TWENTYFIFTH EUROPEAN ROTORCRAFT FORUM

PAPER N. C5

PARALLEL IMPLEMENTATION OF AERODYNAMICS APPLICATION: MESSAGE PASSING VS COORDINATION FRAMEWORK

BY

S.SANCESE*, A. MESSINA*, P. CIANCARINI*, U. IEMMA†
*Dipartimento Scienze dell'Informazione, Università Bologna, Italy
†Dipartimento Ingegneria Meccanica e Industriale, Università Roma Tre, Italy

SEPTEMBER 14–16, 1999
ROME
ITALY

ASSOCIAZIONE INDUSTRIE PER L'AEROSPAZIO, I SISTEMI E LA DIFESA
ASSOCIAZIONE ITALIANA DI AERONAUTICA ED ASTRONAUTICA

ABSTRACT

Sophisticated and specialized tools are necessary to the full exploitation of modern computing resources. Once available, the computing environment endowed with such tools may be defined a Problem Solving Environment (PSE) which provides all the computing facilities to solve a well defined target class of problems.

Numerical simulation is a major part of a PSE which requires efficient parallelization. In this paper, we study how two well known platform for parallel programming, namely PVM and Linda, compare for designing a computation-intensive aerodynamics simulation program running on a cluster of networked workstations.

The program has been developed as a component of a distributed problem solving environment oriented to the domain of rotorcraft aerodynamics.

We compare the available programming environments for PVM and Linda in our domain from a software engineering point of view, namely we discuss how effective they are in the design phase of a distributed application with special requirements of load balancing and data allocation.

1 Introduction

In the recent years, the scientific and industrial communities have required more and more high performance computing resources. Currently the typical available hardware is a distributed computing environment which consists of autonomous computers of different computing power connected by high speed networks. Such distributed architectures are usually the result of a balance of computing power need, flexibility and generality of purposes, and economicity.

Programming a cluster of workstations is usually done using some platform like PVM [5] or MPI [3], which offer a library of primitives to design distributed applications and some tools to debug and tune them. Alternative platforms as Linda [15] are also available. In fact, sophisticated and specialized tools are necessary to the full exploitation of the available computing resources [10].

Ideally, a computing environment richly endowed with such tools might be defined a *Problem Solving Environment* (PSE) [4], insofar as it provides all the computing facilities to solve a well defined target class of problems, for instance in the field of aerospace design [14]. Numerical simulation is a major part of a PSE which requires much computing power and large memory space.

For the aeronautical industry, simulation tools are particularly convenient, being cheaper, faster, and safer than the real thing. The interest in this area mainly concerns the ability to face the problems related to a highly reliable prediction of the aircraft physical behaviour, so that all the production activities can benefit: design, production, flight testing, etc.

In this way, it is possible to reduce the number of expensive and time-wasting re-design loops usually imposed by the lack of integration among the different design phases, and at the same time to satisfy the requirements of flexibility and compliance with future trends.

From a computer science point of view, the best solution to these needs consists of building software tools which can be effectively used by the engineers to design new aircraft. This is of course not a trivial task, as there are at least two difficult issues:

- algorithms for scientific simulations are almost always very computational-intensive: hours or even days of computing may be needed for a single run;
- visualization of the results of the simulation algorithm can be not straightforward, particularly when tracking and steering are to be implemented (usually the user interface has to be especially designed to the application domain, in order to allow the desired information to be conveyed to the user [11]).

Two strategies are possible to solve the computational problem: finding fast and efficient algorithms, and implement them on parallel architectures. We have used both these approaches.

Actually, in our research project we have developed for aeronautical industry a prototype PSE useful for designing and simulating (parts of) an aircraft. This kind of problems need high-performance architectures to be solved in a reasonable time [14, 2]. We present here a mathematical model as well as the basic structure of the related numerical algorithm for the simulation of potential transonic flows using a boundary integral equation method. In such an algorithm there are two time consuming steps: the evaluation of all the influence coefficients matrix and the construction of the known vector terms at

each time step of the simulation. Matrices involved in the computation of non–linear terms can reach a dimension of $10^5 \times 10^5$ floating point elements. These matrices in general are not sparse, so an important issue is how to allocate their representation when using a cluster of workstations to run the programs that manipulate them. Luckily, both computing steps described above can be easily distributed to take advantage of parallel computing techniques since each subset of coefficients is completely independent from each other.

In order to build the prototype PSE we have used two different software platforms on the same hardware, namely a cluster of networked workstations: PVM and Linda. In this paper we describe our experience, and compare the usage of PVM and Linda in our application.

This paper has the following structure: Sect.2 discusses the basic algorithms and why they are expensive in terms of space and time. Sect.3 presents a PVM implementation; Sect.4 presents a Network Linda implementation. Sect.5 includes some performance results.

2 The numerical algorithm

The algorithm used in this work has been described in [13, 8, 9], where extensive mathematical details can be found. Here the mathematical model used for the simulation of the physical phenomenon is simply outlined.

The dynamics of a non-viscous, non-conducting, compressible fluid is completely described by the set of the Euler equations,*i.e.*, conservation of mass, momentum, and energy. Under the assumptions of isentropic and irrotational flow, the velocity field can be expressed in terms of a velocity potential function ϕ, such as $\mathbf{v} = \nabla\phi$ (see [13]). The resulting equation for the velocity potential ϕ appears here in the form of non–homogeneous wave equation,

$$\nabla^2\phi - \frac{1}{a_\infty^2}\frac{\partial^2\phi}{\partial t^2} = \sigma, \tag{1}$$

where a_∞ represents the speed of sound in the undisturbed flow.

The advantages of this approach are essentially due to the fact that the linear operator on the LHS of the equation is the D'Alambert one; thus, the linear compressibility effects are completely captured by the wave operator, and reduced to a boundary contribution in the final integral formulation [13, 12]. The term σ in Eq. 1 has the form

$$\sigma = \nabla \cdot \mathbf{b} - \frac{\partial \hat{b}}{\partial t}, \tag{2}$$

where

$$\mathbf{b} = \left(1 - \frac{\rho}{\rho_\infty}\right)\nabla\phi, \qquad \hat{b} = \frac{\rho}{\rho_\infty} + \frac{1}{a_\infty^2}\frac{\partial\phi}{\partial t}, \tag{3}$$

and the ratio between the local value of the density ρ and the value it assumes in the undisturbed flow, ρ_∞, is obtained from the Bernoulli's equation (which is a first integral of the momentum equation for barotropic fluids)

$$\frac{\rho}{\rho_\infty} = \left[1 - \frac{1}{h_\infty}\left(\dot{\phi} + \frac{\nabla\phi^2}{2}\right)\right]^{1/\gamma - 1}, \tag{4}$$

where h_∞ is the specific hentalpy, and γ is the ratio of the specific heats. Note that σ is non–linear in $\nabla\phi$ and is not negligible when $|\nabla\phi|$ approaches the value of the local speed of sound a (*i.e.*, the Mach number $M = |\nabla\phi|/a$ approaches 1). Furthermore, when $M > 1$ (transonic condition) the behaviour of Eq. 1 changes from elliptic to hyperbolic. This change is physically explained considering that in the supersonic conditions the pressure disturbances can not propagate upwind.

The boundary conditions complete the differential problem. These are: the impermeability of the surface S_B of a body moving within the fluid with velocity $\mathbf{v}_B$,

$$\frac{\partial\phi}{\partial n} = \mathbf{v}_B \cdot \mathbf{n}, \qquad \text{for } \mathbf{x} \text{ on } S_B, \tag{5}$$

and the vanishing of the potential at an infinite distance from the body surface, ($\phi = 0$ for $\mathbf{x} \to \infty$). In the case of lifting bodies (not considered here) additional boundary conditions are required on the surface of the wake, in order to take into account the convection of the zero–thickness layer of vorticity generated at the trailing edge of the body (quasi–potential flow [12]).

The boundary integral formulation of the differential model outlined above, is obtained as follows. Consider the adjoint problem of Eq. 1,

$$\nabla^2 G - \frac{1}{a_\infty^2}\frac{\partial^2 G}{\partial t^2} = \delta(\mathbf{x}_*, t_*), \tag{6}$$

where $\delta(\mathbf{x}_*, t_*)$ is the Dirac delta function in $\mathbf{x}_*, t_*$. The "initial" conditions and the boundary condition at infinity associated with the above problem are, respectively, $G(\mathbf{x}, \infty) = \dot{G}(\mathbf{x}, \infty) = 0$, and $G(\infty, t) = 0$. The solution of Eq. 6 is then

$$G(\mathbf{x}, \mathbf{x}_*, t, t_*) = \frac{-1}{4\pi r}\delta(t - t_* + \theta), \tag{7}$$

where $r = |\mathbf{x} - \mathbf{x}_*|$ and θ is the time required by the acoustic signal to travel from the source point $\mathbf{x}$ to the observation point $\mathbf{x}_*$. Eq. 7 represents the fundamental solution of Eq. 1. Multiplying Eq. 1 by G, Eq. 6 by ϕ, subtracting, integrating in time and over the entire domain $\mathcal{V}$, applying the Gauss theorem, and using the boundary condition at infinity for G and ϕ, and integrating with respect to time (taking into account the initial conditions on ϕ and G) yields

$$\phi(\mathbf{x}_*, t_*) = \iint\!\!\!\!\!\!\bigcirc_{\mathcal{S}_B} \left[G\frac{\partial \phi}{\partial n} - \phi\frac{\partial G}{\partial n} + \frac{\partial \phi}{\partial t} G \frac{\partial \hat{\theta}}{\partial n} \right]^\theta d\mathcal{S} + \iiint_{\mathcal{V}} G\,[\sigma]^\theta \, d\mathcal{V}, \tag{8}$$

where $[...]^\theta$ denotes evaluation at the retarded time $t = t_* - \theta$. In Eq. 8, the effects of the moving body on the value of the velocity potential at the point $\mathbf{x}_*$ at time t_*, are represented by a distribution of source and doublet singularities on the surface $\mathcal{S}_B$ with intensity proportional to the value of the potential and its normal derivative on the surface itself, plus a sources distribution in the fluid volume surrounding the body, with an intensity proportional to the value of the non–linear terms.

If $\sigma = 0$ (*i.e.*, in the linear case) and $\mathbf{x}_* \in \mathcal{V}$, Eq. 8 is an integral representation of $\phi(\mathbf{x}_*, t_*)$ as a function of ϕ, $\partial\phi/\partial n$ on $\mathcal{S}_B$. On the other hand, if $\mathbf{x}_*$ is on $\mathcal{S}_B$, Eq. 8 represents a compatibility condition between ϕ and $\partial\phi/\partial n$ on $\mathcal{S}_B$ for any function ϕ satisfying Eq. 1. Since $\partial\phi/\partial n$ is known from the boundary conditions, then Eq. 8 yields a boundary integral equation for ϕ. In the nonlinear case ($\sigma \neq 0$) the value of $\nabla\phi$ in $\mathcal{V}$ needs to be extracted by numerical differentiation in order to evaluate the distribution of σ. The integral formulation Eq. 8 presents two major advantages with respect to classical CFD methods based on the direct numerical solution of the original differential equations. These are: the boundary condition at infinity are automatically satisfied, thus the evaluation of the potential in $\mathcal{V}$ is required only where σ is not negligible (a small portion of the volume surrounding the body); the iterative procedure required for the convergence of σ takes advantage of the evaluation of nonlinear terms at retarded times. Indeed, σ is unknown only for those points for which $\theta < \Delta t$ (where Δt is the time step used in the numerical solution).

In order to solve numerically the problem, Eq. 8 is discretized using a zeroth–order formulation. The surface of the body $\mathcal{S}_B$ is divided into M elements, and the fluid volume $\mathcal{V}$ surrounding $\mathcal{S}_B$ into Q volume elements. The integral terms in Eq. 8 are approximated with the sum of the contribution of each single surface and volume elements. Two sets of collocation points are defined on $\mathcal{S}_B$ and in $\mathcal{V}$, as the centers of the surface and volume elements. In the following we indicate with $[.]_B$ and $[.]_V$ the arrays containing the values of the variables at these sets of points.

The value of the potential on the body surface is then obtained by the solution of the system

$$[\phi]_B = \mathsf{A}^{-1}\,[b]_B\,, \tag{9}$$

with

$$[b]_B = \mathsf{B}_B\,[\chi]_B + \mathsf{C}_B\,[\phi]_B + \mathsf{H}_B\,[\sigma]_V\,. \tag{10}$$

Once the solution on the body surface is known, the value of the potential in the field is simply obtained as

$$[\phi]_V = \mathsf{B}_V\,[\chi]_B + \mathsf{C}_V\,[\phi]_B + \mathsf{H}_V\,[\sigma]_V\,. \tag{11}$$

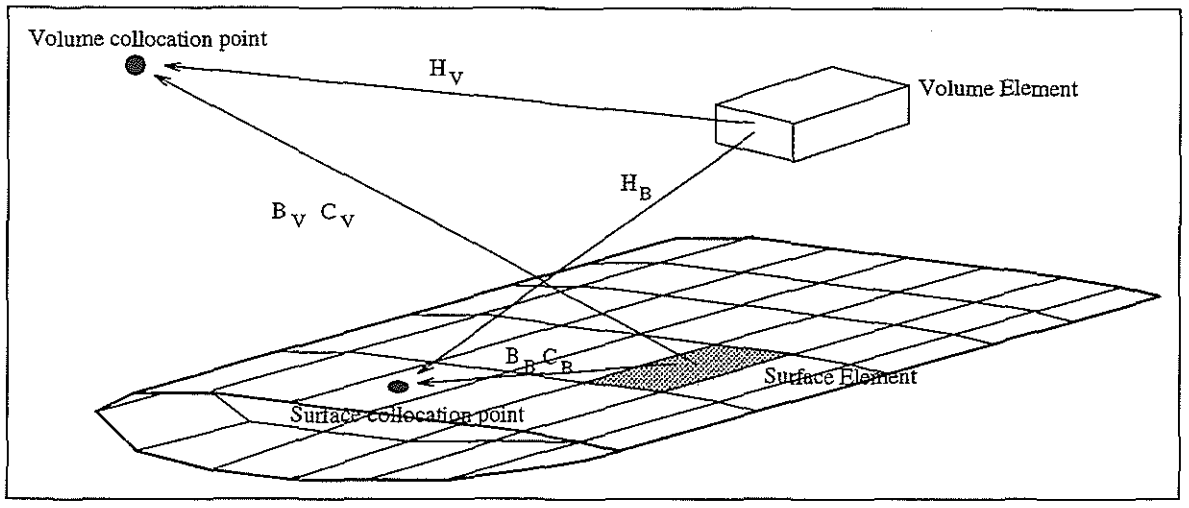


Figure 1: Scheme of the body–field influence system.

In the previous equations, the matrices B_B, C_B, H_B represent the influence of the singularity distributions on the potential on $\mathcal{S}_B$, whereas B_V, C_V, H_V play the same role for the potential in $\mathcal{V}$ (see Fig. 1). Note that the right hand sides of Eq. 9 and 11 are non–linear, thus the system has to be solved using an iterative procedure.

The formulation presented above has been widely validated in the past through comparisons with existing numerical results obtained with well assessed CFD methods (e.g. Iemma and Morino [8], [9], [6], [7]) and is being applied to two– and three–dimensional analysis of airplane wings and helicopter rotors in steady motion.

In the following, the matrices $B_B, C_B, F_B, H_B, B_V, C_V, F_V, H_V$ are called Influence Coefficients (IC) and depend only on the geometrical characteristics of the aerodynamical problem. The potential is calculated both on the surface of the discretized body ($[\phi]_B$) and in the discretized volume ($[\phi]_V$), in the form of arrays.

The temporal evolution of the system can be studied by iterating the solution procedure for different time steps.

Two phases are clearly defined as composing the whole algorithm: the construction of the IC matrices and the computation of the terms $[b]_B$ and $[\phi]_V$, needed for the time domain solution of the system (TDSL).

The suitability of a parallel implementation comes mostly from the IC matrices, in which each element can be independently calculated. These matrices are very large: for instance H_V can reach a dimension of $10^5 \times 10^5$ floating point elements. Hence, the design of a distributed implementation has to deal with both the computational load and memory requirements for data allocation.

3 A PVM approach

We will now illustrate an existing application based upon the above formulation and designed using PVM [5] as the message-passing communication package.

Message passing offers a straightforward way to implement parallel programs in distributed environments. However, simple as it is, message passing offers little comfort to the programmer: each coordination operation has to be implemented directly in terms of low level send/receive operations.

In our application the most difficult issues to deal with are the *size* of the data structures and the *minimization* of the communication overhead.

Data structures are accommodated in the multicomputer RAMs by partitioning them in pieces and assigning each section to a different host. This implies that each host has to perform all the computation relative to the section of data it holds. The partition of data is *static* because it is determined at the start of execution depending on problem size and on the number of worker processes. Once these parameters are fixed, it is not possible to change the amount of work processes are assigned to.

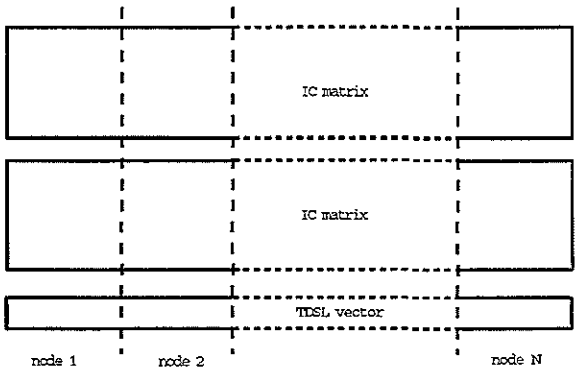


Figure 2: Data distribution. Different sections of the data structures are allocated on different nodes.

A collection of worker nodes is in charge of the computations on different sections of the data structures, given that no relation holds between different computed elements. This approach has the advantage of requiring null IPC in the influence coefficients phase because computation is done on local data only. Of course, in the TDSL phase communication is required to keep the potential vector updated in order to perform matrix-vector products.

3.1 Influence coefficients computation phase

Figures 3 and 4 show the pseudocode for the IC computation phase with a message passing approach, based on the static data partition concept discussed above.

```
for (i=1; i<=nproc; i++)
   recv(<i-th section of B, C and F matrices>) from i-th worker
compute (AINV)
bcast(AINV)
```

Figure 3: PVM approach. Pseudocode for the master process in the IC phase.

Computing work is carried out asynchronously by the workers on their respective sections of data. Sections are defined by the `first` and `last` local variables. When a worker ends computing its part, it performs a `send()` operation to the master, communicating its section of B_B, C_B and F_B. The master process collects the partial results, then builds and broadcasts the A^{-1} matrix with them. Data collection is a synchronous operation achieved by multiple `recv()`. IC computation ends when all the workers have received the A^{-1} matrix from the master.

```
for each IC matrix (B, C, F, H, BV, CV, FV and HV)
   first = first element in matrix section
   last = last element in matrix section
   for (i=first; i<=last; i++)
      compute (<i-th matrix element>)
send (<B, C and F matrices>) to master
recv(AINV) from master
```

Figure 4: PVM approach. Pseudocode for the worker process in the IC phase.

3.2 Time-domain solution phase

Figures 5 and 6 are pseudocodes for master and worker processes in the TDSL phase. As can be seen, they are structured similarly to those discussed for the previous phase.

```
for (t=0; t <= ntime; time++)
   for (iter=0; iter <= niter; iter++)
      for each TDSL vector V {
         for (i=1; i<=nproc; i++)
            recv(<i-th section of V>) from i-th worker
         bcast(V)
```

Figure 5: PVM approach. Pseudocode for the master process in the TDSL phase.

```
for (t=0; t <= ntime; time++)
   for (iter=0; iter <= niter; iter++)
      for each TDSL vector V {
         first = first element in vector section
         last  = last  element in vector section
         for (i=first; i<=last; i++)
            compute (V[i])
         send (V[first-last]) to master
         recv(whole V)
```

Figure 6: PVM approach. Pseudocode for the worker process in the TDSL phase.

Workers are free to compute their respective sections of solution vectors, one element at the time. When the `V[first-last]` section is complete, it is sent to master for collection and the worker process blocks on the `recv()` operation. The block condition is hold until the master process has collected all the vector sections and broadcast the whole vector to all the workers.

4 The Linda implementation

Languages for programming parallel systems have been defined and used for many years without significant changes in the basic communication primitives. Programming languages providing constructs for explicit parallelism are usually based on sequential processes and some set of synchronization and communication primitives. Two processes interact either through atomic operations on a shared resource (e.g., a semaphore or monitor), or by **send** and **receive** operations naming the peer process or an explicit channel over which the two processes communicate.

A third possibility is represented by the emerging class of parallel languages which conceptually are based on concurrent computations inside a shared data space of tuples. The main representative in this class is Linda [15].

The adoption of the Linda coordination model in designing the software architecture of the program provides a direct support for distributed data structures and agenda parallelism.

Agenda Parallelism is a way to coordinate parallel activities focusing the attention on simple sub-activities which compose the global work to be performed. An agenda of tasks representing these sub-activities is initially built, from where each computing agent is free to pick up one which can be carried on independently and in parallel.

In fact, *Agenda Parallelism* [1] is quite natural in Linda, so it is not surprising we have chosen it as the coordination strategy to implement our application. One of the most flexible realization of this paradigm is the so-called *master-worker* scheme, where a *master* agent is in charge of writing tasks in the shared agenda and *worker* agents pick and perform the tasks. This approach has the benefits of being automatically load-balancing because each worker can compute at its maximum rate, even in highly heterogeneous environments as time-sharing multi-user multicomputers.

4.1 Influence coefficient computation phase

The activities involved with this phase are:

- computation of the matrices of linear systems of equations B_B, C_B and F_B;

- computation of the matrix A^{-1} (AINV);
- computation of the matrices of non-linear systems of equations H_B, B_V, C_V, F_V and H_V;

```
while (END_BCF != getNewTask(&taskType, &row))
    out ("Task of IC", taskType, row);
rd (<B, C, and F>);
while (NOMORETASKS != getNewTask(&taskType, &row))
    tasks++;
    out ("Task of IC", taskType, row);
<compute AINV>;
out (<AINV>);
while (tasks--)
    in ("Task done");
out ("Task of IC", POISON, POISON);
```

Figure 7: Linda approach. Pseudocode for the master process in the IC phase.

In Figure 7 we show the pseudocode of the master process in the IC phase.

The parallel computation of the IC matrices is performed on a per-row basis, that is each task in the agenda indicates the computation of a single row of one of the matrices. A *bag of tasks* is created in the Tuple Space (TS) by means of tuples tagged `"Task of IC"`.

This process is split in two, generating all the tasks for the linear matrices first, then waiting for computed data to appear in the TS by means of appropriate `rd()` operations.

The computation of the non-linear matrices is done similarly, except that the master process collects from TS a number of token tuples tagged `"Task done"`, one for each task created. When all the tokens are collected, a *poison pill* tuple is output in order to terminate the phase. The master process also computes and outputs in TS the A^{-1} matrix while the workers are computing non-linear coefficients.

In Figure 8 we show the pseudocode of the worker process in the IC phase.

The process iterates until a `POISON` condition is found. First, a `"Task of IC"` is picked up from the bag of tasks in the TS and identification parameters for the task are assigned to local variables `taskType` and `row`. The `taskType` parameter tells the worker the kind of computation to perform and the `row` parameter indicates which row to compute. If a `POISON` task is not encountered, the worker performs actual computation depending on `taskType` and outputs results in TS, otherwise the termination condition is recognized, the poisonous tuple is reinstated in TS and the phase ends.

Computation of one linear task produces one `"Row of BCF"` tuple containing the influence coefficients. Computation for non-linear tasks comprehends one row of each of the non-linear matrices and is marked finished when a `"Task done"` tuple is output. In this case at least three tuples are created (four if the `row` parameter allows).

The algorithm used is highly parallel because no relation holds between any two tasks and this advantage is exploited with the "bag-of-tasks" approach: no relation between activities means that no restriction is imposed on the ordering of the activities, thus the highest possible degree of parallelism is achieved.

4.2 Time-domain solution phase

Once the matrices in the IC computation phase are computed, it is possible to solve the non-linear system of Eq. 9 in the TDSL phase.

Here, unlike in the IC computation phase, some small dependencies hold between the activities. In fact, each vectorial equation in the system has to be completely solved before the next equation can be solved and this creates three distinct sub-phases inside each time-step. Coordination is used in this phase to keep these computational constraints satisfied. However, inside each sub-phase it is still possible to coordinate the activities using the master-worker architecture and the bag of tasks data structure.

Computational loads have been found to be smaller with respect to the IC phase (each vector element requires fewer flop to compute), so that we assigned a larger granularity of parallelism. The choice here

```
while ( taskType != POISON )
    in ( "Task of IC", ? taskType, ? row );
    if ( taskType != POISON )
        switch (taskType)
            case BC:
                coefbBC(row);
                out ("Row of BCF", BC, row, <data>);
            case F:
                coefbF(row);
                out ("Row of BCF", F, row, <data>);
            case OTHERS:
                coefvBCV(row);
                out ("Row of IC", BCV, row, <data>);
                coefvFV(row);
                out ("Row of IC", FV, row, <data>);
                coefvHV(row);
                out ("Row of IC", HV, row, <data>);
                if (row < ncnb_w)
                    coefbH(row);
                    out ("Row of IC", H, row, <data>);
                out ( "Task done" );
    else
        out ( "Task of IC", POISON, POISON );
        rd  ( "AINV", ? ainv : );
        rd  ( <B, C and F> );
```

Figure 8: Linda approach. Pseudocode for the worker process in the IC phase.

has been to compute the potential vectors as distributed data structures made up of distributed *chunks* of elements. Each chunk of data is actually a section of a distributed vector and lives in TS as a tuple.

```
for (t=0; t <= ntime; time++)
    for (iter=0; iter <= niter; iter++)
        out (<tasks for rhs and phivu>);
        <synchronize with workers>
        out (<tasks for phiu, vb, vf, psiu and hkiu>);
        in  (<pressure coefficients cp>);
        in  (<velocities vb and vf>);
```

Figure 9: Linda approach. Pseudocode for the master process in the TDSL phase.

Figure 9 shows the pseudocode for the master process in the TDSL phase. The solution process is accomplished by the inner iteration loop in order to reach numerical convergence of the solution. The outer loop is needed in order to compute the solution in the time domain.

The two `out()` operations summarize the agenda-writing functions of the master process: for each vector it is created a bag of tasks, starting with $[b]_B$ and $[\phi]_V$. The following synchronization operation is needed in order to stop workers until at least a part of the bag of tasks are ready, as shown in the workers' code. At the end, the master process collects the computed results in form of pressure and velocity coefficients for output purposes.

The actual computation is accomplished in the worker code shown in Figure 10.

For each vector element to be calculated it is needed one row of the corresponding IC matrices in order to perform the inner product. Under the dynamical agenda paradigm, this implies that one `rd()` operation is needed in order to compute a single potential vector element because it is not possible for a worker to predict which task it is going to get next. Also, in the case where the TDSL phase iterates for multiple time steps, the IC rows are not "reused" for successive time steps computations

```
if (ICbuffering)
   rd (<hv matrix>);
for (t=0; t <= ntime; time++)
   for (iter=0; iter <= niter; iter++)
      <synchronize with master>
      compute(rhs, phivu);
      compute(phiu);
      compute(dpht, vb, vf, cp);
      compute(psiu, hkiu);
```

Figure 10: Linda approach. Pseudocode for the worker process in the TDSL phase.

and this may cause excessive IPC overhead. For this reason it has been added the option to buffer part of the IC data in the workers' local memory in order to eliminate multiple `rd()` operations for the same IC data over multiple time steps. This option significantly increases the memory requirements for the application, but allows better runtime performance. The bufferization option takes places at the very start of the TDSL phase, when the whole H_V matrix can be retrieved from TS. Actual computation starts with a synchronization with the master process, then the different subphases are worked out one at the time, creating the relative distributed data structures when completed.

Figure 11 shows the actual implementation of TDSL subphases. For example it is shown the pseudocode for the computation of the vector $[\phi]_V$ (in the code referred to as `phivu`). It is intended that each subphase completes the computation of a single vector by means of chunks.

```
while ( inp("phivu task", ? first, ? numItems) )
   in ("phivu", first, ? float * :);
   for (index=first; index<first+numItems; index++)
      if (!ICbuffering)
         rd (<index-th row of bv, cv, fv and hv>);
      compute (phivu[index]);
   out ("phivu", first, <numItems elements of phivu>);
```

Figure 11: Linda approach. Pseudocode for the worker process in the TDSL phase.

The implementation of the agenda paradigm in the TDSL subphases relies on the use of the `inp()` predicate. Workers reach in the bag of tasks while tasks are available and the termination condition is detected by means of the `inp()` operator itself. This implementation has been chosen because of its simplicity, not requiring any additional termination protocol like those used in the IC phase. The `first` and `numItems` variables are the identifiers of the chunk of vector to compute for the current task, representing the index of the first element to compute and the size of the chunk, respectively. The starting `in()` operation removes from TS the previous time step instance of the vector chunk, so that TS is not flooded with old tuples as time advances. Actual computation for each vector element is achieved in the `for()` loop, where the inner products are carried out. Here, any IC row is retrieved from TS via appropriate `rd()` operations, as long as the IC buffering option is disabled. When all the vector elements in the chunk are computed, a chunk tuple is output in TS, tagged with the name of the vector and the index of the first element contained. Concurrent computation of vector chunks builds the distributed vector.

5 Performance evaluation

We have tried to evaluate how well our agenda application performs when run in a time-sharing multi-user multicomputer composed of a cluster of workstations. For this purpose we have used a cluster of 15 SUN SPARClassics running on a standard Ethernet network.

While always running in non-dedicated mode, all the tests were performed during weekends or night periods.

Two issues have been investigated in our test work: performance and scalability of our Linda application and its load balancing capabilities. The comparison term for both measurements has been the PVM application.

The main performance metric adopted has been the *Hardware Performance* R_H [16] defined as:

$$R_H = F_H / M_{et}, \tag{12}$$

where F_H is the total number of flop and M_{et} is the *Master Elapsed Time* of the application. This metric is appropriate because the two applications ran on the same architecture.

The raw performance has been measured using the same aerodynamical problem with increasing dimension of state vectors, which corresponds to higher spatial resolution. Corresponding Mflop give a measure of the increase in computational workload. The multicomputer was composed of a maximum number of 15 nodes (master + 14 workers) and exactly one process (master or worker) has been placed on each node. The reported values are an average of at least three independent measures.

The results for increasing number of nodes in the IC computation phase are reported in Figure 12 for 9 and 15 nodes, respectively.

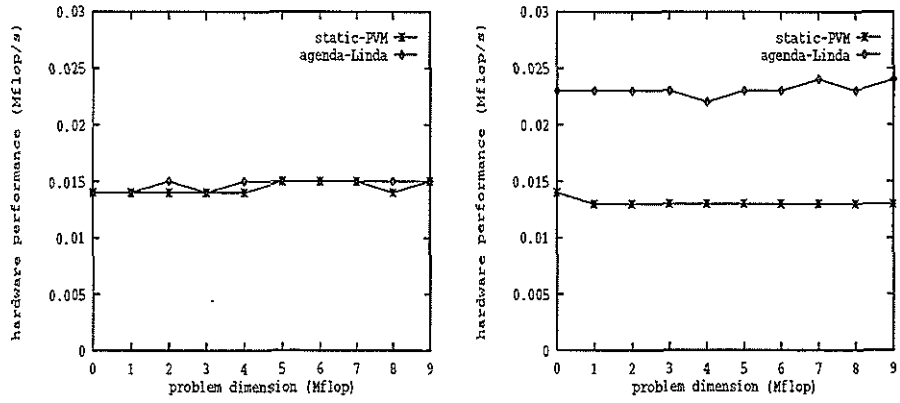


Figure 12: Hardware performances of the Linda and PVM applications for increasing problem dimension. Left: 9 nodes multicomputer. Right: 15 nodes multicomputer.

It is clear that the Linda application scales properly with the number of nodes, i.e.

$$[Mflop/s]_{9nodes} / [Mflop/s]_{15nodes} \simeq 9/15,$$

and is roughly twice as fast as the PVM application for every computational workload measured, when the number of nodes increases to the maximum number available.

The previous two tests demonstrate that virtual shared memory systems can be at least as efficient as message passing systems, which are reputed to be the best suited for distributed architectures.

In fact, our application has been found to run considerably faster when the number of nodes exceeds 10.

Figure 13 show the performance of the Linda and PVM applications when run over an aerodynamical problem in the time domain phase, with dimension about 2.5 Mflop. Data refer to clusters of 4 nodes (master + 3 workers, at left) and 9 workstations (master + 8 workers, at right), respectively. The increasing parameter on the abscissa is now the number of iterations used in each time steps.

Performance increases for both applications, but the Linda application clearly scales better.

In order to evaluate load balancing performance in non-dedicated environments we set up a test using a 4-nodes multicomputer (master + 3 worker). We ran a time-consuming external application on one of the workers' nodes in order to simulate a time-sharing multi-user environment. On the busy processor the CPU time has been measured to be equally shared between the worker process and the "interfering" application, so that the computational power of the node could be considered reduced by roughly 50%.

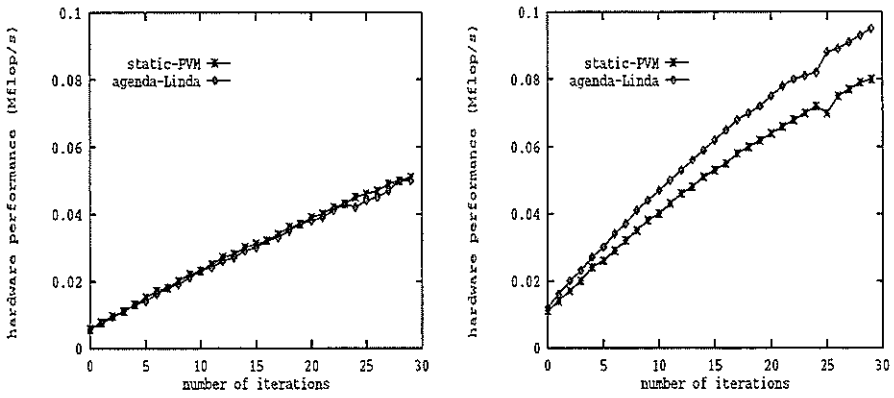


Figure 13: Hardware performances of the Linda and PVM applications for increasing number of time steps in the TDSL phase. Left: 4 nodes multicomputer. Right: 9 nodes multicomputer.

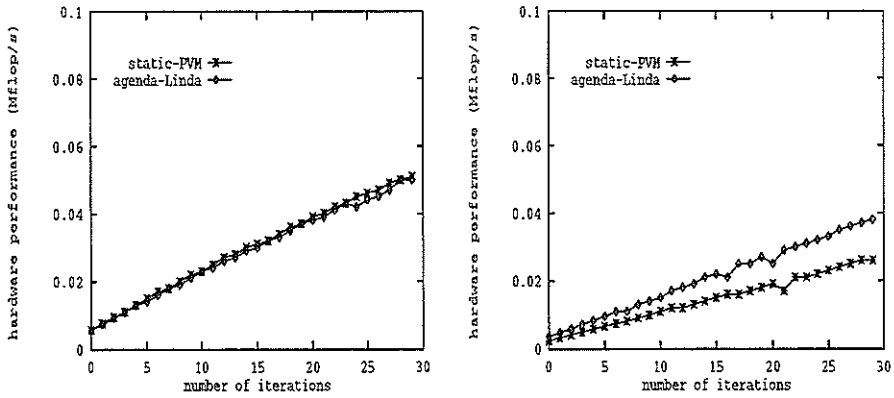


Figure 14: Load balancing. Hardware performances of the Linda and PVM applications for increasing number of time steps in the TDSL phase. At left, "dedicated" environment. At right, environment is loaded with an external application which reduces the computing capabilities of one of the nodes by about 50%. 4 nodes multicomputer.

Figure 14 shows the performance results in these conditions.

Figure at left (the same as in Figure 13) shows that the base performance of the applications is roughly the same when run in optimal conditions. However, optimal working conditions are not always possible in a cluster environment: interactive applications from other users can greatly vary how a multicomputer performs with respect to load balancing.

Our test shows that the application implemented with Linda shows a considerably better performance in this environment. In fact, the static work partition scheme cannot handle the added complexity of the computational environment and this causes the application to perform as slowly as the slowest node in the network. Instead, the automatic load balancing capabilities of Linda's tuple space produce performances comparable to the real aggregate computational power of the cluster and thus a smaller performance degradation.

The curves in Figure 14 reflect the expected values for the performance. The PVM application measured at right performed at about half speed with respect to that shown at left, because the whole computation proceeds with the speed of the slowest node in the network. Instead, the Linda application at right runs at about 70% the speed measured at left, thanks to the automatic load balancing capabilities of agenda parallelism when implemented by Tuple Space.

6 Conclusions

We have compared PVM and Linda in the design of a computation-intensive simulation program. Both platforms are well known, however lesser studied is their usage from a software engineering viewpoint.

The software engineering tools available for PVM and Linda influence the development costs. We have measured much shorter development times in the case of Linda. However, Linda was especially useful as rapid application development platform. We were able to perform several experiments rearranging the coordination of the main components of the program.

Our plans now include the full development of a problem solving environment devoted to aerodynamics applications. We are building this PSE around the main tools offered by the Linda programming environment.

References

[1] N. Carriero and D. Gelernter. *How to Write Parallel Programs.* MIT Press, 1990.

[2] C. Everaars and B. Koren. Using Coordination to Parallelize Sparse-Grid Methods for 3D CFD Problems. (to appear), 1998.

[3] MPI Forum. "MPI: a Message-Passing Interface Standard". *International Journal of Supercomputer Applications*, 8(3/4), 1994.

[4] E. Gallopoulos, E. Houstis, and J. Rice. Computer as Thinker/Doer: Problem Solving Environments for Computational Science. *IEEE Computational Science and Engineering*, 1(2):11–23, 1994.

[5] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *PVM: Parallel Virtual Machine. A User's Guide and Tutorial for Networked Parallel Computing.* The MIT Press, 1994.

[6] U. Iemma, V. Marchese, and L. Morino. High-Order BEM for Potential Transonic Flows. *Computational Mechanics*, 21:243–252, 1998.

[7] U. Iemma, V. Marchese, and L. Morino. Euler Flows via BEM: a Potential/Vorticity Integral Formulation. In *ICES 98 Conference*, in press.

[8] U. Iemma and L. Morino. "Transonic Analysis Using a Boundary Element Method". In 19^{th} *ICAS Conference Proceedings*, Anaheim, California, 1994.

[9] U. Iemma and L. Morino. Steady two-dimensional transonic analysis using a boundary integral equation method. *Journal of Fluids and Structures*, 11:633–655, 1997.

[10] O. Loques, J. Leite, and E. Carrera. P-RIO: A Modular Parallel Programming Environment. *IEEE Concurrency*, 6(1):47–57, 1998.

[11] A. Messina, L. Moltedo, S. Contento, and R. Nicoletti. "Cognitive Properties of Icons for Multidimensional Data Analysis". In V. Skala, editor, *Proceedings of the* 1995 *Winter School of Computer Graphycis and Visualization*, pages 197–208, 1995.

[12] L. Morino. Boundary integral equations in aerodynamics. *Applied Mechanics Review*, 46(8):445–466, August 1993.

[13] L. Morino and K. Tseng. "A General Theory for Unsteady Compressible Potential Flows with Applications to Aeroplanes and Rotors". In P.K. Banerjee and L. Morino, editors, *Boundary Elements Method In Nonlinear Fluid Dynamics, Developments in Boundary Elements Methods*, pages 183–245. Elsevier Applied Science, London, 1990.

[14] J. Murphy. A perspective of HPCN requirements in the European Aerospace Industry. *Future Generation Computer Systems*, 11(4-5):409–418, 1995.

[15] Carriero N., Gelernter D., Mattson T.G., and Sherman A.H. "The Linda Alternative to Message Passing Systems". *Parallel Computing*, 20:633–655, 1994.

[16] Hockney R.W. *The Science of Computer Benchmarking.* SIAM, 1996.

7. References

1. Bourtsev, B.N., "Aeroelasticity of Coaxial Helicopter Rotor", *Proceedings of 17th European Rotorcraft Forum*, Germany, Berlin, Sept. 1991.

2. Bourtsev, B.N., "The Coaxial Helicopter Vibration Reduction", *Proceedings of 18th European Rotorcraft Forum*, France, Avignon, Sept. 1992.

3. Bourtsev, B.N., Selemenev, S.V., "The Flap Motion and the Upper Rotor Blades to Lower Rotor Blades Clearance for the Coaxial Helicopters", *Proceedings of 19th European Rotorcraft Forum*, Italy, Como, 14 – 16 Sept. 1993.

4. Coleman, C.P., "A Survey of Theoretical and Experimental Coaxial Rotor Aerodynamic Research", *Proceedings of 19th European Rotorcraft Forum*, Italy, Como, 14 – 16 Sept. 1993.

5. Akimov,A.I., Butov, V.P., Bourtsev, B.N., Selemenev, S.V., "Flight Investigation of Coaxial Rotor Tip Vortex Structure", *ASH 50th Annual Forum Proceedings*, USA, Washington, DC, May 1994.

6. Акимов,А.И., Бутов, В.П., Бурцев, Б.Н., Селеменев, С.В., "Летные исследования и анализ вихревой структуры винтов соосного вертолета", *Российское Вертолетное Общество, Труды 1го Форума*, Россия, Москва, 20 – 21 Сентября 1994.

7. Bourtsev, B.N., Gubarev, B.A., "A Ka-115 Helicopter a New Development of KAMOV Company", *Proceedings of 22nd European Rotorcraft Forum*, Russia, Saint-Petersburg, 30 August – 1 Sept., 1995.

8. Bourtsev, B.N., Selemenev, S.V., "The Flap Motion and the Upper Rotor Blades to Lower Rotor Blades Clearance for the Coaxial Helicopters", *Journal of AHS*, No1, 1996.

9. Bourtsev, B.N., Kvokov, V.N., Vainstein, I.M., Petrosian, E.A., "Phenomenon of a Coaxial Helicopter High Figure of Merit at Hover", *Proceedings of 23rd European Rotorcraft Forum*, Germany, Dresden, 16 – 18 Sept. 1997.

10. Бурцев, Б.Н., Вайнштейн, И.М., Квоков, В.Н.,Петросян,Э.А.,"Феномен высокого коээфициента полезного действия соосных несущих винтов на режиме висения", *Российское Вертолетное Общество, Труды 3го Форума*, Россия, Москва, 24 – 25 Марта 1998.

11. Bourtsev, B.N., Koptseva, L.A., Animitsa, V.A., Nikolsky, A.A., "Ka-226 Helicopter Main Rotor – as a New Joint Development by KAMOV & TsAGI ", *Aviation Prospects 2000, International Symposium*, Russia, Zhukovsky Moscow Region, 19 – 24 August 1997.

12. Бурцев, Б.Н., Копцева, Л.А., Анимица, В.А., Никольский, А.А., "Несущий винт вертолета Ка-226 – новая совместная разработка фирмы КАМОВ и ЦАГИ", *Российское Вертолетное Общество, Труды 3го Форума*, Россия, Москва, 24 – 25 Марта 1998.

13. Kurt Gotzfried, "Survey of Tiger Main Rotor Loads from Design to Flight Test", *Proceedings of 23rd European Rotorcraft Forum*, Germany, Dresden, 16 – 18 Sept. 1997.

14. Bourtsev,B.N.,Guendline,L.J., Selemenev, S.V.,"Method and Examples for Calculation of Flight Path and Parameters While Performing Aerobatics Maneuvers by the Ka-50 Helicopter based on Flight Data Recorded Information", *Proceedings of 24th European Rotorcraft Forum*, France, Marseilles, 15 – 17 Sept. 1998.

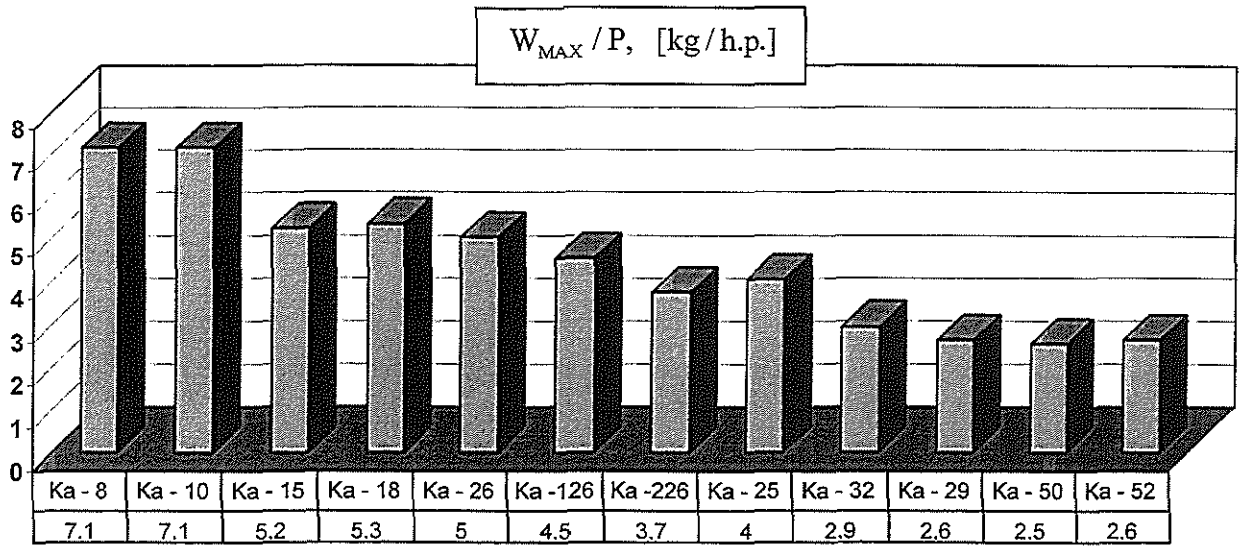
15. Mikheyev,S.V.,Bourtsev,B.N.,Selemenev, S.V., "Ka-50 Attack Helicopter Aerobatic Flight", *Proceedings of 24th European Rotorcraft Forum*, France, Marseilles, 15 – 17 Sept. 1998.

16. Mikheyev,S.V.,Bourtsev,B.N.,Selemenev, S.V., "Ka-50 Attack Helicopter Aerobatic Flight", *ASH 55th Annual Forum Proceedings*, Canada, Montreal, 25 – 27 May 1999.

17. Самохин, В.Ф., Ермилов, А.М., Котляр, А.Д., Бурцев, Б.Н., Селеменев, С.В., "Импульсное акустическое излучение вертолета соосной схемы при крейсерских скоростях полета", *Тезисы докладов на семинаре "Авиационная акустика"*, Россия, Дубна Московской области, 24 – 27 Мая 1999.

Basic Parameters of Coaxial KAMOV'S Helicopters

POWER LOADING



MAX SPEED

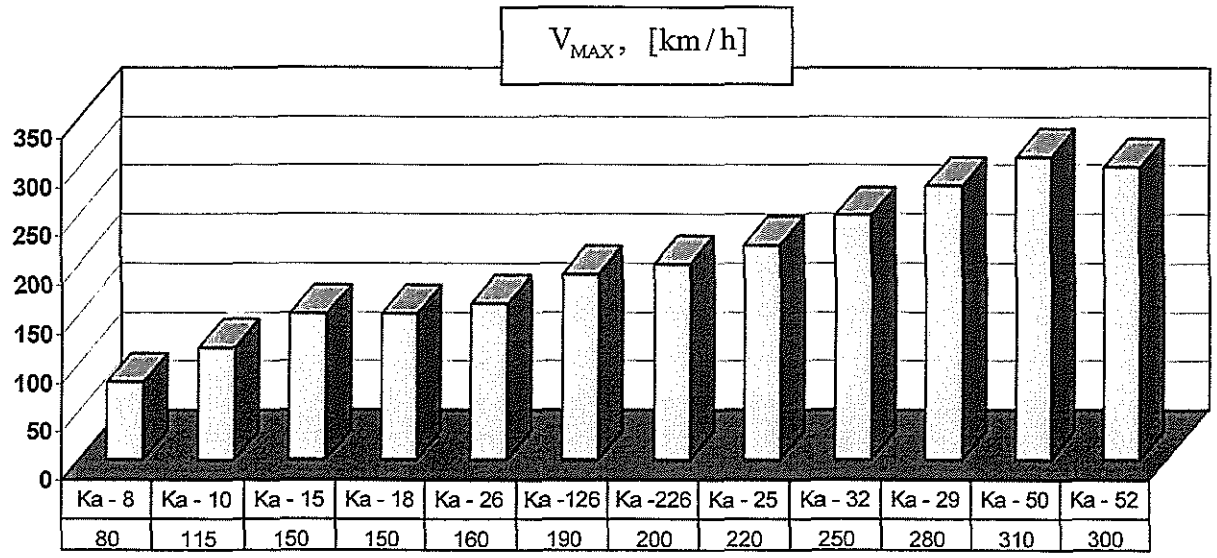
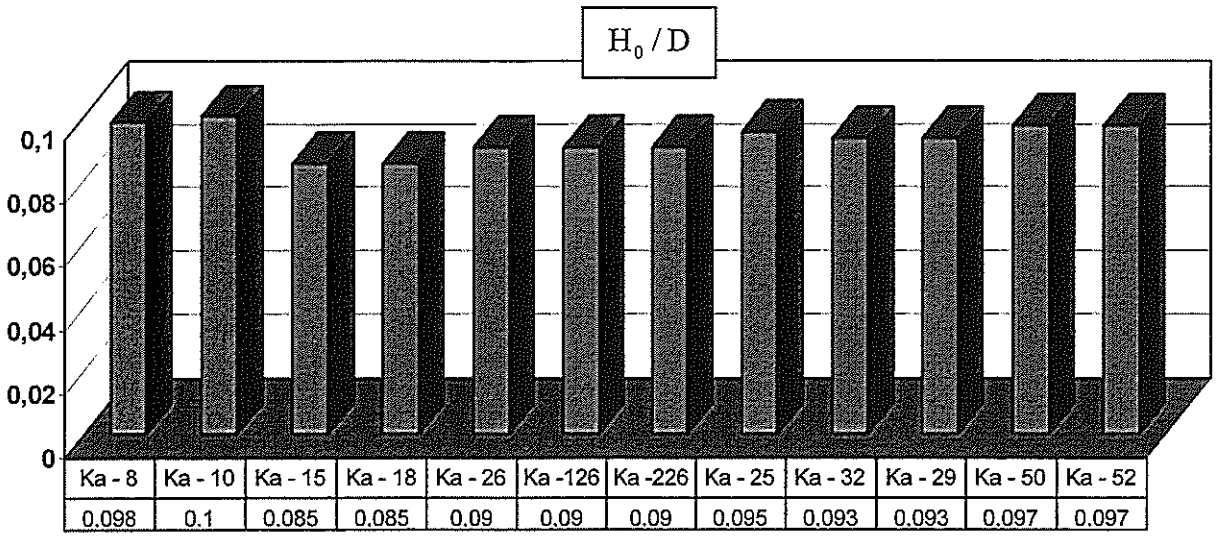


Fig.1A

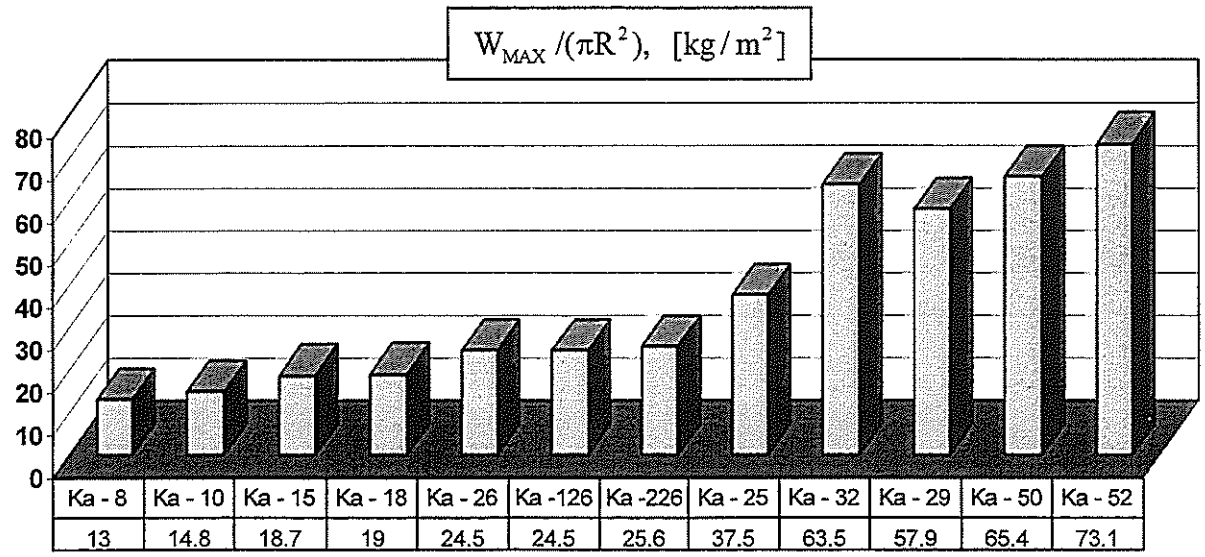
Basic Parameters of Coaxial KAMOV'S Helicopters

RELATIVE HUB CLEARANCES



Ka - 8	Ka - 10	Ka - 15	Ka - 18	Ka - 26	Ka -126	Ka -226	Ka - 25	Ka - 32	Ka - 29	Ka - 50	Ka - 52
0.098	0.1	0.085	0.085	0.09	0.09	0.09	0.095	0.093	0.093	0.097	0.097

DISK LOADING

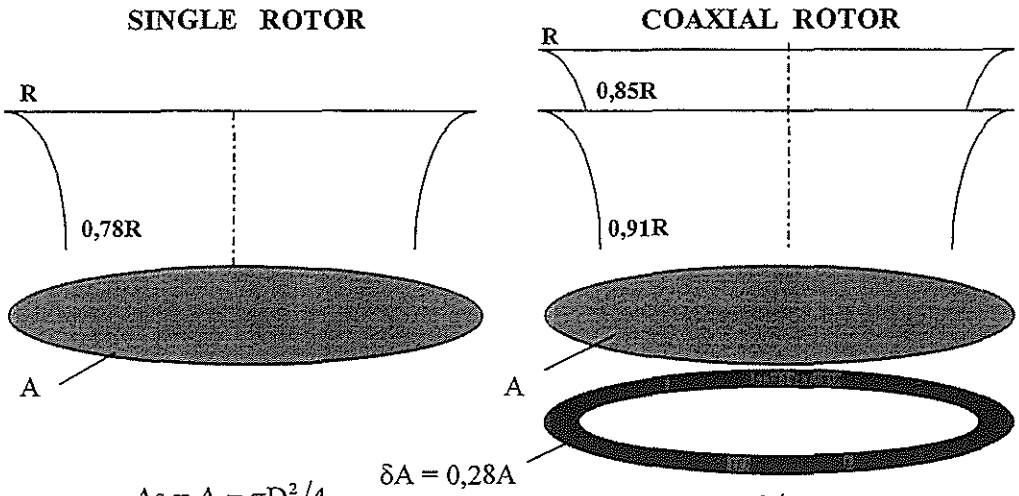


Ka - 8	Ka - 10	Ka - 15	Ka - 18	Ka - 26	Ka -126	Ka -226	Ka - 25	Ka - 32	Ka - 29	Ka - 50	Ka - 52
13	14.8	18.7	19	24.5	24.5	25.6	37.5	63.5	57.9	65.4	73.1

Fig.1B

Single & Coaxial Rotors

active disk areas , effective diameters , power & thrusts at hover



$As = A = \pi D^2/4$

$Ac = \pi D^2/4 + \delta A = 1{,}28\,As$

$D_{EFF} = \sqrt{4As/\pi} = 1{,}13\,D$

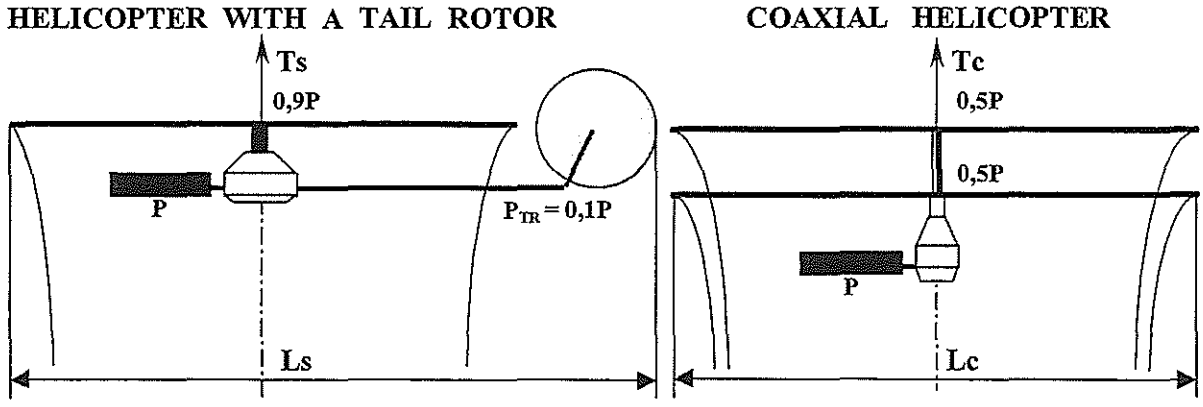
$Ts = (33{,}25 \cdot \eta_S \cdot D \cdot P\sqrt{\Delta})^{2/3}$

$Tc = (33{,}25 \cdot \eta_C \cdot D \cdot P\sqrt{\Delta})^{2/3}$

$$\eta_C = \frac{D_{EFF}}{D} \cdot \eta_S = 1{,}13\eta_S$$

Single & Coaxial Rotor Helicopters

main rotor diameter , power & thrust at hover



$Ts = \left(33{,}25 \cdot \eta_S \cdot Ds \cdot 0{,}9P\sqrt{\Delta}\right)^{2/3}$

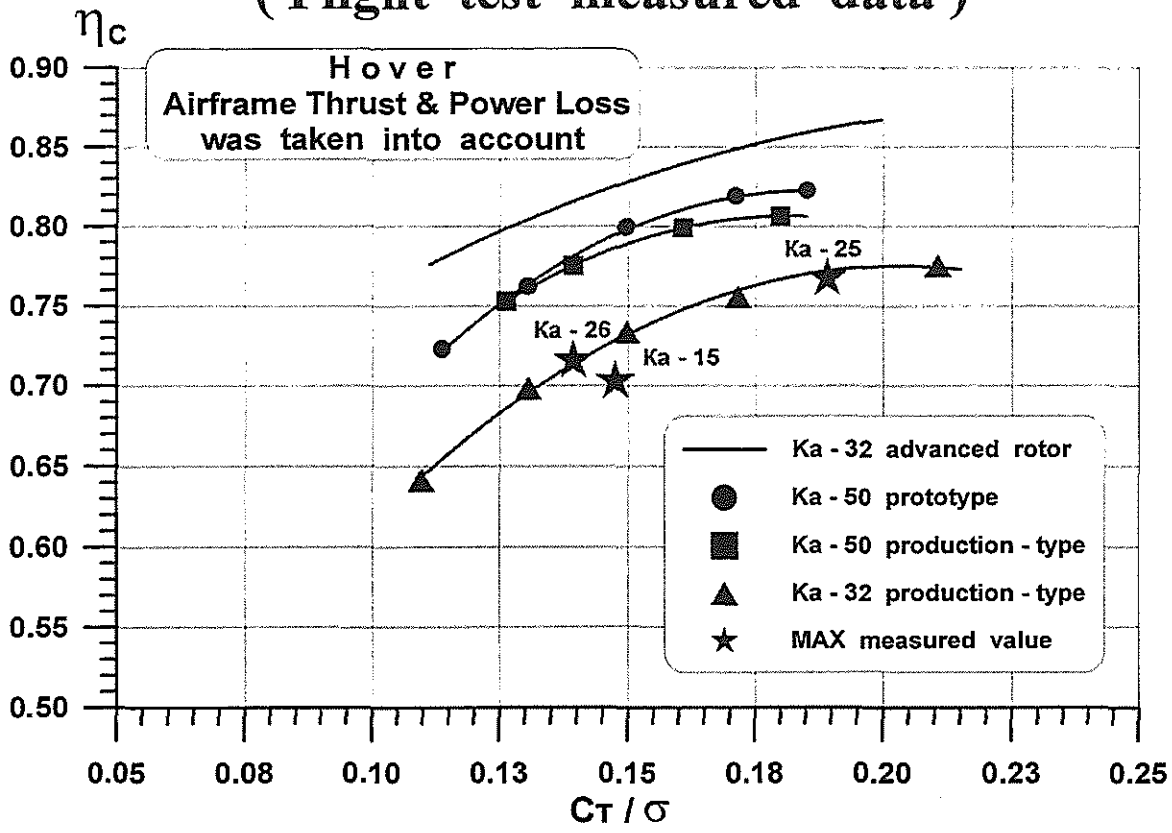
$Tc = \left(33{,}25 \cdot \eta_C \cdot Dc \cdot P\sqrt{\Delta}\right)^{3/3}$

From $\eta_c/\eta_s = 1{,}13$ and $Pc = Ps = P$ and $P_{TR} = 0{,}1P$:

1. At $Dc = Ds$ the thrust ratio is $Tc/Ts = (1{,}13/0{,}9)^{2/3} = 1{,}16$;
2. At $Ts = Tc$ the diameter ratio is $Ds/Dc = 1{,}13/0{,}9 = 1{,}26$.

Fig.2

Figure of merit of coaxial rotors
(Flight test measured data)



Wake form in hover & its approximation

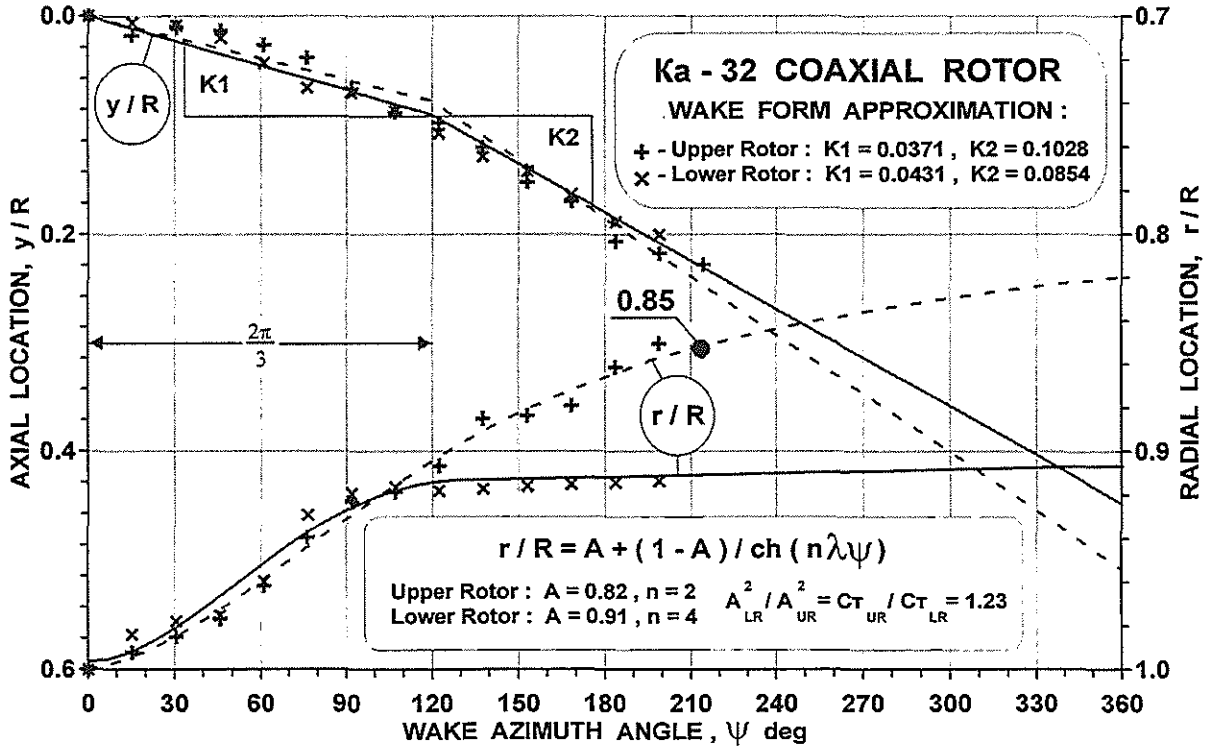


Fig.3

THE STATISTICAL CHART

Power Loadind - Disk Loading - Design Figure of Merit of Coaxial Helicopters & Helicopters with a Tail Rotor

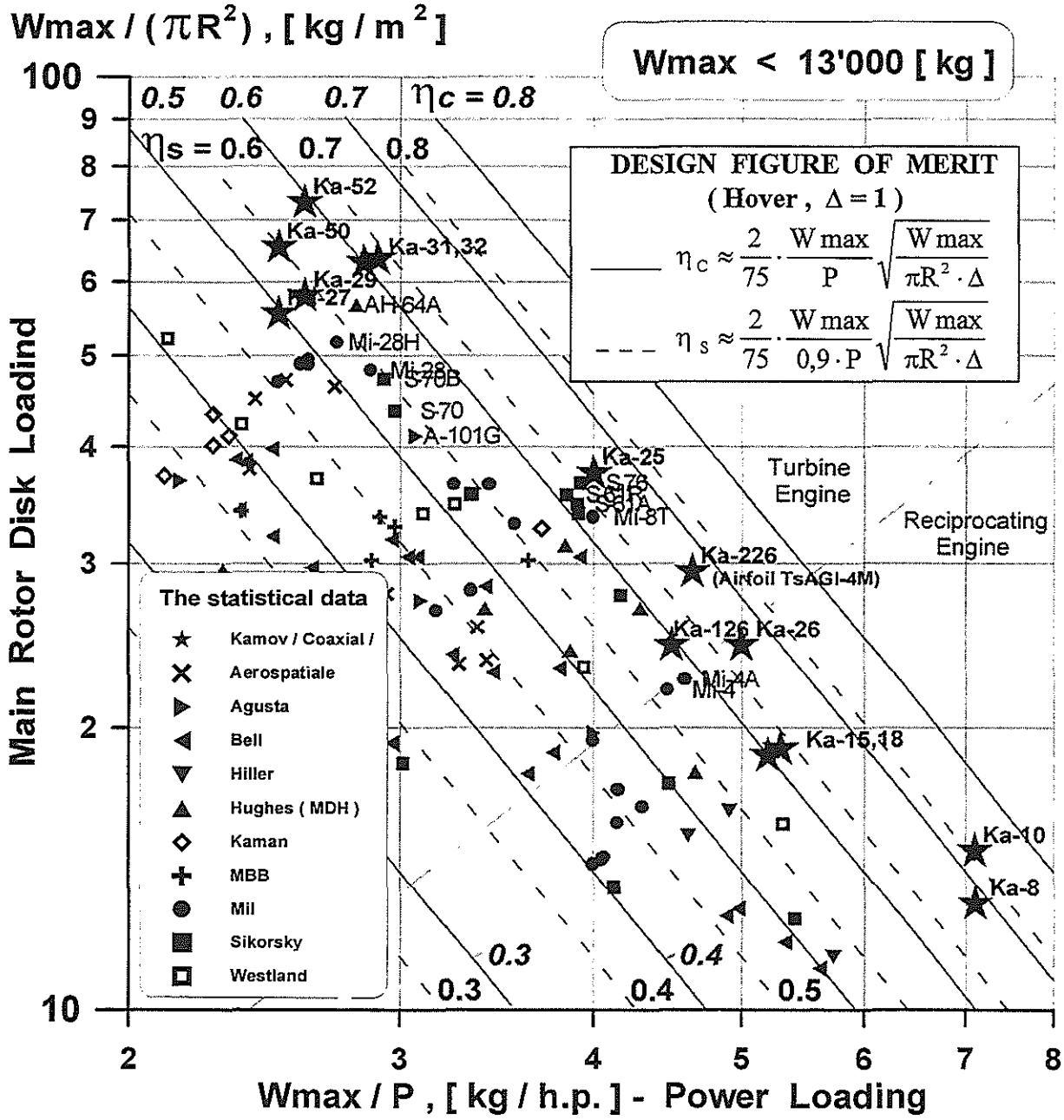


Fig.4

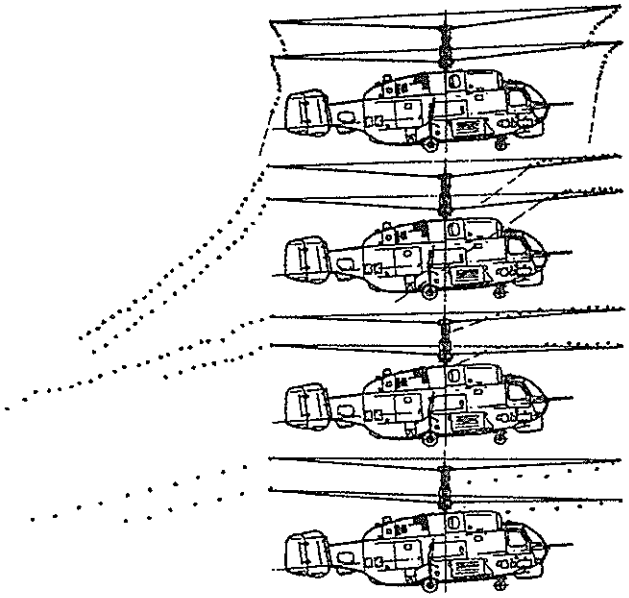
Coaxial Rotor Wake Side Views for Several Flight Speeds

V_{TAS} = 5 [km/h]

V_{TAS} = 73 [km/h]

V_{TAS} = 138 [km/h]

V_{TAS} = 227 [km/h]



Wake Front Boundary Longitudinal Position Versus Flight Speed

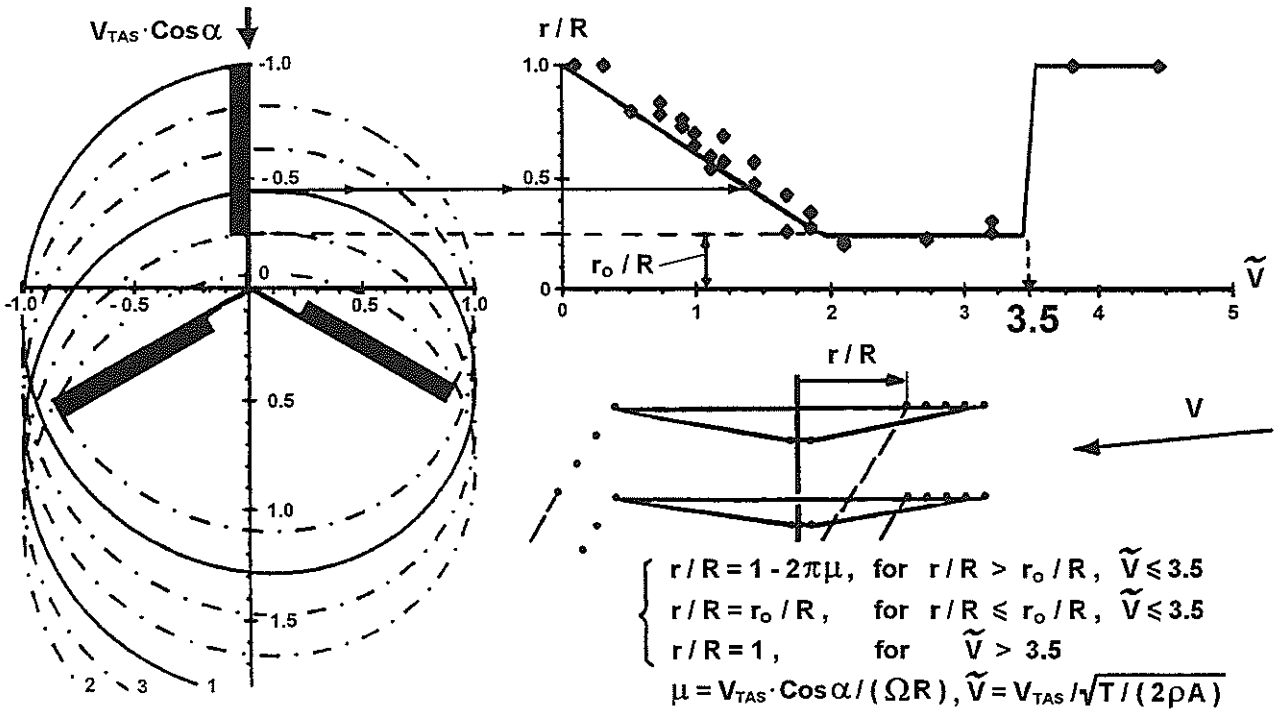


Fig.5

Simulated Aeroelastic Phenomena of Coaxial Rotor

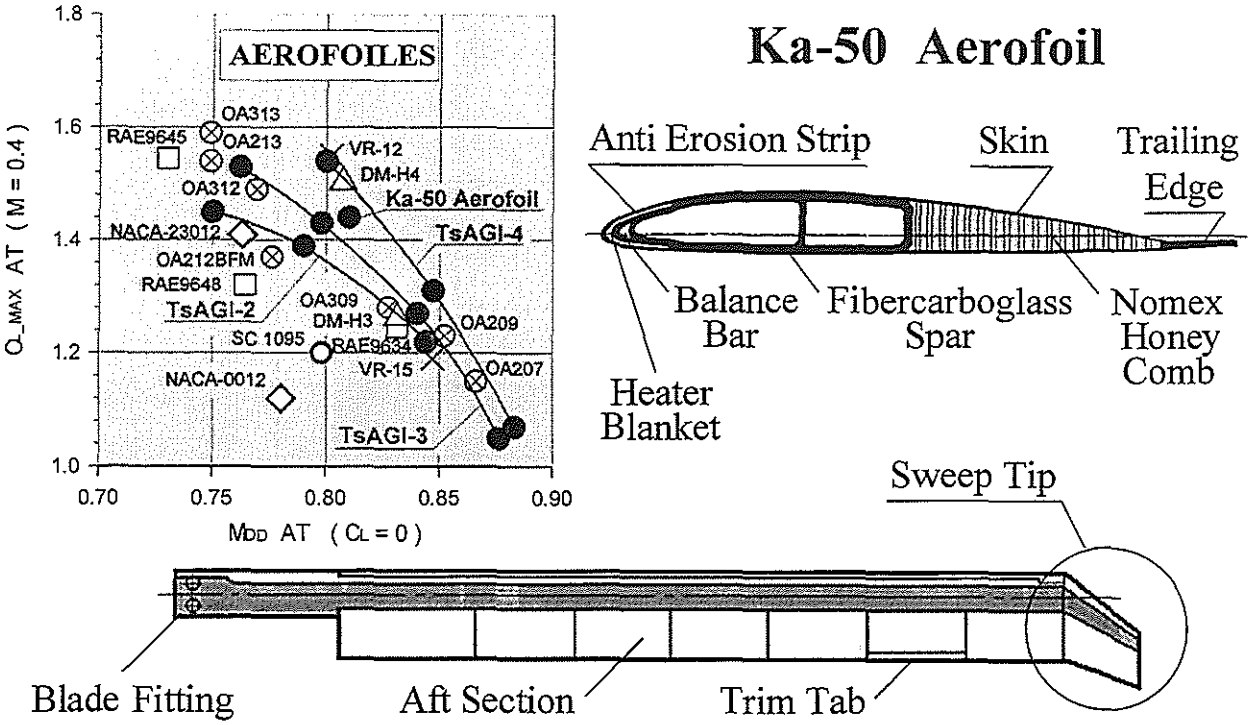
	Simulated Phenomena	SIMULATION VERSION				
		ULISS-6	ULISS-1	ULMFE	FLUT	MFE
1	$EI_x(r/R,\omega t)$ $EI_y(r/R,\omega t)$ $GI_p(r/R,\omega t)$	✓	✓			
2	$\bar{\varphi}_o=\|\|U_{ij}\|\|\times\bar{M}$	✓			✓	✓
3	$V_i(r/R,\psi)$	✓				
4	C_L, C_D, C_M $(\alpha, \dot{\alpha}, M, \dot{M})$	✓	✓	✓		
5	C_{L_MAX} $(\alpha, \dot{\alpha}, M)$	✓	✓	✓		
6	Airfoil Aeroelastic Deformation	✓	✓	✓	✓	
7	Upper/Lower Rotor Data	✓	✓	✓	✓	✓

Analysis Results of Coaxial Rotor Aeroelastic Simulation

	Analysis Results	SIMULATION VERSION				
		ULISS-6	ULISS-1	ULMFE	FLUT	MFE
1	Stall flutter boundary	Coaxial Rotors	Blade	Blade		
2	Bending moments, Pitch link loads, Actuator loads	Coaxial Rotors	Blade	Blade		
3	Elactic Deformations	Coaxial Rotors	Blade	Blade		
4	Alternate loads on Hubs	Coaxial Rotors				
5	Blade tips Clearances	Coaxial Rotors				
6	Flight test flutter	Coaxial Rotors	Blade	Blade		
7	Ground test flutter	Coaxial Rotors			Blade	
8	Natural frequencies			Blade		Blade

Fig.6

Blade Aerofoil Data



Ka-50 Aerofoil

Aerodynamic moments of existing TsAGI-2, TsAGI-4 airfoils & advanced TsAGI-4M airfoil

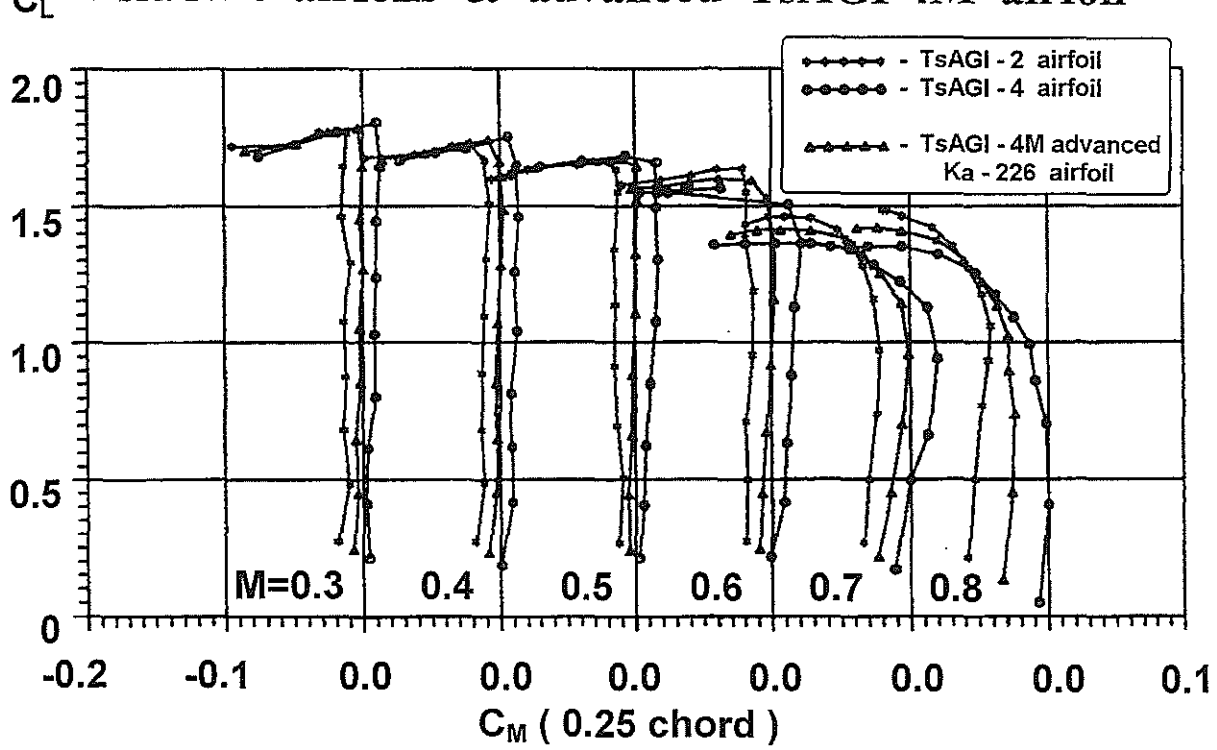
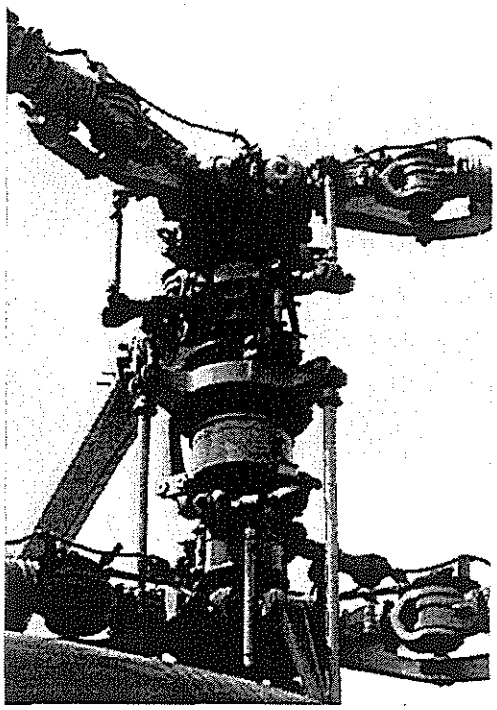
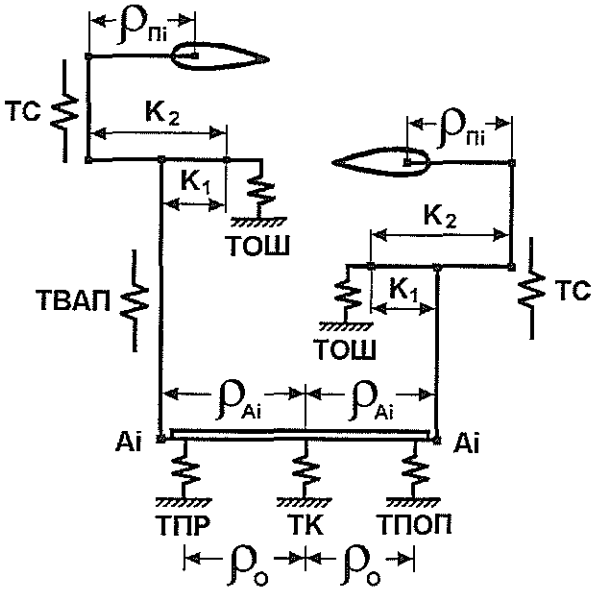


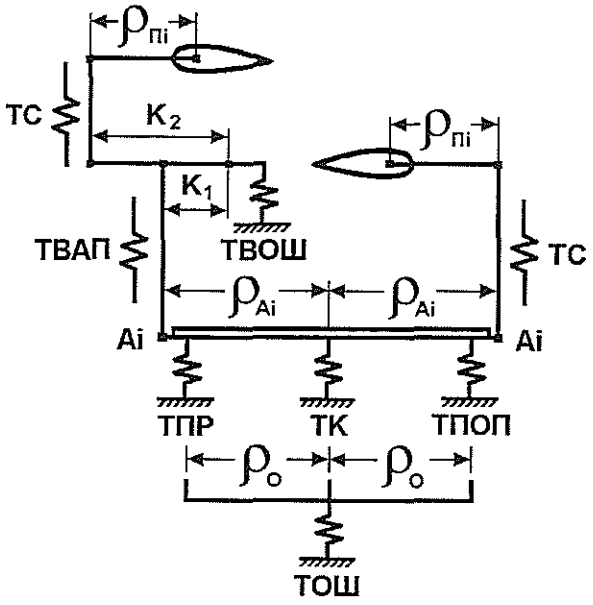
Fig.7

Ka-32 rotors control linkage model





Ka-50 rotors control linkage model



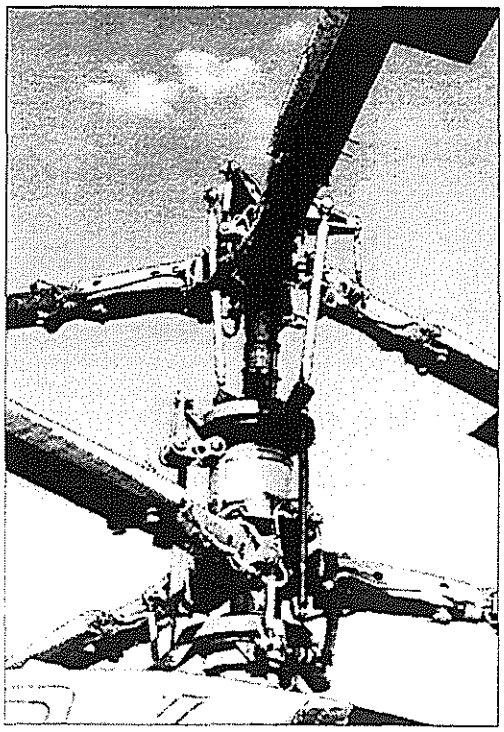
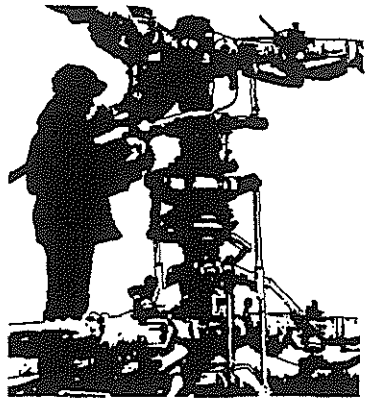
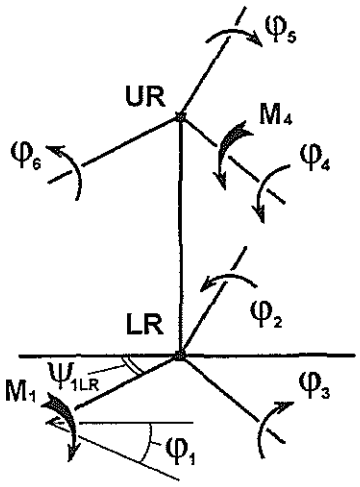


Fig.8

The scheme of experiment on determination of the control linkage elasticity matrix





$$\bar{\varphi} = \| \vartheta_{I,J} \| \times \bar{M}$$

$$\begin{vmatrix} \bar{\varphi}_1 \\ \bar{\varphi}_2 \\ \bar{\varphi}_3 \\ \bar{\varphi}_4 \\ \bar{\varphi}_5 \\ \bar{\varphi}_6 \end{vmatrix} = \left\| \begin{array}{ccc:ccc} \vartheta_{11} & \vartheta_{12} & \vartheta_{13} & \vartheta_{14} & \vartheta_{15} & \vartheta_{16} \\ \vartheta_{21} & \vartheta_{22} & \vartheta_{23} & \vartheta_{24} & \vartheta_{25} & \vartheta_{26} \\ \vartheta_{31} & \vartheta_{32} & \vartheta_{33} & \vartheta_{34} & \vartheta_{35} & \vartheta_{36} \\ \hdashline \vartheta_{41} & \vartheta_{42} & \vartheta_{43} & \vartheta_{44} & \vartheta_{45} & \vartheta_{46} \\ \vartheta_{51} & \vartheta_{52} & \vartheta_{53} & \vartheta_{54} & \vartheta_{55} & \vartheta_{56} \\ \vartheta_{61} & \vartheta_{62} & \vartheta_{63} & \vartheta_{64} & \vartheta_{65} & \vartheta_{66} \end{array} \right\| \times \begin{vmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \end{vmatrix}$$

Elasticity matrix

APPROXIMATION : $\vartheta_{I,J}(\psi_{1LR}) = f(TC, TK, TПОП, TПР, TВАП, TОШ, TВОШ, \psi_{1LR})$

CALCULATION :

$$\vartheta_{I,J}(\psi_{1LR}) = \frac{КАЧ_J}{\rho_{ПЛ}} \cdot \frac{КАЧ_I}{\rho_{ПЛ}} \cdot \{TK \cdot [КА_I \cdot (\mathrm{Sin}\varphi_I + \mathrm{Cos}\varphi_I) - 1] \cdot [КА_J \cdot (\mathrm{Sin}\varphi_J + \mathrm{Cos}\varphi_J) - 1] +$$
$$+ КА_I \cdot КА_J \cdot (TПОП \cdot \mathrm{Cos}\varphi_I \cdot \mathrm{Cos}\varphi_J + TПР \cdot \mathrm{Sin}\varphi_I \cdot \mathrm{Sin}\varphi_J)\} + TC + TВАП + \vartheta_{0\,I,J};$$

$$\vartheta_0 = \begin{Vmatrix} TОШ & 2TОШ \\ 2TОШ & 4TОШ + TВОШ \end{Vmatrix}; \quad \varphi_I = \begin{cases} \psi_{1LR} + ОП - \frac{\pi}{2} - \frac{2\pi}{K}(I-1); & I = 1, 2, \ldots, K \\ 2\pi - (\psi_{1LR} + ОП + DFI) - \frac{\pi}{2} - \frac{2\pi}{K}(I-K-1); & I = K+1, \ldots, 2K \end{cases}$$

Main rigidities of the elasticity matrix & dynamic rigidities obtained from frequency testing

$$\begin{cases} M\ddot{\varphi} + \vartheta^{-1}\varphi = 0 \\ \varphi = u \cdot e^{ipt} \\ (\vartheta M - E/P_K^2) \cdot u = 0 \\ P_K^2 = 1/(\lambda_K I) \end{cases}$$

MEASURED Ka – 32 ROTOR LINKAGE RIGIDITY ⇒

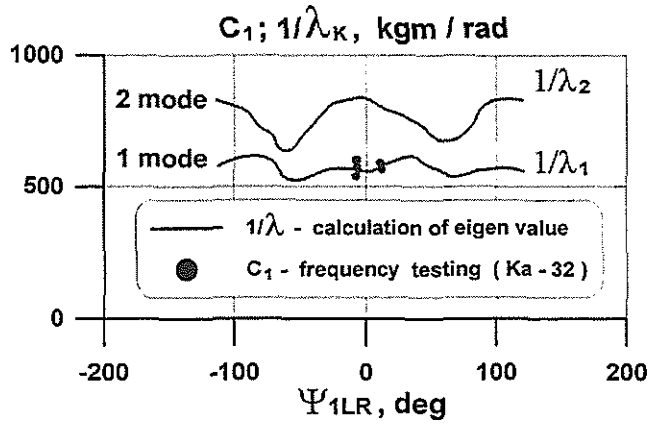
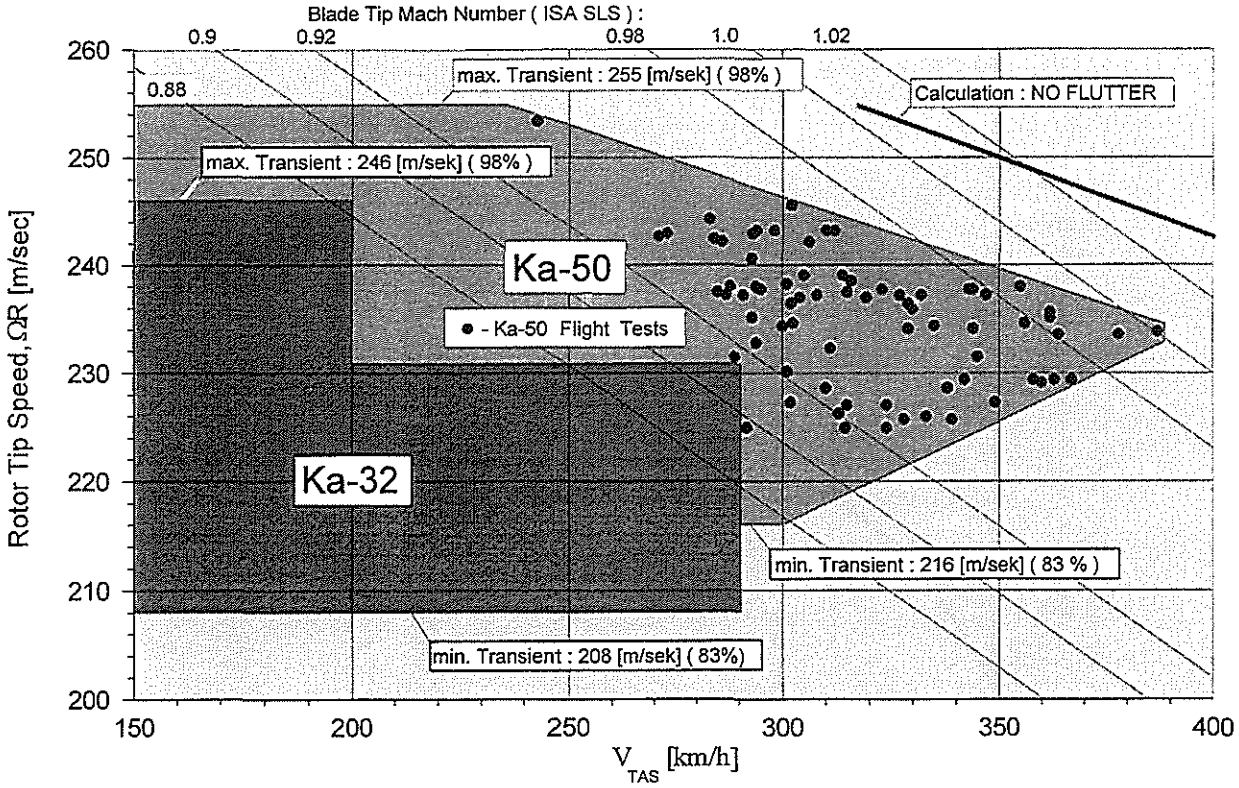


Fig.9

Demonstrated Rotor Speed Range



Stall Flutter Boundary

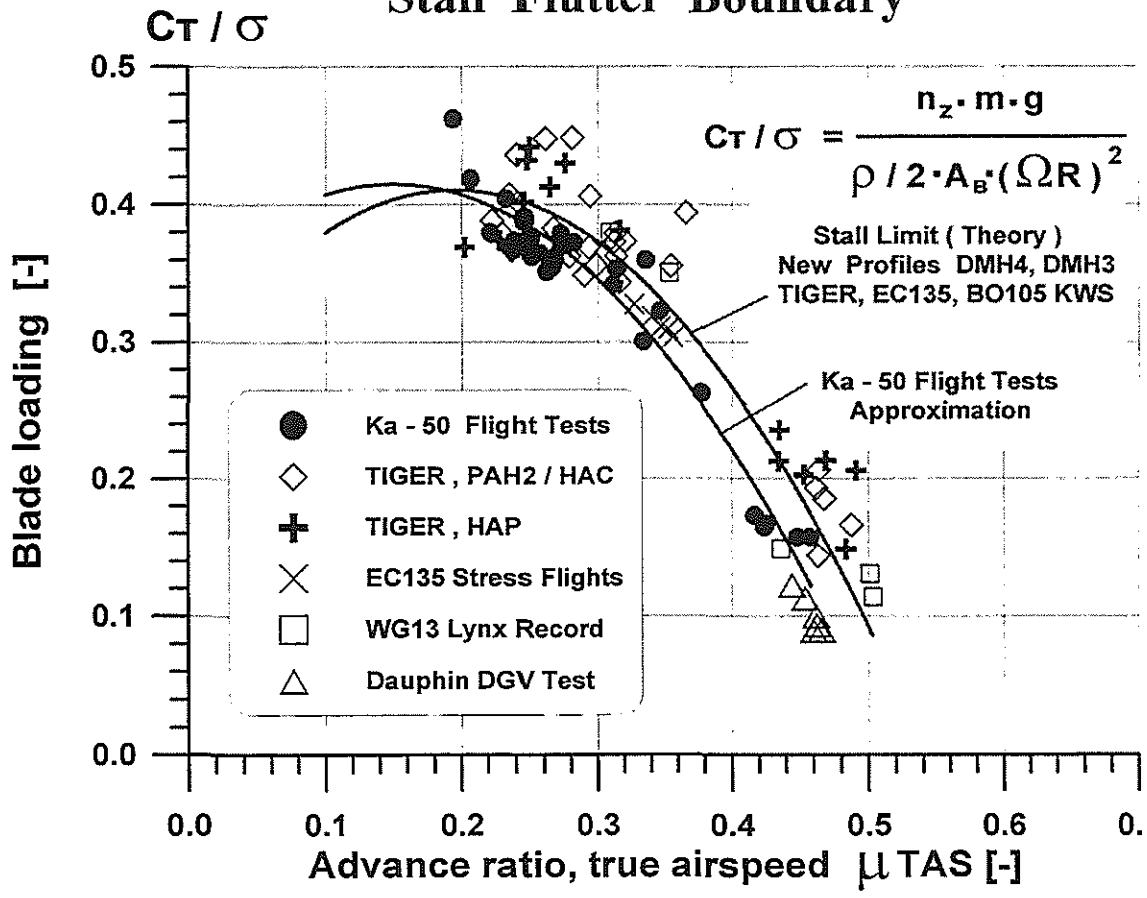


Fig.10

Ka - 25 Helicopter 3ω Vertical Vibrations

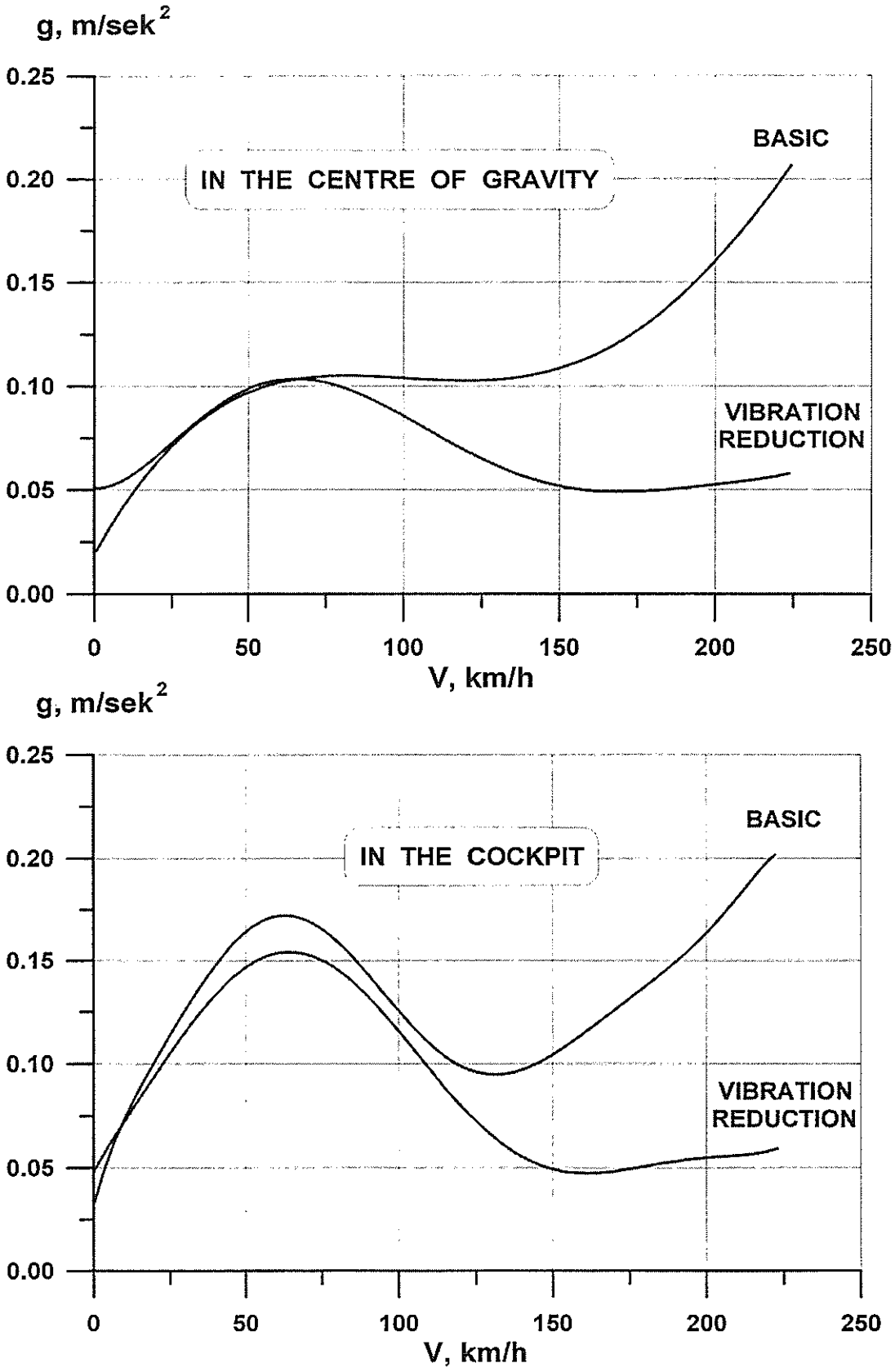
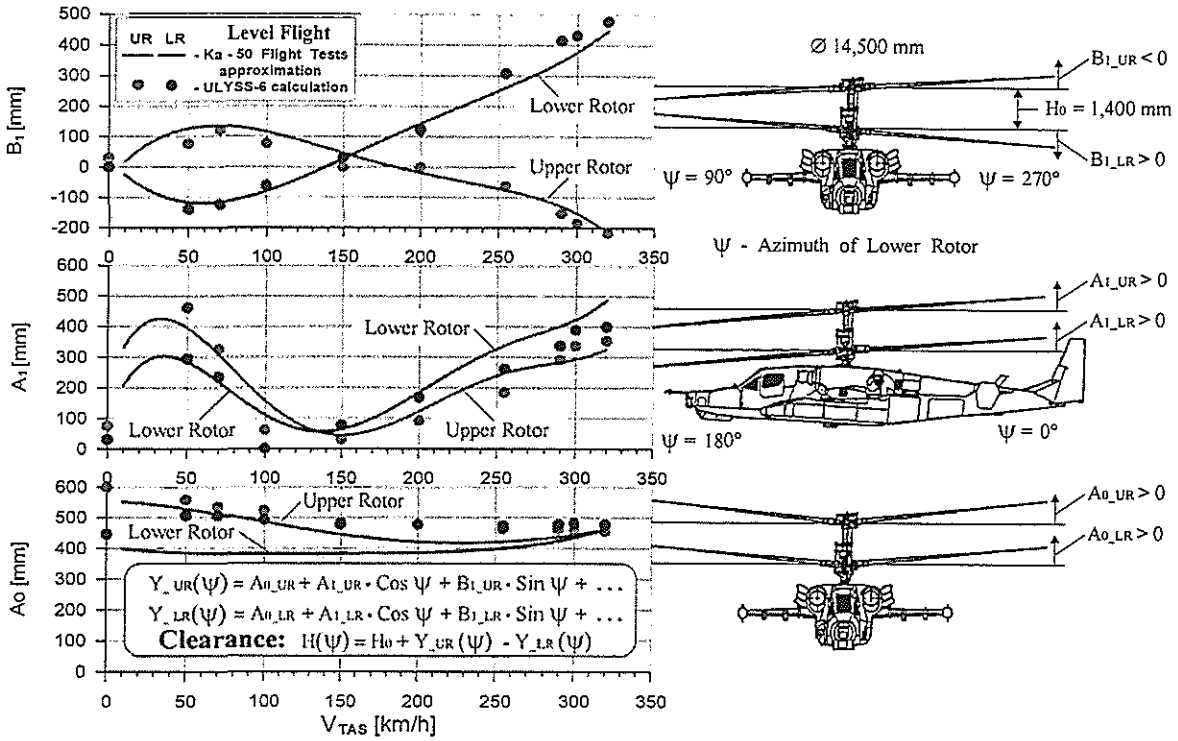


Fig.11

Blade Tip Coefficients

Comparision of Calculations and Flight Test Results



The Upper-to-Lower Rotor Blade Tips Clearancies

Versus Level Forvard Speed & Blade Azimuth

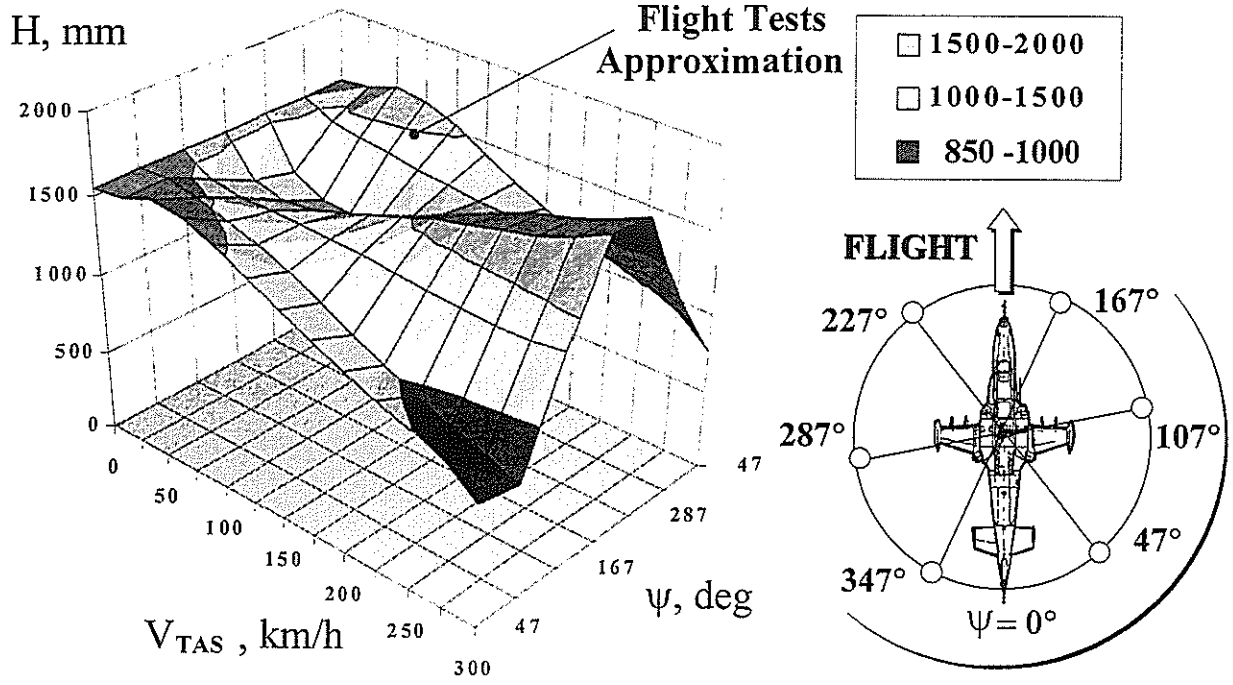
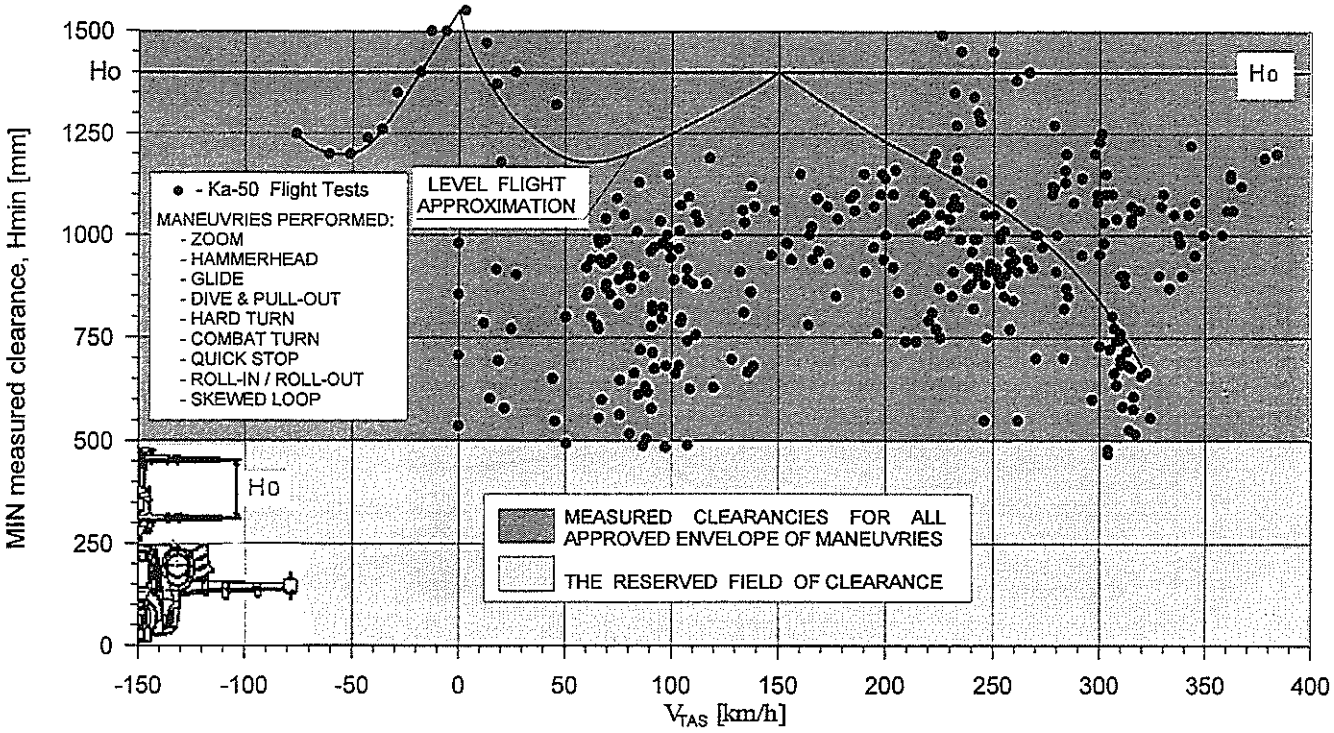


Fig.12

Measured Upper-to-Lower Rotor Blade Tips Clearancies



Load Factor / Speed Envelope

(Structural Qualification)

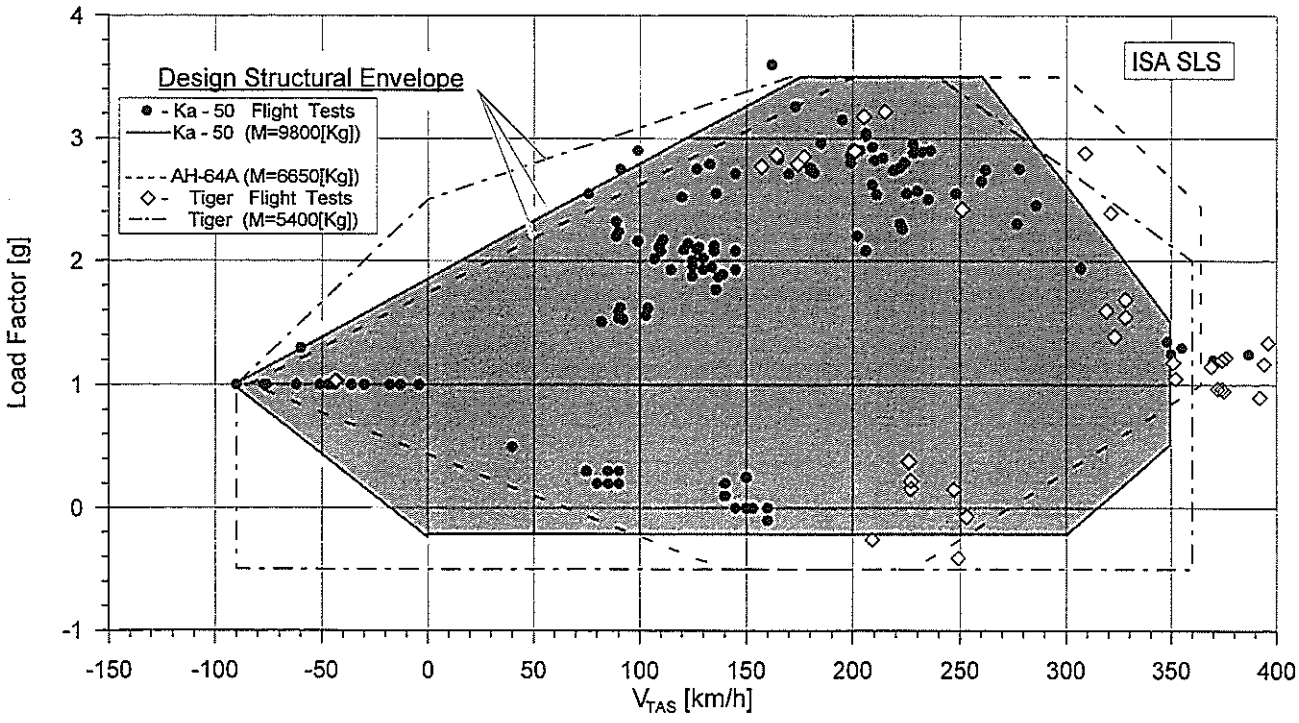


Fig.13

Ka-50 Aerobatic Maneuvries

MANEUVER	The Measured Parameter Values (min/max)				DESCRIPTION
	Airspeed V_{TAS} [km/h]	Load Factor [g]	Pitch attitude [deg]	Roll attitude [deg]	
Hard Turn (Right/Left)	280 ÷ 60	1.0 → 2.9 → 1.0	20 ÷ 50	0 ÷ -70	Unsteady Turn with Pitch & Roll
Flat Turn (Right/Left)	220 ÷ 0	1.0 → 1.5 → 1.0	±5	±20	Jaw Attitude ±80 ÷ ±90 [deg]
Hammerhead (Right/Left)	280 ÷ 0	1.0→ 2.9 → 1.0 → 2.9 → 1.0	0 ÷ ±90	±90	
Dive	0 ÷ 390	1.0 → 0.25 → 2.9 → 1.0	0 ÷ -90	±30	Push-Down, Dive & Pull-Out
Skewed Loop (Right/Left)	280 ÷ 70	1.0 → 2.9 → 1.2 → 3.5 → 1.0	0 ÷ 360	±150	
Quick Stop (Right/Left)	150 ÷ 40	1.0 → 2.0 → 1.2 → 1.0	0 ÷ 40	±55	Pitch / Roll Decceleration
Pull-Up with the Tail Forward	-90 ÷ 0	1.0 → 1.5 → 1.0	0 ÷ -70	±10	Backward Acceleration & Pull-Up with the Tail Forward/Up

Flight Path While Performing Skewed Loop

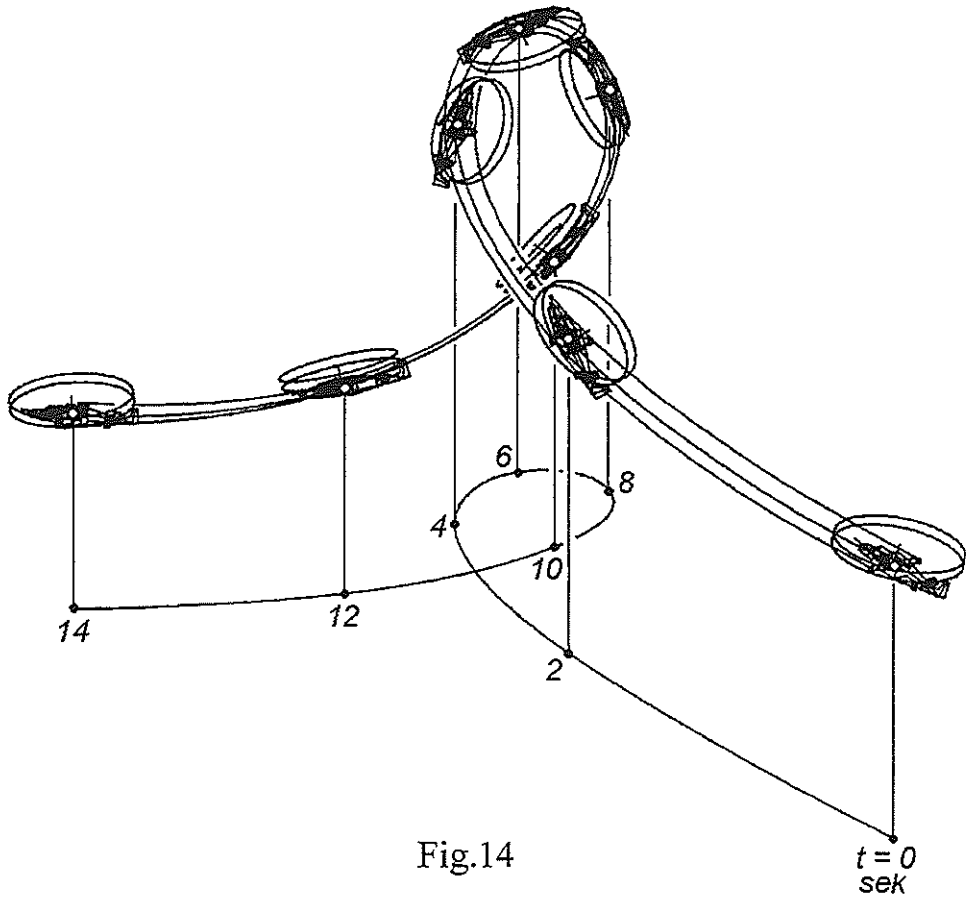


Fig.14

TWENTYFIFTH EUROPEAN ROTORCRAFT FORUM

Paper n° G23

HELICOPTER ACTIVE NOISE AND VIBRATION REDUCTION

BY

T. MILLOTT, W. WELSH
SIKORSKY AIRCRAFT, CT, USA

SEPTEMBER 14-16, 1999
ROME
ITALY

ASSOCIAZIONE INDUSTRIE PER L'AEROSPAZIO, I SISTEMI E LA DIFESA
ASSOCIAZIONE ITALIANA DI AERONAUTICA ED ASTRONAUTICA

HELICOPTER ACTIVE NOISE AND VIBRATION REDUCTION

Thomas A. Millott
Sr. Acoustics Engineer

William A. Welsh
Sr. Technical Engineer

Sikorsky Aircraft Corporation
Stratford, CT

1 ABSTRACT

High levels of noise and vibration continue to hamper the utility of helicopters. Lower cabin noise and vibration levels will reduce crew and passenger fatigue, resulting in fewer crew task errors, and thereby improving the mission effectiveness of helicopters. Reduced airframe vibration levels will lead to longer life spans for critical components, lower maintenance costs, and higher reliability. Although passive approaches for these problems have been implemented, they carry significant cost and weight penalties. In addition, passive technology has reached its limit in attaining further reductions within helicopter cost and weight constraints. The innovative use of active control technology provides the potential to reduce noise and vibration levels below those currently achievable with purely passive approaches, or alternatively, to achieve reductions comparable to passive approaches but with a lower weight penalty. Sikorsky has developed and flight tested active noise and vibration control systems. A prototype active vibration control (AVC) system has been flight tested on an UH-60 aircraft and achieved significant reductions in the main rotor 4P vibrations felt inside the helicopter cabin and cockpit. A productionized version of the AVC system is currently undergoing development flight testing on the Sikorsky S-92 Helibus™. Also, an active noise control (ANC) system has been successfully flight tested on an S-76 aircraft and achieved tonal noise reductions of up to 20dB in the helicopter cabin.

2 INTRODUCTION

Active Noise Control

Interior noise is an increasingly important discriminator in the commercial helicopter market, with "acceptable" noise levels traditionally being achieved passively, albeit with substantial weight penalties. Increasing performance demands (i.e., longer range, higher payloads) have driven the pursuit of lighter-weight solutions. Furthermore, continuing reductions in the noise levels commonly experienced by passengers in various other modes of transportation, including ground vehicles and commercial fixed-wing aircraft, have heightened the awareness and sensitivity of passengers to helicopter internal noise. In the last decade the continuing trend towards cheaper, faster, and more powerful computers has lead to the evolution of active noise control (ANC) from a laboratory experiment to a practical approach for reducing aircraft cabin noise levels (Ref. 1-2). ANC, properly integrated with traditional passive techniques, offers substantial promise of reducing helicopter cabin noise levels with lower weight penalties than purely passive treatments. This will benefit the helicopter industry by improving commercial acceptance and expanding the helicopter market.

There are three primary components of helicopter interior noise: (1) large amplitude, low frequency rotor harmonics; (2) broadband noise; and (3) higher frequency structure-borne tones generated within the gearbox, power train, and hydraulic system. A typical spectrum is shown in Fig. 1. The low frequency tones (<200Hz) are less important to passenger comfort than this figure would imply due to the natural attenuation of the ear at low frequencies. The high frequency gear-mesh tones (>700Hz) fall into the speech interference range and, since these tones generally rise far above the broadband noise floor (Fig. 1), are generally considered the most intrusive and irritating component of noise in a typical helicopter. The primary gear-mesh tone generally falls in the frequency range of 700-1000Hz, and is typically much louder than its higher harmonics. It is this tone which is typically the most irritating to passengers and crew, and unfortunately falls into a frequency range in which traditional passive treatments are not very weight efficient.

Recently, a multi-year program was initiated at Sikorsky Aircraft Corporation to develop a flight-worthy ANC system to actively cancel gear-mesh noise inside the cabin interior. Initial concepts explored for the helicopter gear-mesh ANC problem included both structural acoustic control of accelerometers/microphones using force generating actuators and acoustic control of microphones using speakers. However, the speaker/microphone control approach that has proven successful in lower frequency tonal applications such as turbo-prop aircraft (Ref 1) and for the low frequency rotor harmonics in helicopters (Ref 2) is not practical for this problem due to the high frequencies and resulting number of participating acoustic modes. In this approach, the speakers must set up a sound field inside the cabin that matches and cancels that created by the disturbance. For global noise reductions this requires that the number of speakers be at least equal to the number of relevant acoustic modes (Ref 3). There are several hundred acoustic modes present in the gear-mesh frequency range (>700Hz) for a cabin the size of the S-76 (Ref. 4), thus requiring at least as many speakers to achieve global noise reductions.

A more promising approach for the gear-mesh ANC problem is to use a choke-point methodology to prevent the structure-borne gear-mesh vibrational energy from entering the cabin by placing actuation where the gearbox is mounted to the structure. The approach described in this paper, and flight tested on the S-76 helicopter (Ref. 5), involves the use of point force (proof-mass) actuators surrounding the gearbox mounts to actively cancel the gear-mesh vibrations before they enter the airframe. This approach avoids interrupting the primary load path of the helicopter, thus avoiding flight safety concerns, and is generally more effective with a wider frequency range of operation than tuned passive absorbers. A schematic of the gear-mesh ANC approach is shown in Fig. 2. The approach taken in Refs. 6-7 to address helicopter gear-mesh noise used structural actuators mounted parallel with struts supporting the gearbox to introduce the canceling forces directly into the load path. However, in the case of the S-76, the mechanical interface between the gearbox and the airframe is significantly different, in that the gearbox is bolted directly to a pair of transmission beams integral to the airframe structure. Thus the actuators were mounted on the transmission beams near the gearbox mounting points as in Ref. 8. The gear-mesh ANC system utilizes microphones distributed throughout the cabin as feedback control sensors, as shown in Fig. 2.

Active Vibration Control

Helicopter vibration is largely a result of vibratory aerodynamic loads generated by the rotor as it moves edgewise through the air in forward flight. Traditional passive approaches to reducing vibrations involve the use of tuned-mass absorbers. These absorbers tend to be very heavy, are only effective in a very narrow band about the tuned frequency, and generally only

reduce vibrations in close proximity to their mounting location. Furthermore, passive systems appear to be approaching an asymptotic limit in terms of weight efficiency. A major leap forward in terms of passive vibration reduction in helicopters was the introduction of the hub absorber. The hub absorber, which reduces the inplane NP (blade passage frequency) vibratory loads at the hub, results in significant reductions in the NP vibrations in the cockpit and the cabin. An example of the magnitude of vibration reduction provided by the hub absorber is shown in Fig. 3 for the Sikorsky S-61. As described in Ref. 9 and also shown in Fig. 3, significant reductions in the failure rates of critical components were observed in connection with the lower vibration levels, which ultimately translated into reduced maintenance costs.

Since the introduction of the hub absorber, however, the vibration levels of modern helicopters appear to be hovering about an asymptotic limit of 0.10-0.15 g. This is due to the fact that the hub absorber does not affect the vertical vibratory hub load component that propagates unattenuated into the airframe. This component is generally attacked in Sikorsky helicopters using tuned passive absorbers located in the cabin. These devices are generally very heavy and incur significant weight penalties. Lower vibration levels can be achieved by increasing the number of passive absorbers, but this tends to result in diminishing returns. Thus modern helicopters appear to be approaching a vibration limit driven by weight efficiency. What is required is a paradigm shift in vibration reduction technology.

Active vibration control (AVC) systems have the potential for significantly reducing helicopter vibration while decreasing weight dedicated to vibration reduction. Many studies and tests have shown that active systems are effective. Early AVC studies summarized in Refs 10-12 focused on higher harmonic control (HHC) and showed that significant vibration reduction could be achieved using this approach. HHC systems introduce additional higher harmonic control inputs into the conventional swashplate to attempt to minimize NP vibrations as measured by accelerometers located in the helicopter cockpit and cabin. Although receiving considerable attention over a period of 25 years, HHC has yet to be incorporated into production aircraft. Some of the major impediments are the excessive hydraulic power requirements of HHC (Ref 10) and concerns that high frequency operation of the main rotor servos may potentially cause excessive wear to this flight critical system. A similar approach commonly referred to as individual blade control (IBC) utilizes active pitch links instead of the conventional swashplate to oscillate the blade (Ref 13). This approach has some advantages over HHC such as enabling greater control over the blade motions, but suffers from the same drawbacks of high power consumption and, in this case, being in series with the primary flight control system of the helicopter. A third approach, commonly known as active blade control (ABC), utilizes control surfaces such as trailing edge flaps on the blade to affect the blade motion (Ref 13). Moving a flap requires an order of magnitude less power than oscillating the entire blade, and does not use components of the primary flight control system. This approach offers significant promise of providing significant vibration reduction, in addition to possibly reducing the helicopter radiated noise signature and enhancing performance. However, there are many technological barriers that must be overcome before this approach can be implemented on a production helicopter.

For implementation on production aircraft, several criteria should be met by an AVC system. It must: 1) reduce aircraft vibration more effectively and over a greater range of flight conditions than passive systems; 2) result in a lower weight penalty that passive systems; 3) have low energy consumption; 4) be based on proven, reliable technology; and 5) have reasonable life-cycle costs.

One AVC approach that seems to fulfill these criteria is derived from the active control of structural response (ASCR™) method. ACSR, developed by Westland Helicopters Ltd.

(Refs. 14-15), places its actuation between two hardpoints in the fuselage to produce equal and opposite forces between these two hardpoints. The ACSR approach utilizes these dual-point actuators to minimize the response of the fuselage to the NP vibratory loads. In the case of the production ASCR system for the Westland/Agusta EH101 helicopter (Ref. 15), dual-point actuators are integrated into four of the struts supporting the main gearbox.

Unlike the EH101, Sikorsky Aircraft helicopters such as the UH-60 bolt the main gearbox directly to the airframe instead of using struts. As a consequence, ACSR dual-point type actuators are not applicable. The AVC actuator configuration discussed in this paper, and flight-tested on the UH-60 (Ref. 16), utilized single-point inertial actuators mounted at various locations in the helicopter fuselage. Unlike dual-point actuators that produce a force on the fuselage by actuating between two fixed points, single-point inertial actuators produce a force by oscillating a reaction mass that is free to vibrate. In general, to develop large forces either a large reaction mass or a large stroke of the mass is required. These two approaches, however, are neither weight efficient or power efficient. In the former case the system mass is excessive, and in the latter case the power consumption is excessive.

The single-point actuators used in the flight test, termed servo-inertial force generators (SIFGs), solved this problem by utilizing a mechanical resonance and a novel inner-loop electronic control system. Tuning the SIFGs to be near the NP frequency minimizes the required mass and power to achieve a given vibratory force output. The SIFG actuation system was designed and developed by Moog Inc. The SIFGs are devices that integrate a Sikorsky UH-60 passive vibration absorber with a hydraulic servo-actuator. The moving mass of the absorber is usually connected to the helicopter structure via three leaf springs. However, by inserting a hydraulic actuator between the structure and the middle spring, the passive vibration absorber is transformed into an active device. A schematic of the SIFG is shown in Fig. 4. Since the SIFGs were based on a modification of the passive absorber, they were capable of operating in two different modes – a passive mode (i.e., hydraulics off) in which they operated like a traditional tuned passive absorber, and an active mode in which the hydraulic actuator drives the SIFG to a desired force output. Each SIFG was capable of generating +/- 1500 lbs at NP.

A schematic of the AVC system flight tested on the UH-60 is shown in Fig. 5. The AVC system utilizes accelerometers distributed throughout the cockpit/cabin as feedback control sensors, as shown in Fig 5. The tuned passive absorber typically installed in the cabin overhead just forward of the main gearbox and in the helicopter nose were removed and two SIFGs were installed in the cabin overhead. The SIFGs could be operated in passive mode to operate just like the passive absorbers, or alternatively, operated in active mode as part of an AVC system. When in active mode, a closed-loop controller utilizes feedback from the accelerometers to determine the optimal SIFG commands to minimize the NP vibrations.

3 CONTROL ALGORITHM

The algorithm used in the ANC and AVC systems is based upon that developed by Sikorsky Aircraft and United Technologies Research Center for helicopter higher harmonic control (HHC) of rotor vibrations (Ref 17-18). In this approach the disturbance frequency is obtained from a tachometer sensor, a harmonic analyzer is used to identify the desired tonal information (i.e., magnitude and phase of frequency components of interest), and a minimum variance control algorithm is used to generate control signals based on an estimate of the plant transfer function. In Ref. 19, numerous approaches for tonal control are described, and

the connection between the underlying approach used for HHC, and other tonal control approaches, is illustrated.

In the narrow bandwidth required for control about each tone, the actuator/sensor transfer function is roughly constant, and thus the system can be modeled in the frequency domain as linear time-invariant using a single quasi-steady transfer function matrix, denoted by T. The derivation of the control algorithm given below follows from Refs. 17-18 and is described more fully in Ref. 4. Assuming linearity, the change in the sensor response vector z due to a change in the actuator command vector u can be written as a local model:

$$\Delta z = T\,\Delta u + w$$

The control law is derived to minimize the quadratic performance index:

$$J = z^T z + u^T W_u u + (\Delta u)^T W_{\delta u} (\Delta u)$$

that is a weighted sum involving the squared magnitudes of the sensor measurements, control commands, and rates-of-change of control. Substituting the local system model into the above expression and solving for the control u which minimizes J yields:

$$u_{k+1} = u_k - Y_k(W_u u_k + T_k^T z_k); \quad Y_k = (T_k^T T_k + W_u + W_{\delta u})^{-1}$$

The subscript k has been added to indicate the recursive nature of the feedback control algorithm resulting from the introduction of control rate weighting. The matrix Y_k determines the rate of convergence, but does not affect the steady state solution (Ref. 4). Greater control over the stability of the above control law is obtained with a step-size multiplier $\beta<1$:

$$u_{k+1} = u_k - \beta Y_k(W_u u_k + T_k^T z_k)$$

The behavior of the above control law is described in detail in Ref. 4.

4 FLIGHT TEST SETUP AND PROCEDURE

Active Noise Control

Two ANC flight tests were conducted on a Sikorsky S-76B commercial helicopter with a nominal gross weight of 10,000 lbs. The first flight test, conducted in 1995, is the first known successful flight test of a high frequency gear-mesh ANC system on a helicopter. The primary focus of this developmental flight test was proof-of-concept and architecture validation. The second flight test, conducted in 1996-97, focused on validating the pre-production ANC algorithms and architecture, and determining system requirements and performance tradeoffs for production.

The aircraft was equipped with a partial utility interior consisting of only the cabin ceiling and sidewall trim panels. A total of 64 microphones were distributed throughout the cabin interior, with the majority mounted on the cabin ceiling. Several microphone configurations were evaluated during the flight tests. A schematic of a typical microphone configuration tested is shown in Fig. 6.

Proof-mass actuators were bolted to each side of the transmission beams. The actuators were located on the beams as close to the gearbox mounting points as possible, since this is where the gear-mesh vibrational energy enters the airframe. Extensive ground testing validated that this approach was capable of achieving greater than 20dB tonal noise reductions. This method of mounting the actuators on the transmission beams is considered a viable approach for retrofitting an ANC system on current helicopter production lines and aircraft already in service.

The ANC flight tests included conditions such as: (1) ground runs at flat pitch (Q ~ 15%) and light-on-wheels (Q ~ 45%); (2) out-of-ground effect (OGE) hover; (3) steady flight ranging from 40 knots to V_{CR} at 145 knots, up to V_H at ~155 knots; and (4) transient maneuvers such as takeoffs, accelerations, turns, autorotations, decelerations, approaches and flares to landing.

Active Vibration Control

The AVC testing was performed on a UH-60 BlackHawk that was ballasted to 16,800lbs. The UH-60 is a four-bladed helicopter with a nominal rotor speed of 258 rpm, yielding a NP frequency of 17.2 Hz. The two baseline UH-60 tuned passive absorbers were removed and the resulting NP vibrations were measured in flight to establish a baseline. Two SIFGs were then installed in the cabin overhead just forward of the main gearbox (see Fig. 7). Accelerometers were installed at ten locations on the cockpit-cabin floor, as shown in Fig. 7, in both the vertical and lateral directions. Ten of these accelerometers, 8 in the vertical direction and 2 in the lateral direction, were selected as AVC feedback sensors.

Several flights were performed with the SIFGs operating in both passive and active mode to evaluate the benefits of AVC over conventional passive vibration reduction. The performance of the AVC system was evaluated in steady level flight at various airspeeds and rotor speeds, and during maneuvering flight.

5 FLIGHT TEST RESULTS

Active Noise Control – First Flight Test

All results presented for the ANC flight tests are for the reductions achieved in the primary bull gear tone (778Hz) of the S-76. A typical time history of gear-mesh tonal noise when the ANC system is activated, and then deactivated, is shown in Fig. 8 for an OGE hover condition. The quantity plotted in the figure represents the average reduction achieved on 24 controlled microphones. As shown in the figure, an average reduction of 9dB was achieved in the primary gear tone within 10 seconds of the ANC system being activated.

Similar reductions to those shown in Fig. 8 were achieved over a wide range of steady flight conditions, as shown in Fig. 9, with tonal reductions of 7-9dB over this speed range, including 8dB at V_{CR}. The two curves plotted in Fig. 9 represent the average gear tone level at the 24 controlled microphone locations with ANC "off" (upper curve) and with the ANC system "on" (lower curve), for various steady airspeeds.

Though the performance of the ANC system during this first flight test was sufficient to validate the system architecture for proof-of-concept, the noise reductions were much poorer than the 20dB expected based on extensive ground testing of the ANC system. Post-flight simulations based on T-matrix and ambient measurement data collected during the flight test indicated that much greater reductions should have been achieved with the architecture

implemented. Analysis of flight test data revealed excessively poor signal-to-noise ratios during the system identification procedure used to construct the T-matrix. Thus, in preparation for the second flight test, a more sophisticated system identification procedure was developed which better accounted for the high background noise levels encountered during the flight test.

Active Noise Control – Second Flight Test

The new system identification method developed for the second flight test greatly improved the estimate of the T-matrix, which resulted in dramatic improvements in ANC performance compared to the first flight test in all conditions tested, including ground runs, hover, steady forward flight, and transient maneuvers such as speed sweeps. It should be noted that all the results presented in this section were obtained using a PC-based ANC system. This system was utilized more extensively than the prototype ANC computer since the PC-based system was capable of controlling a greater number of microphones, which is more representative of the planned production version of the ANC system.

Typical noise reductions at the various microphone locations in an OGE hover condition are shown in Fig. 10. This figure shows an average reduction of 18dB on the 36 controlled microphones, compared to only 9dB on 24 microphones achieved during the first flight test (Fig. 8). It is interesting to note from Fig. 10 that the maximum tonal noise level measured in the cabin was 23dB lower with the ANC system "on" than with the ANC system "off". This is a very substantial improvement. The gear-mesh tonal reductions achieved in all steady flight conditions are very similar to those shown in Fig. 10; the OGE hover case was selected for presentation as a critical ANC condition due to the relatively high gearbox torque and resulting high gear-mesh noise levels in this condition.

Further examination of Fig. 10 reveals some interesting qualities of the noise reductions which are very noticeable to a passenger in the helicopter cabin, but may not be evident from a casual examination of the figures. The high degree of spatial variation in the ambient noise levels (i.e., with ANC "off") with microphone position should be noted in Fig. 10. For example, there is about a 20dB difference in the ambient tonal noise level between microphones 9 and 10, even though they are only about one foot apart. This spatial variation is quite evident to passengers whenever they move their heads, even for small motions, e.g. when just leaning forward. With the ANC system activated ("on") however, this spatial variation is significantly reduced, as shown in the figure. Due to the reduced overall noise levels, this reduced spatial variation is almost imperceptible to passengers, even when moving about the cabin.

ANC performance was substantially improved over the entire flight envelope, including speed sweeps from hover to V_H, compared to results obtained during the first flight test. As shown in Fig. 11, average gear-mesh tonal noise reductions of 14-16dB were achieved during a quasi-steady speed sweep, compared to typical reductions of only 7-9dB obtained during the first flight test (Fig. 9).

Also included in Fig. 11 is the ANC performance during a transient maneuver consisting of a typical acceleration from OGE hover to V_H, followed by a deceleration back to hover. It should be mentioned that the acceleration commenced immediately after take-off to hover without waiting for the ANC system to fully converge to a steady state solution. This was done to simulate actual flight procedures. As shown in Fig. 11, the ANC system not only remained stable, but maintained 8-14dB reductions relative to steady state ambient levels during the acceleration phase, and 12-14dB reductions during the deceleration phase. During accelerations, the actual ambient gear tone levels (not shown on the figure) are typically ~3dB

higher than steady flight levels due to the higher torque loads required from the gearbox. Conversely, ambient gear-mesh tonal levels are generally ~3dB lower during decelerations due to reduced gearbox torque requirements.

A typical time history of ANC performance is shown in Fig. 12 for steady flight at 120 knots. The quantity plotted in the figure represents the average reduction achieved on the 36 controlled microphones. As is evident from the figure, the controller achieved a 10dB noise reduction after three seconds, a 12dB reduction after five seconds, and then slowly converged to a steady 16dB noise reduction after 30-40 seconds. Faster convergence rates (i.e., higher controller bandwidths) were also tested without driving the controller unstable. However, these faster rates had no impact on ANC performance during steady flight conditions, and produced only minimal improvements in performance during transient maneuvers such as that shown in Fig. 11.

Active Vibration Control

The NP vibration results for 145 knots and 100% Nr are shown in Fig. 13. The three bars in the figure for each of the ten accelerometer locations represent a comparison of the ambient vibration, passive vibration control, and active vibration control cases. Also shown in the figure is a comparison of the average value of the ten accelerometers for the three cases. As expected, the AVC levels shown in Fig. 13 are lower than for the ambient case, but more significantly, these levels are substantially better than the passive case. This is a crude indication that AVC is more efficient than passive vibration control, i.e. that lower vibrations can be achieved for the same or lower weight. Especially interesting is that the vibration reductions are global in nature, i.e. reductions are realized at the forward cabin accelerometer locations which are not part of the AVC feedback sensor suite.

A global measure of the aircraft vibration is the average vibration level defined as the root-mean-square of the NP vibration magnitudes measured by the cockpit-cabin accelerometers. The average vibration measured at 12 accelerometer locations (10 AVC and 2 midcabin accelerometers) are plotted in Fig. 14 for a forward speed sweep at 100% Nr. Notice that the average NP vibrations are dramatically reduced over the entire range of flight speeds, and that the AVC reductions are significantly better than the passive case.

As discussed previously, one major advantage of AVC over passive systems is their ability to adapt to changing rotor speeds. The passive system on the UH-60 aircraft is a fixed tuned vibration absorber system optimized to a rotor speed of 100% Nr. A comparison between the passive and AVC system performance versus rotor speed is shown in Fig. 15. The classical "bucket" is shown for the passive system, i.e. the vibration is a minimum at 100% Nr. In contrast, the AVC system can readily follow the rotor speed variation, producing a virtually flat response. Of special interest is that the AVC case yields lower vibration levels than passive at 100% Nr. This is further confirmation that AVC can be more efficient than passive methods. Of course the SIFG hydraulic actuators must work harder as the Nr departs from 100%, but sufficient hydraulic power is available over the typical range of rotor speeds.

Sikorsky S-92 AVC

Encouraged by the SIFG flight test on the UH-60, a trade-off study of AVC versus tuned passive vibration absorbers was performed during the preliminary design phase of the S-92, and the decision was made to include the AVC system as baseline on all S-92 aircraft. The results of the trade study indicated that an AVC system would provide a lighter-weight

solution than a comparable tuned passive absorber system. Furthermore, an AVC system of comparable weight to a passive system would provide greater vibration reduction.

The S-92 began development flight testing in December 1998 and included an AVC system from first flight. Typical AVC performance is plotted in Fig. 16 for a steady Vcruise condition. The plot compares the NP vibration levels with AVC off and AVC on at 10 accelerometer locations in the cockpit-cabin. The AVC system is similar to the SIFG system, except that the SIFGs have been replaced with purely mechanical actuators referred to as force generators (FGs). The results shown in Fig. 16 were obtained with 3 FGs. It should be noted that the FG locations flight tested so far are trial locations only and are most likely not optimal. But as shown in the figure, AVC achieves substantial vibration reduction in cruise. Developmental flight testing of the AVC system on the S-92 will continue through 1999 and 2000 and will focus on optimizing the FG locations with the goal of maximizing vibration reduction for systems with one, two or three FGs.

6 CONCLUDING REMARKS

Active Noise Control

An approach for actively controlling high frequency structure-borne tonal noise in helicopters has been validated in a flight test program on the S-76 aircraft. Structural actuation near the gearbox mounts has been used to cancel the disturbance before it enters the airframe. This approach has been successfully demonstrated to produce substantial reductions in the primary gear-mesh tone of the helicopter, over a wide range of flight conditions. These reductions have bee maintained during maneuvers such as typical accelerations and decelerations with good system stability. Application of this ANC technology will provide noise suppression and create a quieter passenger environment.

Active Vibration Control

AVC systems using single point actuators is a viable vibration reduction technology. The system uses minimal power due to the resonant behavior of the SIFGs and can revert to a passive absorber in the event of a controller failure. A global vibration metric shows that AVC is more efficient than passive systems for a wider range of forward speeds and rotor speeds. AVC systems have the capability to adapt to changes in helicopter loading and rotor speed whereas traditional passive tuned absorbers cannot. AVC technology, properly integrated with traditional passive vibration reduction systems, offers promise of yielding a significant improvement in helicopter vibration levels.

7 REFERENCES

1. Elliott, S.J., Nelson, P.A., Stothers, I.M., and Boucher, C.C., In-flight experiments on the active control of propeller-induced cabin noise, Journal of Sound and Vibration, (1990) 140(2), pp. 219-238.
2. Boucher, C.C., Elliott, S.J., and Baek, K.H., Active control of helicopter rotor tones, Inter-noise '96, pp. 1179-1182.
3. Nelson, P.A., and Elliott, S.J., Active Control of Sound, Academic Press Limited, 1992.

4. MacMartin, D.G., Davis, M.W., Yoerkie Jr, C.A., and Welsh, W.A., Helicopter Gear-Mesh ANC Concept Demonstration, Active '97, Budapest - Hungary, August 1997, pp. 529-542.
5. Millott, T.A., Welsh, W.A., Yoerkie, C.A., MacMartin, D.G., Davis, M.W., Flight Test of Active Gear-Mesh Noise Control on the S-76 Aircraft, Proceedings of the 54th American Helicopter Society Annual Forum, Washington, D.C., May 1998.
6. Elliott, S.J., "Active Control of Structure-Borne Noise", Journal of Sound and Vibration (1994) 177(5), 651-673.
7. Sutton, T.J., Elliott, S.J., Brennan, M.J., and Heron, K.H., Active isolation of noise transmission through a helicopter gearbox support strut using multiple magnetostrictive actuators, 1996 International Conference on Noise and Vibration Engineering, Leuven, Belgium, pp.315-328, 1996.
8. Yoerkie Jr., C.A., Welsh, W.A., and Sheehy, T.W., Helicopter Active Noise Control System, US Patent Number 5,310,137, May 1994.
9. Veca, A.C., Vibration Effects On Helicopter Reliability and Maintainability, USAAMRDL Technical Report 73-11, April 1973.
10. Miao, W., Kottapali, S.B.R., Frye, H.M., Flight Demonstration of Higher Harmonic Control (HHC) on S-76, Proceedings of the 42nd Annual Forum of the American Helicopter Society, Washington, D.C., June 1986.
11. Achache, M., Polychroniadis, M., Higher Harmonic Control – Flight Tests of an Experimental System on the SA349 Research Gazelle, 42nd Annual Forum of the American Helicopter Society, June 1986.
12. Welsh, W.A., Higher Harmonic and Trim Control of the X-Wing Circulation Control Wind Tunnel Model Rotor, Proceedings of the 45th Annual Forum of the American Helicopter Society, Boston, MA, May 1989.
13. Friedmann, P.P., Millott, T.A., Vibration Reduction in Rotorcraft Using Active Control: A Comparison of Various Approaches, Journal of Guidance, Control, and Dynamics, Vol. 18, No. 4, July-August 1995.
14. Staple, A.E., An Evaluation of Active Control of Structural Response as a Means of Reducing Helicopter Vibration, 15th European Rotorcraft Forum, September 1989.
15. Staple, A.E., Wells, D.M., Development and Testing of an Active Control of Structural Response System for the EH101 Helicopter, Proceedings of the 16th European Rotorcraft Forum, September 1990.
16. Welsh, W., Frederickson, C., Rauch, C., Lyndon, I., Flight Test of an Active Vibration Control System on the UH-60 BlackHawk Helicopter, Proceedings of the 51st Annual Forum of the American Helicopter Society, Fort Worth, TX, May 1995.
17. Taylor, R.B., Farrar, F.A., and Miao W., "An Active Control System for Helicopter Vibration Reduction by Higher Harmonic Pitch," AIAA paper No. 80-0672, American Helicopter Society 36th Annual Forum, Washington D.C., May 1980.
18. Davis, M.W., "Refinement and Evaluation of Helicopter Real-Time Self-Adaptive Active Vibration Controller Algorithms", NASA Contractor Report, UTRC Report R83-956149-16.
19. Sievers, L.A., and von Flotow, A.H., Comparison and Extensions of Control Methods for Narrow-Band Disturbance Rejection, IEEE: Transactions on Signal Processing, Vol. 40 No. 10, October 1992.

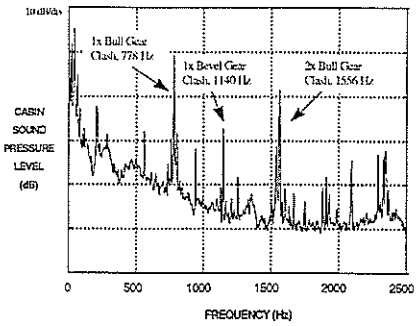


Fig. 1 Interior noise spectrum (unweighted) of a S-76 helicopter

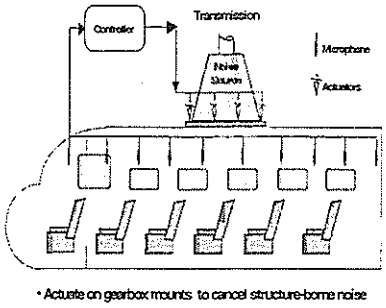


Fig. 2 Control architecture for helicopter gear-mesh ANC

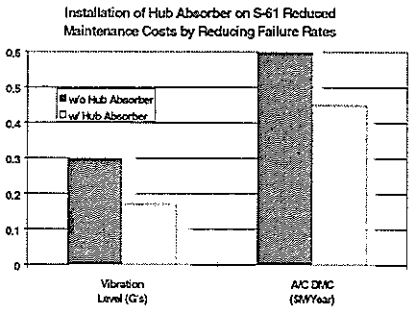


Fig. 3 Impact of hub absorber on vibrations and maintenance costs

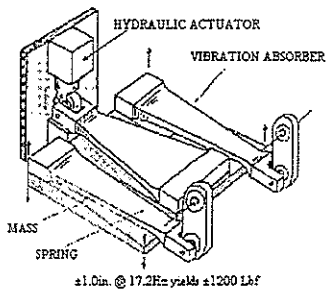


Fig. 4 Servo Initial Force Generator (SIFG)

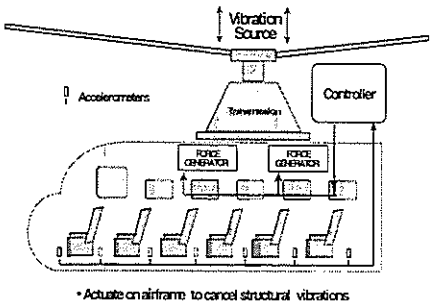


Fig. 5 Control architecture for AVC

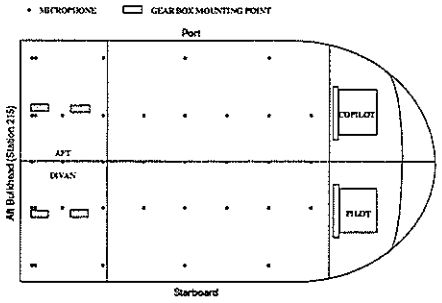


Fig. 6 Typical arrangement of feedback control mics used in S-76 ANC

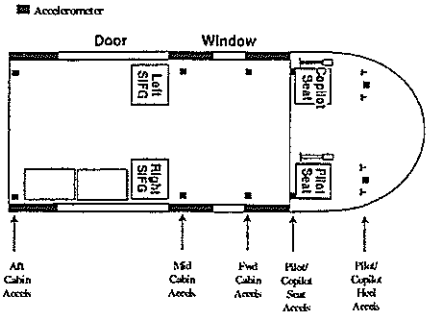


Fig. 7 AVC aircraft installation in a Sikorksy UH-60

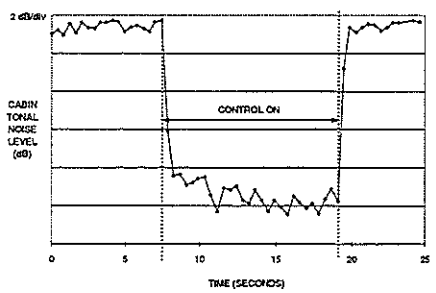


Fig. 8 Typical gear-mesh ANC performance time history in OGE hover.

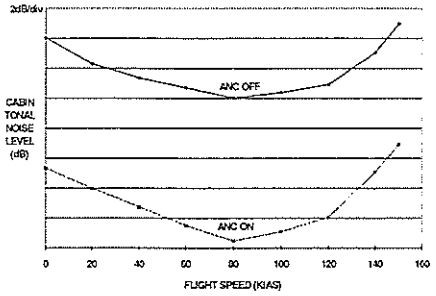


Fig. 9 ANC performance achieved during quasi-steady speed sweep.

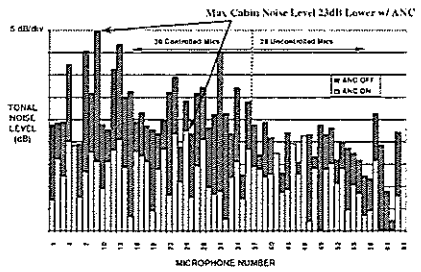


Fig. 10 ANC achieves 18 dB average reductions in OGE hover.

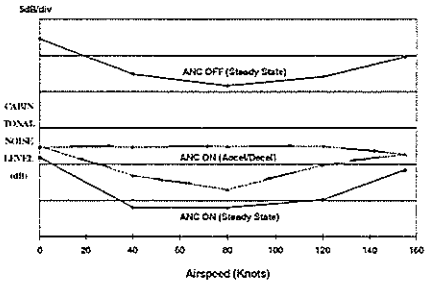


Fig. 11 ANC performance during quasi-steady & transient speed sweep.

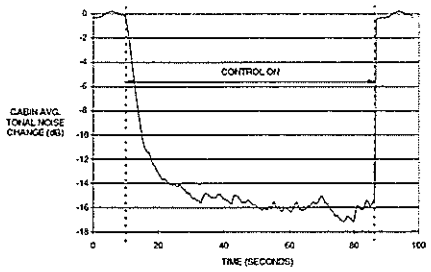


Fig. 12 Typical time history of gear-mesh ANC performance at 120 knots.

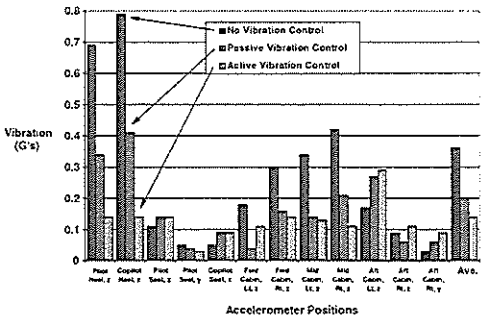


Fig. 13 SHFG vibration reduction during steady flight at Veruise

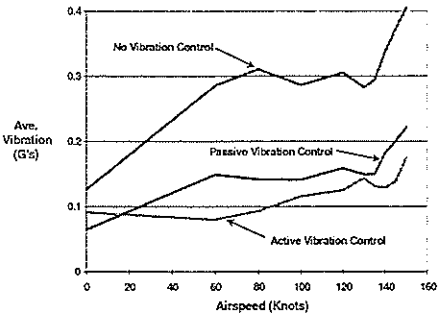


Fig. 14 SHFG vibration reduction during a speed sweep at 100% Nr

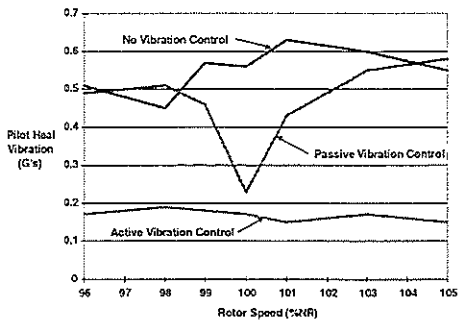


Fig. 15 SHFG performance versus rotor speed

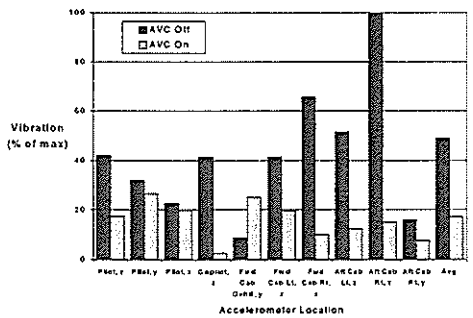


Fig. 16 S-92 AVC performance at Veruise

TWENTY-FIFTH EUROPEAN ROTORCRAFT FORUM

Paper no. G24

THE USE OF DISCRETE OBSERVER THEORY TO TRIM AND STABILIZE PERIODIC COEFFICIENT DYNAMIC SYSTEMS

BY

MICHAEL H. PETERS
Graduate Research Assistant

PHILIP V. BAYLY
Associate Professor

DAVID A. PETERS
McDonnell Douglas Professor
and Chairman

DEPARTMENT OF MECHANICAL ENGINEERING
WASHINGTON UNIVERSITY
ST. LOUIS, MO 63130 USA

SEPTEMBER 14-16, 1999
ROME
ITALY

ASSOCIAZIONE INDUSTRIE PER L'AEROSPAZIO, I SISTEMI E LA DIFESA
ASSOCIAZIONE ITALIANA DI AERONAUTICA ED ASTRONAUTICA

The Use of Discrete Observer Theory to Trim and Stabilize Periodic Coefficient Systems

by

Michael H. Peters
Graduate Research Assistant

Philip V. Bayly
Associate Professor

and

David A. Peters
McDonnell Douglas Professor and Chairman
Department of Mechanical Engineering
Washington University – Campus Box 1185
St. Louis, MO 63130

Abstract

The nonlinear differential equations for rotorcraft simulation (in this case, flap-lag-pitch and ground resonance equations) are sampled once a period (or once a blade passage) and the resultant sampled errors in periodicity and trim are considered a discrete system. The algebraic Riccati Equation is then used to design a controller to trim this discrete system. The resultant controller is applied to the original nonlinear simulation in that the errors at each blade passage are fed back to give discrete control changes. When all states cannot be measured, a discrete observer is used to estimate them. The resultant algorithm is shown to be a robust tool both for trimming and for stabilizing a nonlinear set of rotorcraft equations.

Notation

a	lift curve slope, /radian
[A]	matrix multiplying states in discrete system
$[A]_t$	modified matrix = $A - BG$
b	blade semi-chord, m
[B]	matrix multiplying controls in discrete system
c_d	blade profile drag coefficient
C_T	thrust coefficient, $T/\rho(\Omega R)^2 \pi R^2$
$[C_z]$	observer matrix
$[C(\psi)]$	damping matrix, period T
$d_k, \hat{d}$	desired constraint values
dpl	lag damper setting, percent critical damping
f_j	state-derivative functions
$f_\beta, f_\theta, f_\zeta$	generalized loads
$J_\beta, J_\theta, J_\zeta$	blade principal moment of inertia in flap, torsion, lag, $kg-m^2$
g_k	constraints
G_k	constraint integrand
$[J]$	Jacobian matrix
K_β	flap stiffness, N-m/rad
K_ζ	lag stiffness, N-m/rad
K_θ	pitch stiffness, N-m/rad
$[K(\psi)]$	stiffness matrix, period T
$[M]$	mass matrix
P	stiffness parameter for blade

	flap $1+K_\beta/\Omega J_\beta$
Q	number of blades
R	radius of rotor, m
T	period of system
V_i	induced flow, normal to disk
V_x	inplane flow of disk
V_z	normal flow to disk
x_j	state variables
Z_n	discrete states
β	flap angle positive flap and angle positive up, rad
β_0	coning angle, rad
β_s	lateral cyclic flap angle, rad
γ	Lock number $\rho acR^4/J_\beta$
$\Delta(\,)$	difference of ( )
θ_b	pilot pitch control $\theta_b=\theta_0+\theta_s\sin\psi+\theta\cos\psi$
θ_0	collective pitch angle
θ_s	longitudinal cyclic pitch angle
θ_c	lateral cyclic pitch angle
Θ_m	vector of controls
λ	inflow ratio, $(V_i+V_z)/\Omega R$
$[\Lambda]$	square matrix with eigenvalues on the diagonal
μ	advance ratio, $V_x/\Omega R$
ρ	density of air, kg/m^3
σ	rotor solidity, $Qc/\pi R$
ψ	rotor azimuth angle = Ωt, rad
Ω	rotor angular velocity, rad/sec
ω_ζ	dimensionless nonrotating lead-lag frequencies at $\theta=0,\sqrt{K_\zeta/\Omega^2 J_\beta}$
ω_θ	dimensionless nonrotating elastic pitch frequencies at $\theta=0,\sqrt{K_\theta/\Omega^2 J_\beta}$
$(\,)_n$	n-th iteration of vector

Introduction

The analysis of a rotary-wing system consists of three steps. First, one must assemble a set of ordinary differential equations that describes the physics of the system. Unlike modeling in many other technical disciplines, the equations so assembled are generally nonlinear, have periodic coefficients, and contain certain unknown parameters called "trim settings". It is the job of the analyst to solve these equations subject to a number of side constraints on the solution equal in number to the number of unknown parameters. Thus, the second step in a rotary-wing analysis is to find simultaneously the unknown parameters and the corresponding solution that satisfies these constraints. (This is called the "trim" solution.) The third step in the analysis is to analyze the stability of the solution about this periodic orbit. Many times, the solution strategy used to find the trim solution holds the key to the stability analysis.

There are several reasons that the trim solution is difficult. First, although there may be only a few trim variables (usually 3-7), there may be thousands of dynamic states. Thus, an iterative solution of the state-space equations for a periodic solution may be prohibitive. Second, there are often hidden states that occur due to the various mathematical algorithms that are used to evolve the system equations through time. Thus, the states may not be in the form to be interrogated numerically. Third, the periodic trim solution is often unstable (due to aeroelastic effects or due to flight mechanics modes) so that it is impossible simply to time-march until transients decay.

Recent work has found that, when all states are known and limited to a hundred or so, a very effective trim strategy is the discrete auto-pilot, Ref. [1]. Such an auto-pilot makes adjustments to the trim settings every blade passage based on an optimized gain and controller. Such a controller can fly even an unstable system to equilibrium and satisfy the trim constraints. Since a discrete auto-pilot does not change the poles of the system, it is perfect for a subsequent stability analysis, Ref. [2].

Despite the successes of such a system, it is still subject to the limitation that all

states must be known and that they must be limited to about one hundred. In this paper, we offer an alternative strategy designed for the case in which states are hidden. In particular, a discrete-time observer is created that gives an estimate of the hidden states that can be used in the discrete auto-pilot. This paper will describe how such an observer can be created and will give some simple examples for helicopter flap-lag-torsion stability.

The use of observers is not new to rotorcraft analyses, Refs. [3] and [4]; but this paper uses them in the sense of discrete systems and for mathematical trim and stabilization rather than real-time vibration control.

Mathematical Background

The nonlinear differential equations for a rotorcraft following general form:

$$\dot{x}_j = f_j\left(x_k, t, \Theta_m\right) \qquad (1)$$

where the x_j are the state variables, Θ_m are the trim parameters, and t is time. For every unknown trim parameter, there is a trim constraint of the form:

$$g_k \equiv \int_0^T G_k\left(x_i, t, \Theta_m\right)dt = d_k \qquad (2)$$

In general, states can be comprised of positions (displacement or rotations), rates, pressures, inflows, engine temperatures or pressures, damper states, etc. Trim variable can be collective and cyclic pitch position, throttle position, prescribed fuselage attitudes, RPM, or any unknown parameter in the equations. The trim constraints can be force or moment equilibrium, time averages or space averages of vehicle positions or velocities, etc. In trimming, one usually considers the equations to be linearized about some periodic equilibrium, $\bar{x}$ during each part of the design process. Thus, for simplicity, we consider the linearization of Eq. (1) about $\bar{x}_j$,

$$\left\{\dot{x}_j\right\} = \left[A(t)\right]\left\{x_j\right\} + \left[b(t)\right]\left\{\Theta_m\right\} \qquad (3)$$

where x_j are the new states which are perturbation states; and it is understood that $A(t)$ and $b(t)$ depend on $\bar{x}_j$. Equation (3) is never actually used in our trim algorithm, but it is an important conceptual tool.

The trim problem is to find the initial conditions $x_j(0)$ and trim parameters, Θ_m, that will result in a periodic solution to either Eq. (1) or Eq. (3) that satisfies the trim constraints. One way which has been tried in the past, Ref. [1], is to design an auto-pilot for the linear system in Eq. (3) that produces a Θ_m proportional to the current errors in the trim constraints. Three disadvantages of this approach are: 1) it produces unsteady $\Theta_m(t)$ when they should be constants, 2) it is hard to find a suitable auto-pilot since this requires solution of a periodic-coefficient Riccati Equation, and 3) the method cannot work if the system is unstable. References [1] and [2] have shown that this can be overcome with a discrete auto-pilot. In particular, if the system is sampled once a period, the errors in periodicity, $x(T) - x(0)$, and the errors in the trim constraints in Eq. (2), $e_k = g_k - d_k$, can be considered as a discrete system vector, Z_n

$$Z_{n+1} = A_n Z_n + B_n \Theta_n \qquad (4)$$

where n is an iteration index, not a vector subscript. Again, Eq. (4) is not actually used in the discrete auto-pilot, but it is a conceptual step that is used to understand how to design such an auto-pilot.

Naturally, it is important to know how the discrete matrices, A_n and B_n, relate to the continuous matrices, $A(t)$ and $B(t)$, or to perturbations of the original nonlinear system in Eq. (1). In particular, we consider how A_n and B_n would be related to the Jacobian matrix that would be formed by

making perturbations to an initial guess for the $x(0)$ and Θ_m.

The Z_n vector is defined as follows:

$$Z_n = \begin{Bmatrix} x_n - x_{n-1} \\ g_n - \hat{d} \end{Bmatrix} \equiv \begin{Bmatrix} \Delta x_n \\ e_n \end{Bmatrix} \qquad (5)$$

where g_n is the nth iteration on constraints, and $\hat{d}$ is the vector of desired constraint values. Next, one defines the following Jacobian matrix (J) for the linear system in Eq. (3). This Jacobian can be computed by time-marching with perturbations to x_j and Θ_m.

$$\begin{Bmatrix} x_{n+1} - x_n \\ g_{n+1} - \hat{d} \end{Bmatrix} = [J] \begin{Bmatrix} x_n \\ \Theta_n \end{Bmatrix} \qquad (6)$$

Here is exactly how A and B are computed from a given J. In general:

1) partition J according to x and Θ.

$$[J] = \begin{bmatrix} J_{xx} & J_{x\Theta} \\ J_{\Theta x} & J_{\Theta\Theta} \end{bmatrix} \qquad (7)$$

2) Rearrange the top partition of equation (6) to be in standard form

$$\begin{aligned} \{x_{n+1}\} &= [J_{xx} + I]\{x_n\} + [J_{x\Theta}]\{\Theta_n\} \\ \{x_n\} &= [J_{xx} + I]\{x_{n-1}\} + [J_{x\Theta}]\{\Theta_{n-1}\} \end{aligned} \qquad (8)$$

3) Subtract to obtain equations for errors in periodicity

$$\{\Delta x_{n+1}\} = [J_{xx} + I]\{\Delta x_n\} + [J_{x\Theta}]\{\Delta\Theta_n\} \qquad (9)$$

4) Similarly, consider that the constraint errors from equation (6) are

$$\begin{aligned} \{e_{n+1}\} \equiv \{g_{n+1} - \hat{d}\} &= [J_{\Theta x}]\{x_n\} + [J_{\Theta\Theta}]\{\Theta_n\} \\ \{\Delta e_{n+1}\} &= [J_{\Theta x}]\{\Delta x_n\} + [J_{\Theta\Theta}]\{\Delta\Theta_n\} \end{aligned} \qquad (10)$$

5) Recombine Eq. (9) into the form of Eqs. (4) and (5)

$$\begin{Bmatrix} \Delta x_{n+1} \\ e_{n+1} \end{Bmatrix} = \begin{bmatrix} J_{xx} & 0 \\ J_{\Theta x} & I \end{bmatrix} \begin{Bmatrix} \Delta x_n \\ e_n \end{Bmatrix} + \begin{bmatrix} J_{x\Theta} \\ J_{\Theta\Theta} \end{bmatrix} \Delta\Theta_n \qquad (11)$$

It is now clear that the partitioned matrices in Eq. (11) are the A_n and B_n of the discrete time model. Thus, one can find A_n and B_n from a numerical Jacobian of the original, non-linear time domain system.

Discrete Controller

What we would now like to do is to design a controller that produces changes to the controls at the end of every period (or blade passage) based on the errors in periodicity and trim constraints (i.e., the Z_n). In general, we would like a feedback law that would give

$$\Delta\Theta_n = -GZ_n \qquad (12)$$

If one can find an appropriate *gain* matrix G, then Eq. (4) would become:

$$Z_{n+1} = (A - BG)Z_n \qquad (13)$$

where $(A - BG)$ is designated A_r.

Although we plan to use $\Delta\Theta_n$ on the original continuous nonlinear equations to perturb the controls after every simulation of blade passage, we will design the controller G based on the *discrete* form in Eq. (4). We desire a G that places all of the eigenvalues of A_r less than unity and as close to zero as possible. Such eigenvalues will drive a difference equation to equilibrium. For linear systems, such a G can be found from an algebraic Riccati equation, Ref. [6]. Once A, B, and G are found, Eq. (13) is a feedback system the states of which converge to an equilibrium; thus, the thought experiment using the linear discrete system is over.

As the simulation runs, we sample the continuous system to obtain e_n and x_n and compare the sample with the previous sample to form a Δx_n. These form Z_n by Eq. (5). Equation (12) is then used to adjust the controls. It is important to keep in mind that x_n must be completely measurable in order for Eq. (12) to be used.

We now come to the consideration of what happens if some states cannot be measured or the case in which we can measure only certain combinations of the states. This can happen for either of two reasons. Either there are too many states to measure, or some states are truly hidden due to the nature of time-marching algorithm. For example, a pure time delay in a dynamic stall model or in a drive-train model has an infinite number of states, and they are all hidden. Another example would be when a structures code is linked to a CFD or free-wake code in which states are hidden in the "black box" that is the aerodynamic algorithm for generating induced flows. The idea here is to use an observer to estimate states that cannot be measured.

If we only know some of Z_k, say,

$$Y_k = C_z Z_k \tag{14}$$

we will make a new system in $\hat{Z}_k$ which will be an estimate of Z_k. The $\hat{Z}_k$ dynamic system should be a discrete, dynamic feedback system having a forcing function dependent on Y_k and that we can control such that $\hat{Z}_k = Z_k$ a time increases. In other words, we want to build a dynamic observer that will estimate $\hat{Z}_k$ given the measurements that are known, Y_k

$$\hat{Z}_k = A_c \hat{Z}_{k-1} + L Y_k \tag{15}$$

for some A_c and L. Thus, we have $\hat{Z}_k$ dependent on $\hat{Z}_{k-1}$ and Y_k.

Now replace Z_k in Eq. (14) above with $\hat{Z}_k$ such that all terms in Eq. (15) are $k-1$. This yields

$$\hat{Z}_k = A_c \hat{Z}_{k-1} + L C_z A_t Z_{k-1} \tag{16}$$

Now define $E_k = Z_k - \hat{Z}_k$ by Eqs. (4) and (16). After collecting Z_{k-1} terms, and setting:

$$A_c = A_t - L C_z A_t \tag{17}$$

we have

$$E_k = (A - L C_z A) E_{k-1} \tag{18}$$

All that is needed is to find an L which will minimize the eigenvalues of $A - LC_zA$. This can be done by eigenvalue placing or by a Riccati solution.

Now, the $\hat{Z}$ term can be used entirely in place of the Z term which could not be measured. Specifically, we use instead of Eq. (12)

$$\Delta\Theta_k = -G\hat{Z}_k \tag{19}$$

where $\hat{Z}_k$ is computed from Eq. (15).

For the case in which we want to stabilize the system (rather than trim it), the mathematics is identical. There are no trim constraints, and the θ_k become the controls used to stabilize. As the controller drives the solution toward a periodic solution, it will automatically stabilize it.

Mathematical Model

We are now ready to apply this theory to an actual case. In this study, we will use the rigid-blade, flap-lag-torsion equations for a rotor in forward light which implies 6 states. We assume quasi-steady aerodynamics and neglect reversed flow. For trim variables, we include collective pitch and the two cyclic pitch values. The corresponding trim constraints are the time-

averaged coning minus the desired value, and the time-averaged tip-path-plane tilts. Equations of motion for this system are based on Ref. [7], but with some corrections, and appear in the Appendix of Ref. [8].

To proceed with our methodology, an initial guess on controls and initial conditions is used to time march through one period for a baseline. Next, a Jacobian of perturbations is computed during which all states are measurable. Later, we will assume that, even if all states are not measurable for use in the feedback law, that the Jacobian is still available by some other method. $[A]$ and $[B]$ are computed from the partitioned $[J]$ as in Eq. (11). To find G , we tried two kinds of software. First, we tried using software which simply placed the eigenvalues of A_t. This resulted in an ill-conditioned G because such software works by first asking the user to input the exact desired placement of the eigenvalues of A_t, $[\Lambda]$, and then iteratively (by brute force) finding a G which satisfies the characteristic equation for the given $[\Lambda]$. This iterative algorithm in MATLAB, called "PLACE", created a matrix of very large numbers (order 1,000 or higher). Such an ill-conditioned G, when used in Eq. (12) tends to give huge $\Delta\Theta_n$.

An alternative approach is to do an optimal control in which the squares of controls and states are minimized over the iterations. The weighting functions of the errors and controls are set equal to unity for our work. The fact that states are in the objective function keeps the system stable. The fact that controls are in the objective function keeps the gains from becoming too large, as they do in eigenvalue placement. The result is a Riccati equation for G. The G computed by this Riccati solver is used as the feedback law in the discrete controller. The results work well in this application.

Next, we turn to creation of the observers. L must be computed by the same type Riccati method used to compute G for Eq. (12). The MATLAB command used here is called "DLQR."

Note that a Riccati solver may require that a matrix be in the form $A-BG$. Equation (18) is not in this form, however. Instead, we have the form $A-GB$. The equivalent system is $A^T-B^TG^T$ which can be "plugged into" the Riccati solver to yield a gain matrix which is the transpose of the G needed for $A-GB$. Also note that the MATLAB command "DLQR" does not work very well in this case because it does not accept non-square A's and B's. We therefore made a copy of the source code for DLQR in a new file, named it ndlqr (for New-DLQR), and changed it to work with non-square matrices. The math is the same as long as the dimensions agree, because the Riccati equation does not care if its arguments are square. The alternative is to put the non-square matrices from Eq. (20) into large square matrices the rest of which are just zeros.

Next, the simulation is started; and, after the n^{th} simulated blade revolution, a $\Delta\Theta_n$ is computed by Eq. (12) or Eq. (19). Then $\Delta\Theta_n$ is applied to the controls. This process is repeated until the error in the trim constraint is within certain parameters at which point the current controls are said to be approximately the "trim" control setting for the given system.

The above is similar to the development in Ref. [8] except that we have improved some of the derivations to give a better observer theory.

Results and Discussions

The equations of motion for all numerical experiments are given in Refs. [8] and [9]. The system of equations used for the numerical experiments were first run with the following constant parameters.

b/R	=	0.10
c_d/a	=	0.0016
dlp	=	0.10

P	=	$(1.05)^2$
σ	=	0.1
γ	=	5.0
μ	=	0.3
ω_ζ	=	0.49
ω_θ	=	14.1
C_T	=	0.003
J_ζ/J_β	=	1.01
J_Θ/J_β	=	.01

The above parameters yield an inherently stable system which simulates a helicopter in forward flight. In other words, even if there is not a closed-loop feedback control, the states will converge to a periodic flight path, although this equilibrium will probably not fulfill the trim constraints.

We begin with a discrete controller when all states can be measured. Figure 1 gives the error in trim constraints and the control settings as a function of iteration number. The first guess is identically zero. One can see that, after about 20 iterations, the system is trimmed; and the controls have reached their final values. Although not shown, the periodicity errors have also gone to zero. (Since these errors are in terms of radians of flapping angle, the final error of .001 is only .05 degrees.) This is typical of the trim convergence of a discrete auto-pilot with unity weighting matrices. In general terms, the control system is over-damped and could be improved by other weighting factors.

Next, we consider the same system but assume that we can measure only the trim-constraint errors. No states are assumed to be measurable. Thus, the C_z matrix is assumed to be a 9x3 matrix of all zeroes except for an identity at the far right. This makes the computation of the L for the observer very straightforward. The result is indistinguishable from that of Fig. 1. In other words, the direct measurement of states is not that important to the trim auto-pilot.

In the next case, we consider a system that is unstable. We switch to a stiff-inplane rotor $\left(\omega_\zeta = 0.70\right)$ at a higher advance ratio $\left(\mu = 0.4\right)$. We remove the lag damper (dpl = 0.10) and replace it with 1% negative damping (dpl = -.01). This gives a powerful flap-lag instability. We assume that only the three trim constraints can be measured. Thus, this is a case with an observer. Figure 2 shows the errors in trim constraints and the developing control settings. Also shown on the lower half of Fig. 2 are the evolving controls when all states are measured. Although the lack of state measurements causes some initial oscillations, the ultimate convergence is the same as when all states are measured. The convergence is also about the same as it was for the stable system, Fig. 1.

These results are much better than those previously published in Ref. [8] due to an improvement in the observer equations. The observer equations here are slightly different than those in Ref. [8].

In the next set of results, we apply the control to a 4-bladed ground resonance model. Thus, these are no trim side constraints; and the control to a periodic equilibrium is actually a stabilizing control to eliminate the ground resonance instability. Equations for the system can be found in Ref. [9]. The control is assumed to be an applied roll moment accomplished by force actuators at the base. It is assumed that only 2 of the twelve states can be measured, roll angle and pitch angle. Thus, an observer is used.

Figure 3 shows the resulting roll and pitch angles as well as the necessary control torque to achieve stabilization. Note that torque is discretely changed once per revolution. Figure 4 shows the motion (with the same step input in roll) when the controller is not used One can see the powerful ground resonance instability.

Summary and Conclusions

The method of discrete control with observer theory is applied to the problem of

numerical trim of a rotorcraft simulation. First, the general problem of rotorcraft trim is reformulated in a discrete-time framework. Second, we show how to do numerical perturbations of simulation code in order to find a Jacobian. Third, we show how to use the partitioned Jacobian to form the discrete-time system matrices. These can then be sent to standard Riccati tools to find the controller and the observer.

The methodology is applied to nonlinear, flap-lag-torsion equations for a rotor blade in forward flight and to ground resonance. The results shown that the above scheme can successfully trim both stable and unstable cases in the absence of all measurements. For stable cases, only the trim constraints need to be measured. For unstable cases, the method stabilizes the system.

Acknowledgment

This work was sponsored by the United States Army Research Office AASERT Grant No. DAAH04-96-1-011, Dr. Gary Anderson, Technical Monitor.

References

1. Li, Si-Hao and Peters, David A., "A Combined Periodic-Shooting, Auto-Pilot Technique for Rotorcraft Analysis," International Conference on Computational Engineering Science, Mauna Lani, Hawaii, July 30-August 5, 1995.
2. Schmitt, John M., Bayly, Philip V., and Peters, David A., "Stabilization of Periodic Flap-Lag Dynamics in Rotor Blades," AHS 2nd International Aeromechanics Specialists' Conference, Bridgeport, October 11-13, 1995.
3. Davis, Mark, "Development and Evaluation of a Generic Active Helicopter Vibration Controller," Proceedings of the 40th Annual Forum of the AHS, May 16-18, 1994.
4. Sopher, Robert and Hallock, Daniel, "Time-History Analysis for Rotorcraft Dynamics Based on a Component Approach," 2nd Decennial Specialists' Meeting on Rotorcraft Dynamics NASA Ames Research Center, Moffett Field, CA, November 1984.
5. Peters, David A., and Barwey, Dinesh, "A General Theory of Rotorcraft Trim," *Mathematical Problems in Engineering,* Vol. 2, 1996, pp. 1-34.
6. Ramirez, W. Fred, *Process Control and Identification,* Academic Press, Boston, 1994, Chapter 4.
7. Li, Si-Hao, *A Hybrid Periodic-Shooting, Auto-Pilot Method for Rotorcraft Trim Analysis,* Doctoral Dissertation, Washington university in St. Louis, May 1997.
8. Peters, Michael H. and Peters, David A., "Discrete Control Theory and Dynamic Observers Applied to Rotorcraft Stability and Trim," Proceedings of the 54th Annual Forum of the American Helicopter Society, Washington, D.C., May 20-22, 1998.
9. Powell, et. al, *A Modern Course in Aeroelasticity,* Kluwer Academic Publishers, Dordecht, The Netherlands, Third Revised Edition, 1995, pp. 400-411.

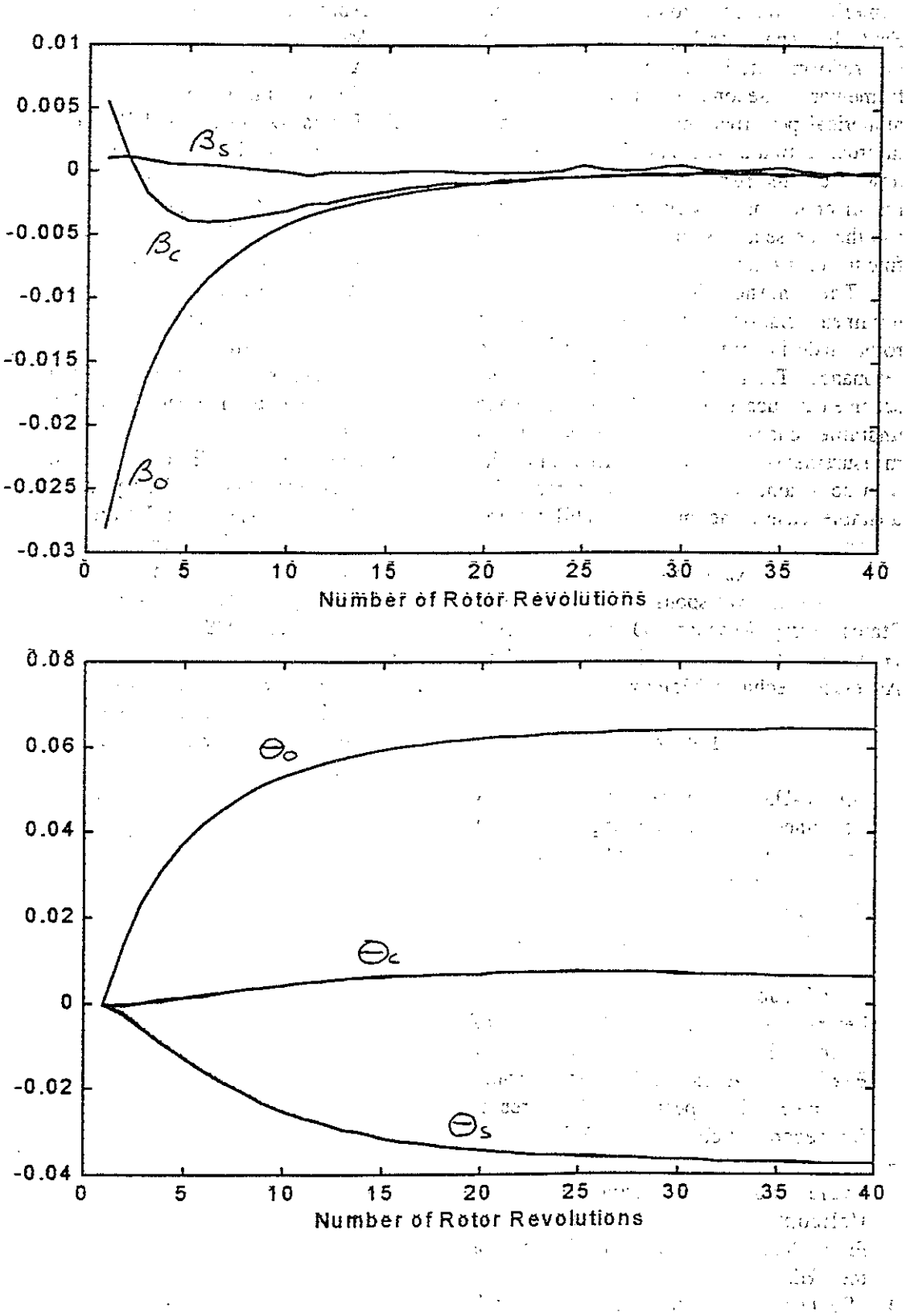


Figure 1. Trim Errors (top) and Controls (bottom) for Flap-Lag-Torsion.

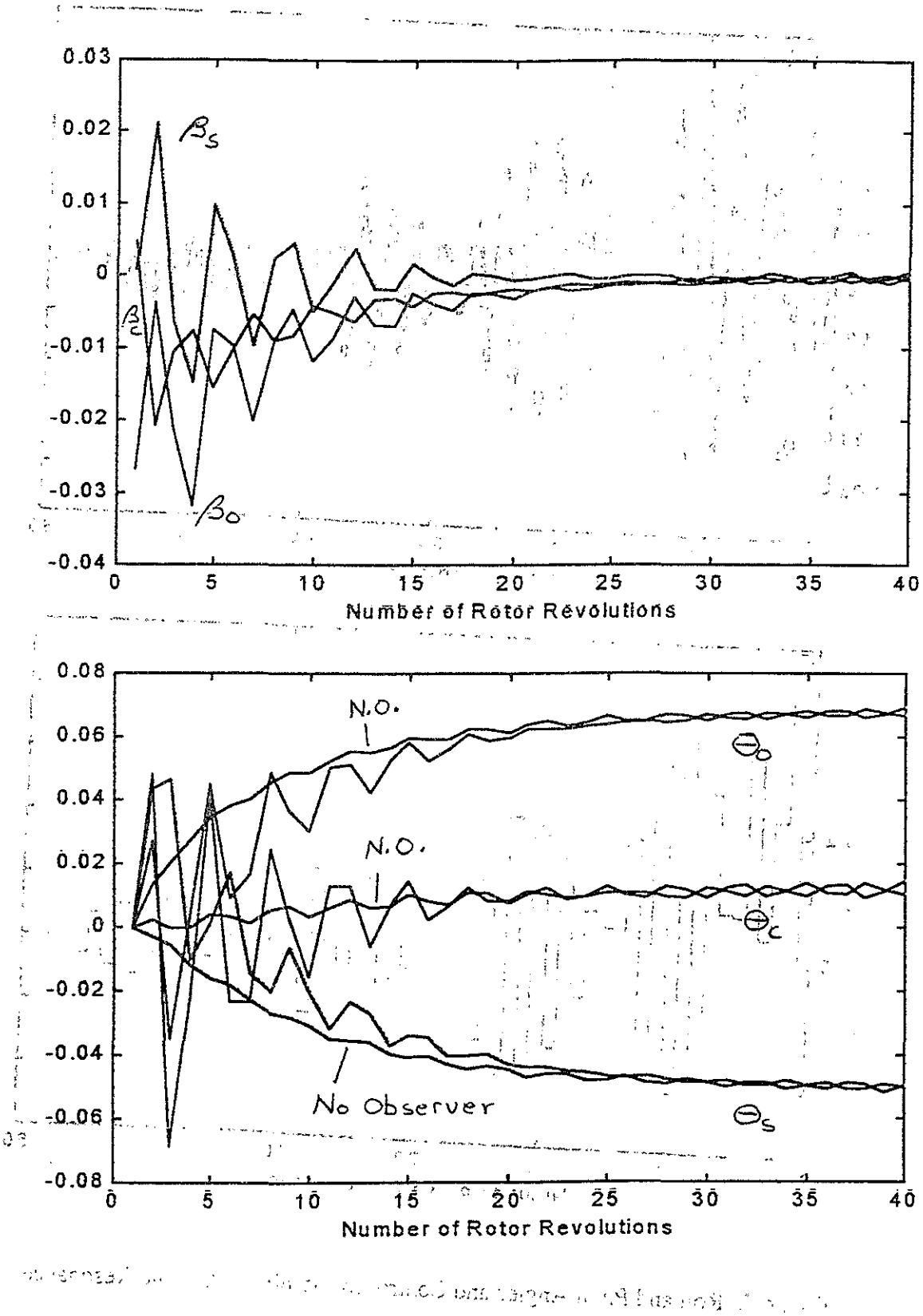


Figure 2. Trim Errors and Controls for Unstable F-L-T with Observer.
[Smoother curves on bottom are with all states measured.]

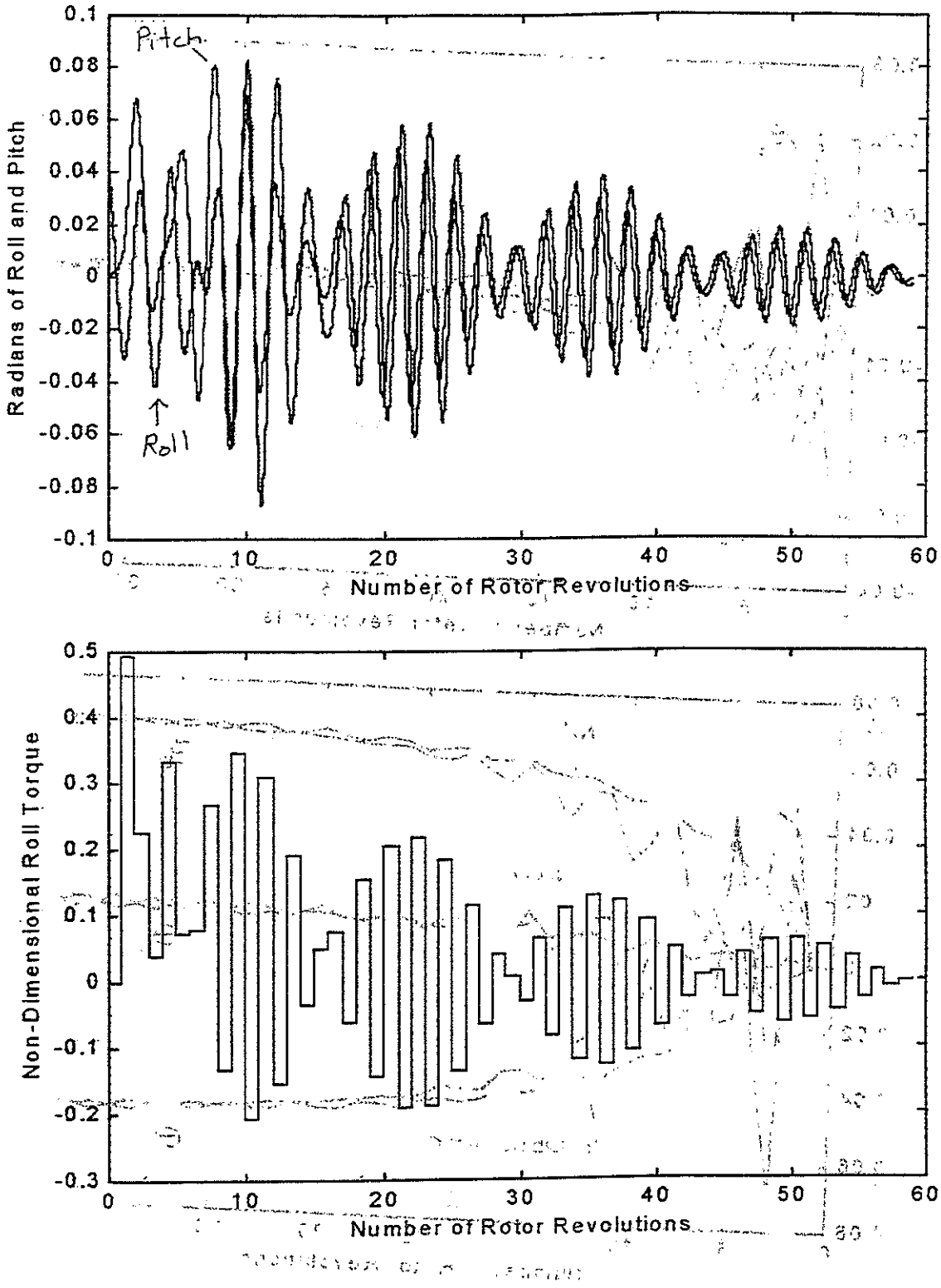


Figure 3. Roll and Pitch Angles and Control for Stabilized Ground Resonance.

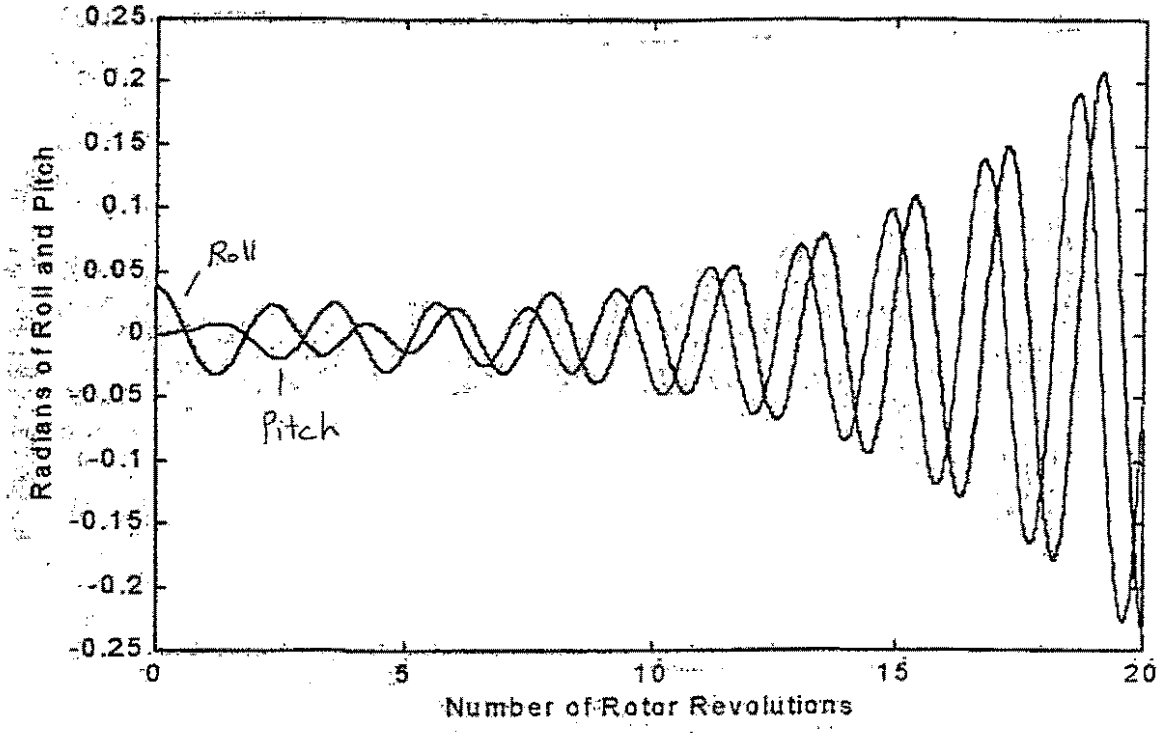


Figure 4. Roll and Pitch Angles for Ground Resonance Case with No Control.