 'a)
       (print preset-messages)
       (zl:cursorpos 'a)
       (princ " ")
       (zl:cursorpos 'a)
       (princ "  Well, here we go ......  ")
       (zl:cursorpos 'a)
       (princ " ")
       (zl:cursorpos 'a)
       (setq ksearch 0)
       (get-value-of 'executive-procedures-complete exec-rules))

;
; Function RUN-RULES runs the current rulebase
;
(defun run-rules ()
       (get-value-of 'executive-procedures-complete exec-rules))
;
; Function SIMULATE runs the simulation program ACRAFT.FORTRAN
;
(defun simulate ()
       (f77:execute ftn-user:acraft :units ((7 "ISAAC:>alex>out1.data") (8 "ISAAC:>alex>out2.data") (9 "ISAAC:>alex>out3.data")
                                   (10 "ISAAC:>alex>out4.data"))))
;
; Function FETCH reutrns a property value of an A-LIST
;
```

```
(defun fetch (keyprop alist)
      (second (assoc keyprop alist)))
;
; Function PUT is just the zeta-lisp function PUTPROP
;
; syntax is : (put 'name-of-variable 'value-of-the-property 'name-of-the-property
;
; To get properties, use the C-LISP function GET
; syntax is : (get 'name-of-variable 'name-of-the-property)
;
(defun put (varname pval prop)
      (zl:putprop varname pval prop))
;
; Function PUTT is the smae as PUT except that it always returns T.
(defun putt (varname pval prop)
      (zl:putprop varname pval prop)
      't)
;
; Function WRITE-RULEDOC writes out the documentation for a rule base
;
(defun write-ruledoc (rules)
      (cond ((null rules) nil)
            ('t (and (zl:cursorpos 'a)
                     (princ " ")
                     (zl:cursorpos 'a)
                     (princ (if-doc (car rules)))
                     (zl:cursorpos 'a)
                     (princ (then-doc (car rules)))
                     (write-ruledoc (cdr rules))))))
;
;
;  Function MAKE-RULESPROP takes RULES and finds all rules that could
;  set the value of VAR (i.e. finds rules that have VAR as a member of their THEN-VARS).
;  It puts them into the 'RULES property of VAR.
;
(defun make-rulesprop (var rules)
      (cond ((null rules) nil)
            ((member var (then-vars (car rules))) (put var (cons (car rules) (get var 'rules)) 'rules)))
      (cond ((neq rules nil) (make-rulesprop var (cdr rules)))))
;
; Function MAKE-RULEBASE could be written now
;  would want to make new versions of g-v-o, g-v-r, r-v-o, etc.
;  I'll wait for now
;
;
; Function IS-IT-RELEVANT determines the relevance of an ALERT to
;  CURRENT-AIRPORT and RUNWAYS (and eventually FLIGHT-PATH)
;
(defun is-it-relevant (alert runway)
       (prog (flag)
       (cond ((eq (get alert 'airport) current-airport) (setq flag t))
             (t (return nil)))
       (cond ((member runway (get alert 'runways)) (return t))
             ((eq '(unknown) (get alert 'runways)) (return 'unknown))
             (t (return nil)))))
```

Appendix G

Typical Screen Displays from Wind Shear Expert System

```
Command: (simulate)
********************************************
    SIMULATION OF THE CURRENT RULEBASE

  COMM-VARIABLES have been set to:

(PIREP TOWER-REPORT LLWAS ONBOARD-RADAR WEATHER-REPORT)
 ... and the value of MESSAGES is:

((PIREP 0 NIL NIL) (TOWER-REPORT NIL NIL NIL) (LLWAS 0 NIL NIL) (ONBOARD-RADAR NIL NIL NIL) (WEATHER-REPORT NIL NIL NIL
))
  PRESET-VARIABLES have been set to:

(ALWAYS-BOTHER-THE-PILOT CURRENT-AIRPORT CRITICAL-AGL DECISION-AGL AIRCRAFT-TYPE VR-FIELD-LENGTH-LIMIT VR V1 ORIGIN-AIR
PORT DESTINATION-AIRPORT CURRENT-RUNWAY OLD-ACTION)
 ... and the value of PRESET-MESSAGES is:

((ALWAYS-BOTHER-THE-PILOT T NIL NIL) (CURRENT-AIRPORT DALLAS NIL NIL) (CRITICAL-AGL 200 NIL NIL) (DECISION-AGL 100 NIL
NIL) (AIRCRAFT-TYPE MD-80 NIL NIL) (VR-FIELD-LENGTH-LIMIT 120 NIL NIL) (VR 110 NIL NIL) (V1 80 NIL NIL) (ORIGIN-AIRPORT
 DALLAS NIL NIL) (DESTINATION-AIRPORT DENVER NIL NIL) (CURRENT-RUNWAY 26 NIL NIL) (OLD-ACTION CONTINUE))

  Well, here we go ......

EXECUTIVE: Executing search of rulebases,
          to complete executive procedures.
EXECUTIVE: We have not completed terminal operations,
          so we should continue to search.
               COMMUNICATIONS: This is the first search of the rulebase,
                         so presets will be bound at this time.
INFERENCE ENGINE: Please give me a value for
AGL =0
INFERENCE ENGINE: Please give me a value for
AIRCRAFT-RESPONSE =normal
INFERENCE ENGINE: Please give me a value for
AIRSPEED =0
INFERENCE ENGINE: Please give me a value for
RUNWAY-REMAINING =10000
          MONITOR: We are near the origin airport, stopped, and on the ground,
                    so the mission phase is preflight.
               COMMUNICATIONS: An important message was received,
                         so there is new information for situation assessment.
          MONITOR: Aircraft response is not abnormal,
                    so there is no windshear alert at this time.
          ASSESSMENT: Setting value of risk to 0
                         ...
**MORE**█
Lisp Listener 1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                ASSESSMENT: Resetting obs-risk to 0
                            ...
                COMMUNICATIONS: An assessment of weather on our intended flight path is required.
                            ***********************************************
                            *        ATTENTION          ATTENTION
                            *  CONVECTIVE   WEATHER  INFORMATION  REQUEST
                            *        ATTENTION          ATTENTION
                            *
                            *  PLEASE INDICATE IF CONVECTIVE WEATHER IS
                            *  OBSERVED ON OUR INTENDED FLIGHT PATH:
                            *    0. Convective weather not observed
                            *    1. With localized strong winds - HIGH risk
                            *    2. With heavy precipitation -- HIGH risk
                            *    3. With a rainshower -- MEDIUM risk
                            *    4. With lightning -- MEDIUM risk
                            *    5. With virga -- MEDIUM risk
                            *  ENTER 0 THROUGH 5 TO INDICATE CHOICE
                            ***********************************************
INFERENCE ENGINE: Please give me a value for
OBSERVATION =2
                ASSESSMENT: There is an observation of heavy precipitation on our flight path,
                          and this is a significant indication of convective weather.
           ASSESSMENT: There is an observation of hazardous weather on our current flight path,
                     so the risk level is now increased.
     ASSESSMENT: Risk level is now high!
                WE SHOULD ALTER OUR MISSION PLAN IMMEDIATELY.
     MISSION-PLANNING: We must alter the takeoff plan, and we are in preflight,
                so we should inform the tower that we wish to delay the takeoff.
                MONITOR: Recommended action has changed, but we have not executed a recovery,
                          so there has been no incident.
                COMMUNICATIONS: There are no incidents to report,
                          so no PIREP will be sent.
EXECUTIVE: Executing search of rulebases,
           to complete executive procedures.
           MONITOR: We are near the origin airport, stopped, and on the ground,
                     so the mission phase is preflight.
EXECUTIVE: We have not completed terminal operations,
           so we should continue to search.
                COMMUNICATIONS: This is not the first search of the rulebase,
                          so presets will not be bound at this time.
INFERENCE ENGINE: Please give me a value for
AGL =0
INFERENCE ENGINE: Please give me a value for
AIRCRAFT-RESPONSE =normal
INFERENCE ENGINE: Please give me a value for
AIRSPEED =30
```

Lisp Listener 1

```
INFERENCE ENGINE: Please give me a value for
RUNWAY-REMAINING =9000
          MONITOR: We are near the origin airport, and on the ground,
                    so the mission phase is takeoff.
               COMMUNICATIONS: An important message was received,
                         so there is new information for situation assessment.
          MONITOR: Aircraft response is not abnormal,
                    so there is no windshear alert at this time.
          ASSESSMENT: Setting value of risk to 0
                      ...
               ASSESSMENT: Resetting obs-risk to 0
                              ...
               COMMUNICATIONS: An assessment of weather on our intended flight path is required.
                          ***********************************************
                           *        ATTENTION          ATTENTION
                           *  CONVECTIVE   WEATHER  INFORMATION  REQUEST
                           *        ATTENTION          ATTENTION
                           *
                           *  PLEASE INDICATE IF CONVECTIVE WEATHER IS
                           *  OBSERVED ON OUR INTENDED FLIGHT PATH:
                           *    0. Convective weather not observed
                           *    1. With localized strong winds - HIGH risk
                           *    2. With heavy precipitation -- HIGH risk
                           *    3. With a rainshower -- MEDIUM risk
                           *    4. With lightning -- MEDIUM risk
                           *    5. With virga -- MEDIUM risk
                           *  ENTER 0 THROUGH 5 TO INDICATE CHOICE
                           ***********************************************
INFERENCE ENGINE: Please give me a value for
OBSERVATION =2
               ASSESSMENT: There is an observation of heavy precipitation on our flight path,
                         and this is a significant indication of convective weather.
          ASSESSMENT: There is an observation of hazardous weather on our current flight path,
                    so the risk level is now increased.
     ASSESSMENT: Risk level is now high!
               WE SHOULD ALTER OUR MISSION PLAN IMMEDIATELY.
     MISSION-PLANNING: We must alter the takeoff plan, and airspeed is below V1,
               so we should abort the takeoff immediately.
               MONITOR: Recommended action has not changed,
                         so there has been no incident.
               COMMUNICATIONS: There are no incidents to report,
                         so no PIREP will be sent.
EXECUTIVE: Executing search of rulebases,
          to complete executive procedures.
          MONITOR: We are near the origin airport, and on the █
AIRSPEED =30
Lisp Listener 1
```

Appendix H

Charts for Program Status Report, June, 1988

INTELLIGENT GUIDANCE AND CONTROL FOR WINDSHEAR ENCOUNTERS

INTRODUCTION AND PROGRESS REPORT

Alex Stratton
Princeton University
June 23-24, 1988

Overview

- Introduction
- Expert System Development
- Plans for Future Work

Motivation for the Study

- Strong windshears, especially microbursts, can cause fatal accidents
- Pilot proficiency in coping with microburst windshear is difficult given rarity of phenomenon
- Proper decision making and control strategy can enhance the possibility of avoidance and survival
- F.A.A. Pilot Windshear Guide establishes such a strategy

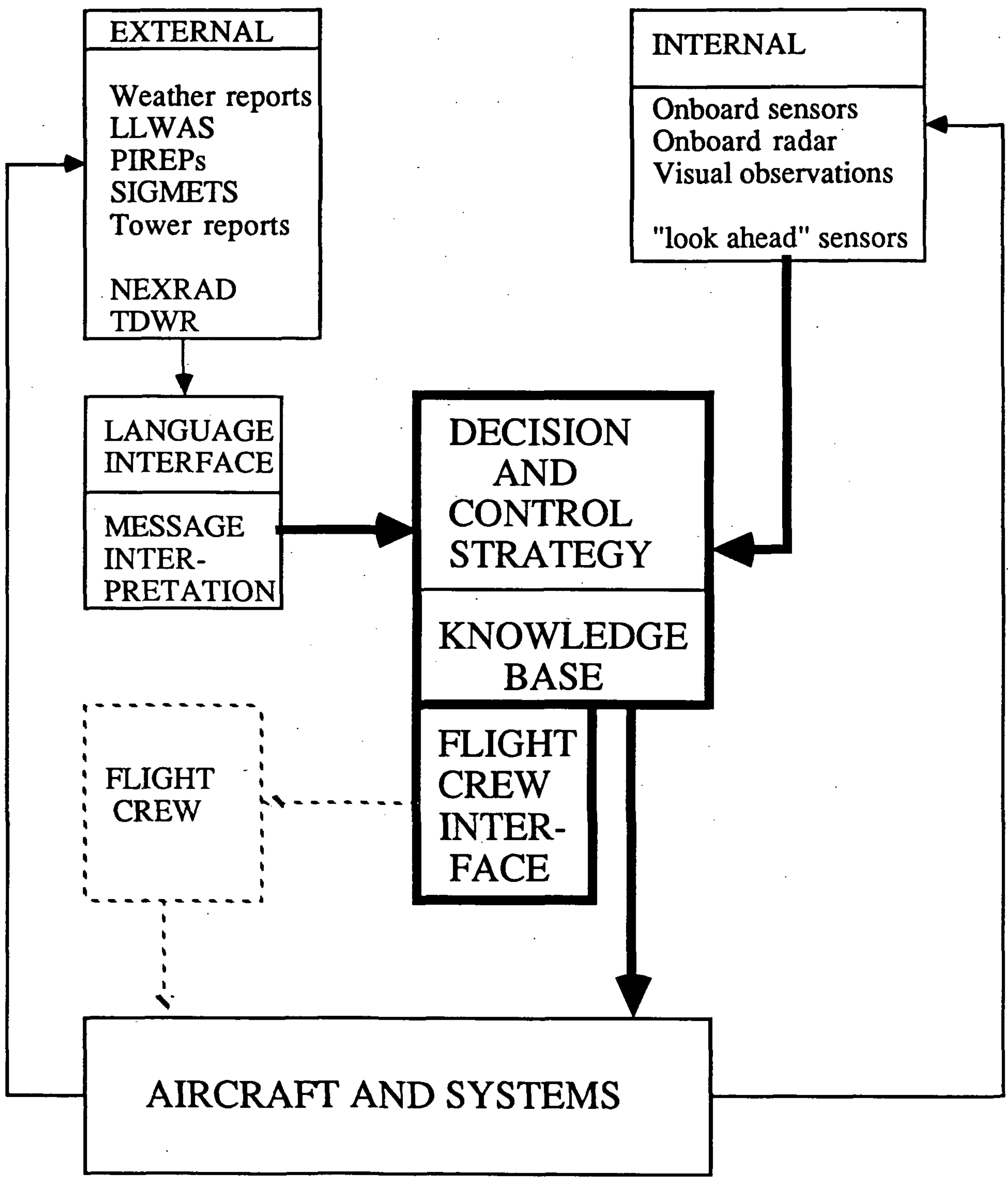
Intelligent G&C Research Goals

- Implement the Pilot Windshear Guide pilot and crew model as a rule-based (intelligent) controller

- Investigate guidance and control concepts applied to windshear encounters

- Develop a rule-based pilot aid for windshear survival

Rule-based Pilot Aid - Schematic



Sources of Decision and Control Knowledge

- Prior knowledge of windfield - Optimal control laws

- No knowledge of windfield - Strategy based on current knowledge in cockpit - Pilot Windshear Guide

Pilot Windshear Guide

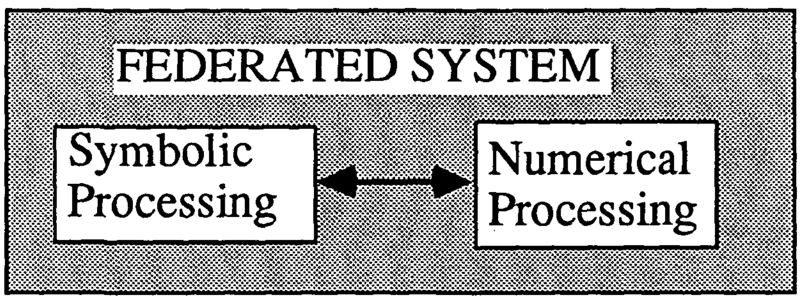
- Completed as part of the FAA's Windshear Training Aid

- Reference guide for flight crews detailing windshear avoidance, recognition, and recovery techniques

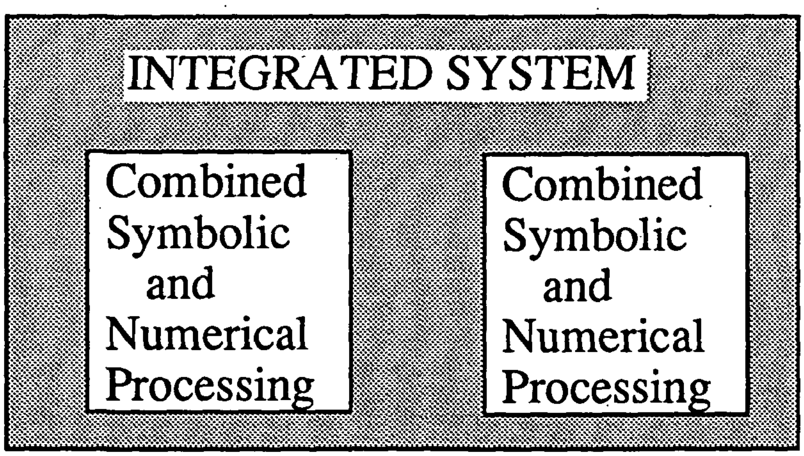
- Provides a model of pilot and crew decision-making that increases flight safety

Rule-based Systems for Control

- Federated systems - Symbolic supervisory control
  Separate numeric processing



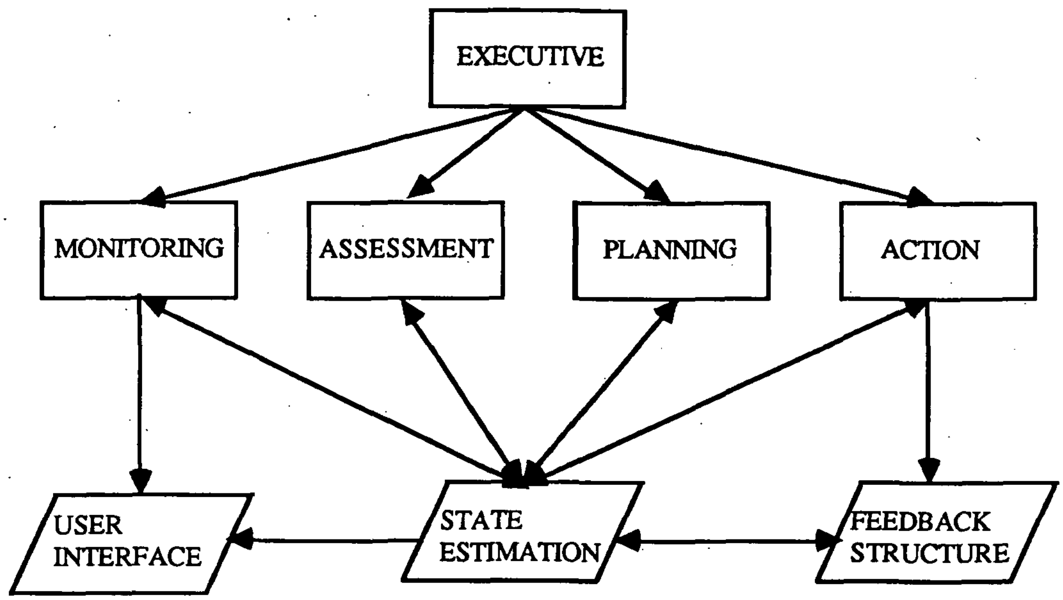
- Integrated systems - Combined symbolic and numeric control



Expert System Control Functions

- MONITORING
- ASSESSMENT -- INTERPRETATION -- DIAGNOSIS
- PREDICTION -- PLANNING
- ACTION -- CONTROL

Expert System General Structure



Monitoring and Assessment

- MONITORING -
  Observe onboard sensors
  Receive reports, alerts, warnings
  Send reports, communicate with flight crew

- ASSESSMENT -
  Determine the risk of continuing
  Decide if the flight plan should be altered
  Decide if windshear has been encountered

Planning and Action

- PLANNING -
  Delay or abort terminal operations if necessary
  Take additional safety precautions

- ACTION -
  Select and implement guidance and control laws
  Send commands to control surfaces and throttle

Expert System Implementation

- LISP -- Knowledge representation, symbolic processing
- FORTRAN -- Numerical procedures, algorithms
- Symbolics 3670 LISP Machine

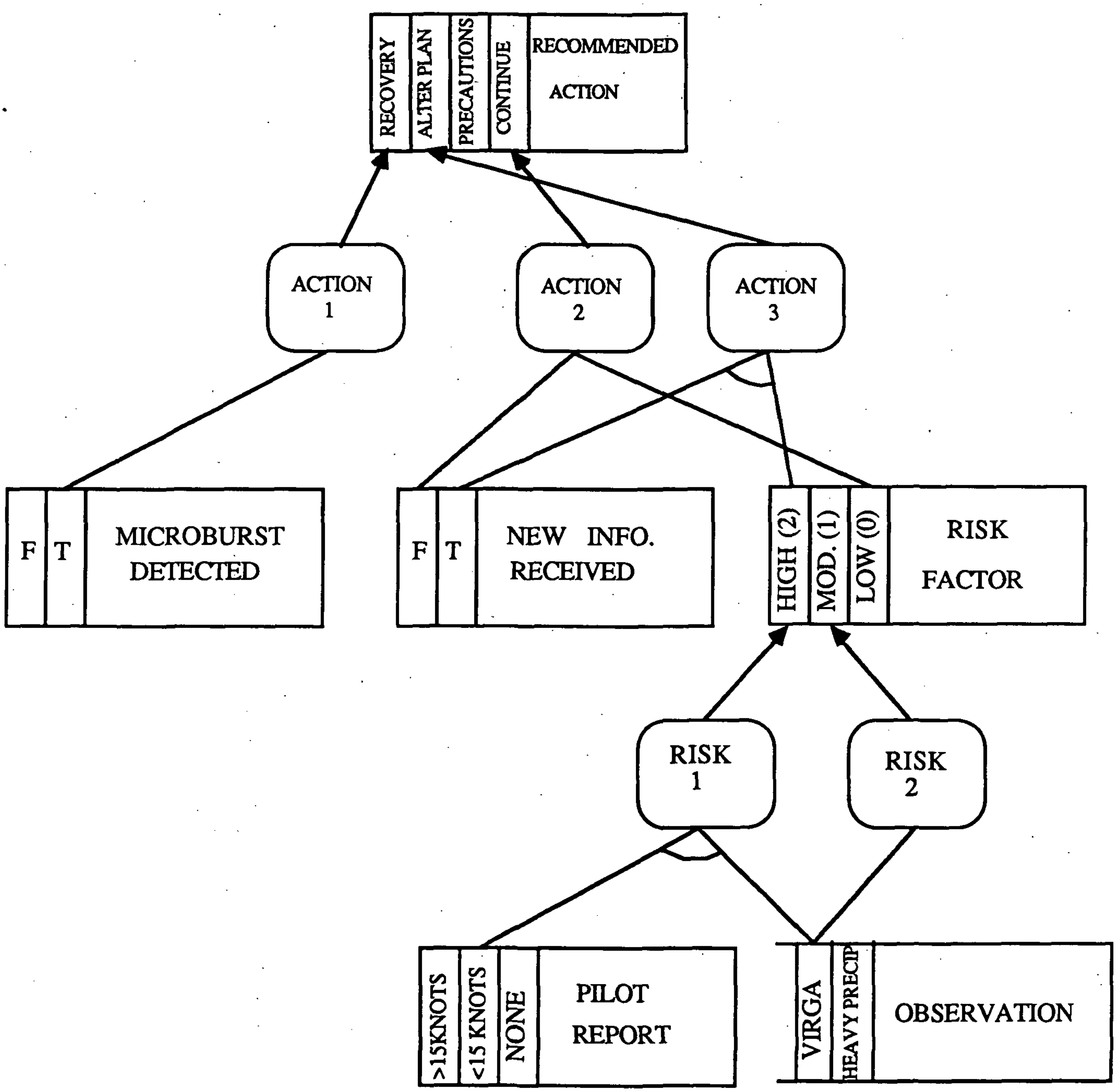
Flow of Control Driven By Goal-Directed Search Procedure

- Basic search procedure evaluates rules to set the value of a goal parameter

- Search procedure is often involked in the premise of a rule, resulting in backward-chaining

- More complicated search procedures are built with the basic procedure

Graphical Representation of Search



Example of Knowledge Acquisition

PILOT WINDSHEAR GUIDE STATEMENT:

"The choice of takeofff flap selection is dependant on the airplane type. The following should be considered unless limited by obstacle clearance and/or climb gradient:

Airplane Type	Takeoff Flap Setting
B727	15
B737	5 to 15
:	:

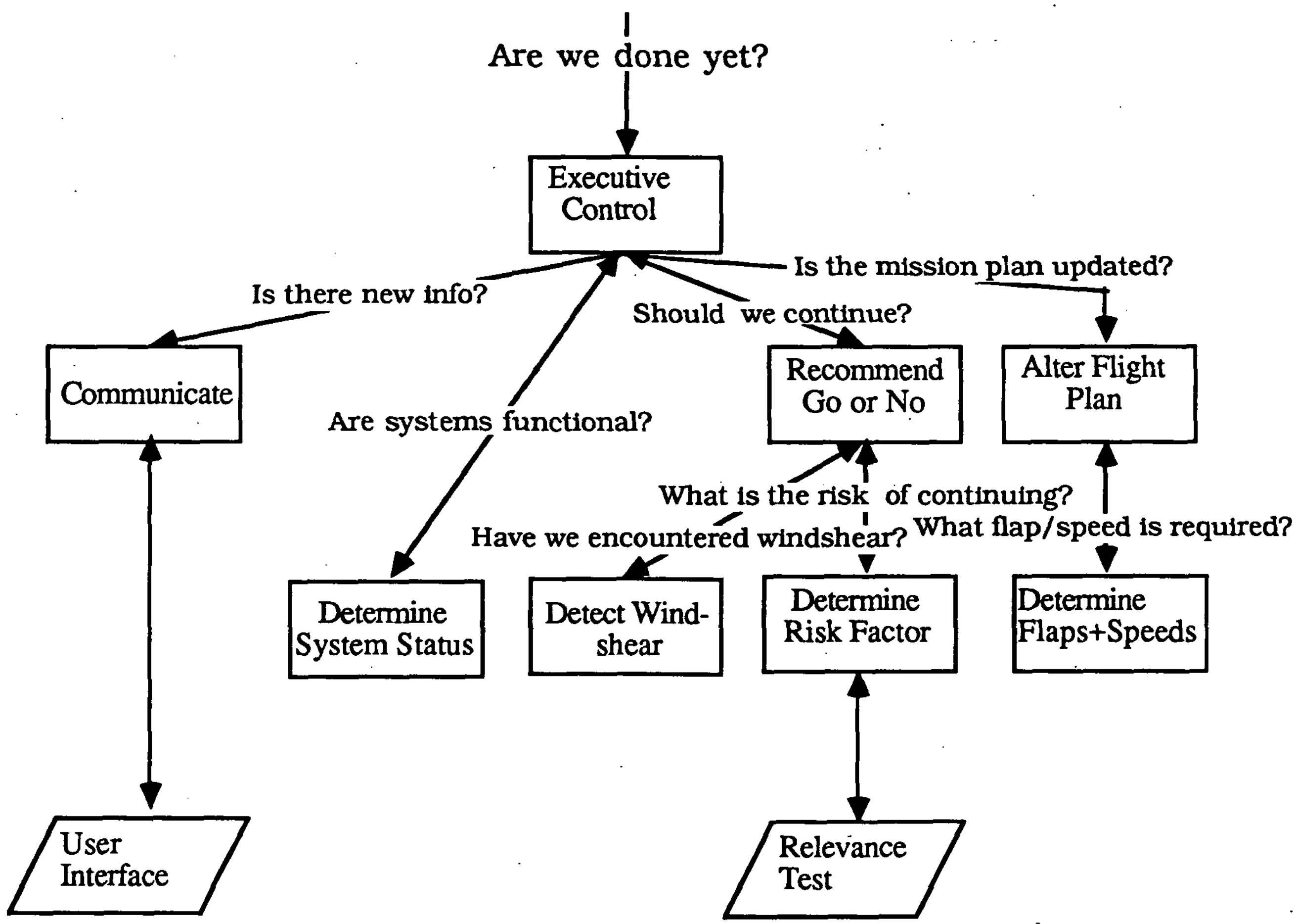
"

PSEUDO-LISP TRANSLATION:

IF recommended-action equals "take-precautions"
AND aircraft-type equals "B727"

THEN set recommended-flap-setting to "15"

Expert System Detailed Structure



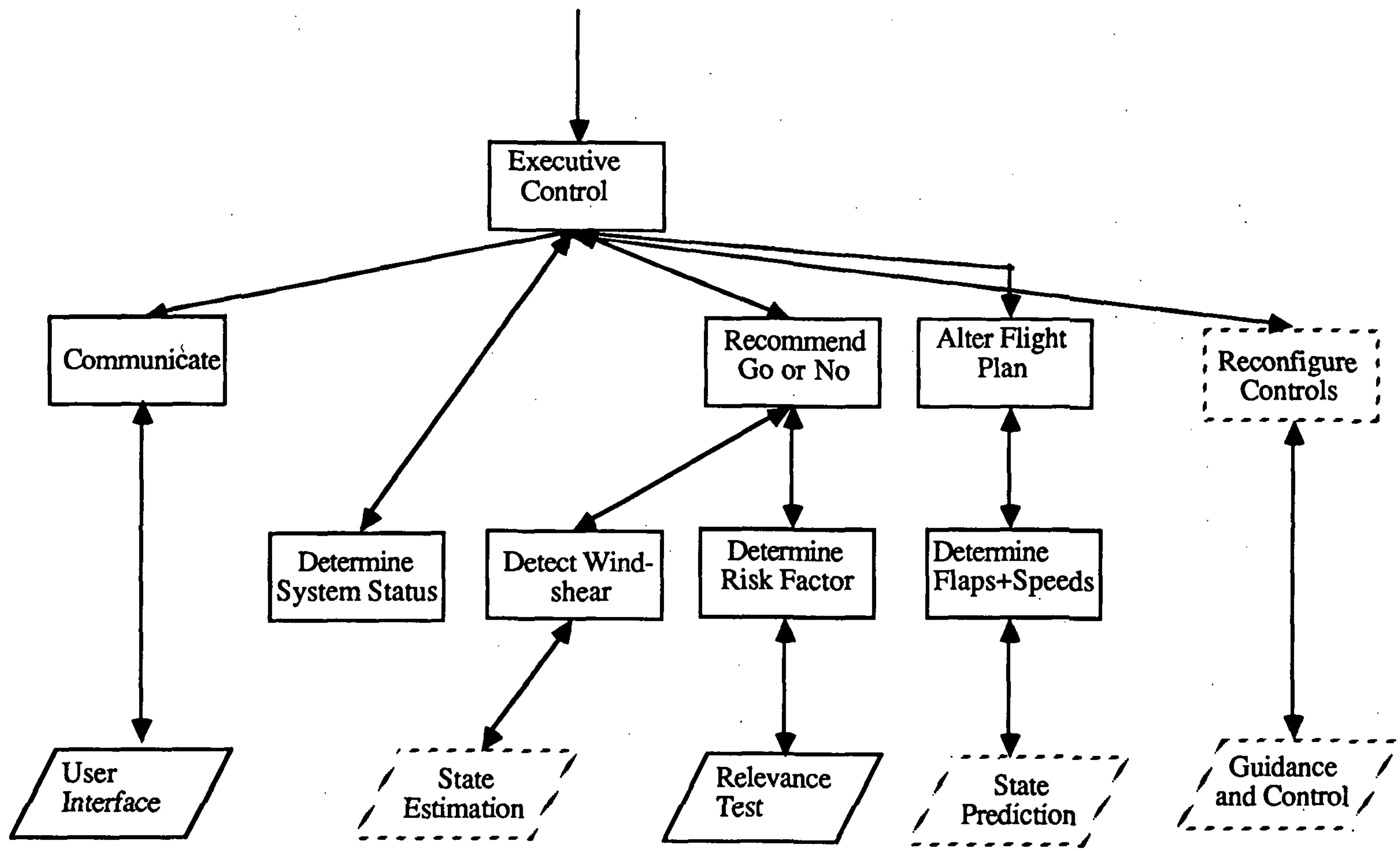
Next Additions to Expert System

- FORTRAN procedures and simulation environment will be completed next

- Low-level rules to interface FORTRAN procedures will complete knowledge base for Pilot Windshear Guide

- Improved user interface will aid the understanding and further development of expert system pilot aid

Next Additions to Expert System



Plans for Future Work

- Consider impact of next-generation sensors and ATC

- Real-time implementation of RBC

- Consider pilot/RBC interface