N94-25101

Future Directions in Flight Simulation

- a user perspective -

Bruce Jackson
Aircraft Guidance and Controls Branch
18 March 1993

Motivation for this talk: I've been involved with aircraft simulations for eleven years (seven as a simulation support provider and four as a simulation user). I have rehosted several simulations from various sources in the first seven years, including the X-29A from Grumman/AFWAL, the AV-8B and F/A-18A from McDonnell-Douglas, and been involved in other simulation development efforts including the F-8 Oblique Wing Research Aircraft, the A-6F, X-31, V-22, F-4S, and F-14D. Since joining NASA I've been involved in developing and sharing a simulation model of the HL-20 with various sites.

It is obvious to anyone that has been involved in a shared simulation that an enormous amount of effort is expended in modifying the software and validating the result; and that there are as many ideas about how it should be done as there are Pratt & Whitney Aeronautical Vest Pocket Handbooks.

This proposal is a plea for help in resolving some of these issues; most of the ideas are not new. I've been encouraged and supported by the following people, whose help I would like to acknowledge: Bruce Hildreth of Systems Control Technology; Roger Burton, Buddy Denham and Jay Nichols of the Naval Air Warfare Center; Doug Sutton of SBE, Inc.; Tom Galloway of the Naval Training Systems Command; Larry Schilling, Marlin Pickett and Joe Pahle of NASA Dryden, who have at least taken an initial stab at solving this; Jerry Elliott, Carey Buttrill, Jake Houck and Dr. John McManus of NASA Langley; and W. A. Ragsdale of UNISYS.

Introduction

- Digital real-time aircraft flight simulations developed in late 1940s as training devices
- Reliance upon simulation-derived results has been growing, due to cost and safety advantages
- LaRC was early leader in sim technology
- Each facility developed own hardware/software architecture independently
- Emphasis has always been on hardware; software written as needed

Langley Research Center was an early leader in simulation technology, including a special emphasis in space vehicle simulations such as the rendezvous and docking simulator for the Gemini program and the lunar landing simulator used before Apollo.

In more recent times, Langley operated the first synergistic six degree of freedom motion platform (the Visual Motion Simulator, or VMS) and developed the first dual-dome air combat simulator, the Differential Manuevering Simulator (DMS).

Each Langley simulator was developed more or less independently from one another with different programming support. At present time, the various simulation cockpits, while supported by the same host computer system, run dissimilar software.

The majority of recent investments in Langley's simulation facilities have been hardware procurements: host processors, visual systems, and, most recently, an improved motion system. Investments in software improvements, however, have not been of the same order.

Concerns

- Simulation models of aircraft are increasing in number, detail, and importance
- Government, industry simulation facilities developed separate, dissimilar architectures
- Teaming arrangements require data exchange
- Few standards have been proposed to facilitate burgeoning simulation models
- Rehosting of flight dynamic models is tedious, labor-intensive, error-prone, inefficient

Reliance upon results from simulation experiments has become increasingly important as a result of improved simulation fidelity, increased flight hour costs, increased development time, and perceived safety-of-flight issues.

All aircraft manufacturing companies and most government agencies have their own simulation facilities. Unfortunately, due to historic reasons, most simulation facilities have evolved independently with dissimilar "architectures", or hardware/software environments - host computers, shared memory, variable names, sign conventions, iteration rates, real-time loop structures, and simulation control mechanisms and conventions.

Due to the immense risk and cost of developing new aircraft, and under economic pressure to reduce this cost, teaming arrangements between various manufacturers have become common, implying that these manufacturers share, to some degree, simulation models of the jointly-developed aircraft. Government oversight agencies likewise expect to receive simulation models of the aircraft during the development phase. However, due to the dissimilar architecture of the facilities, each exchange of a simulation model or software change requires a large manual effort to reformat data and code from one architecture to another, leading to the introduction of differences between the models. Resolving these differences is time consuming.

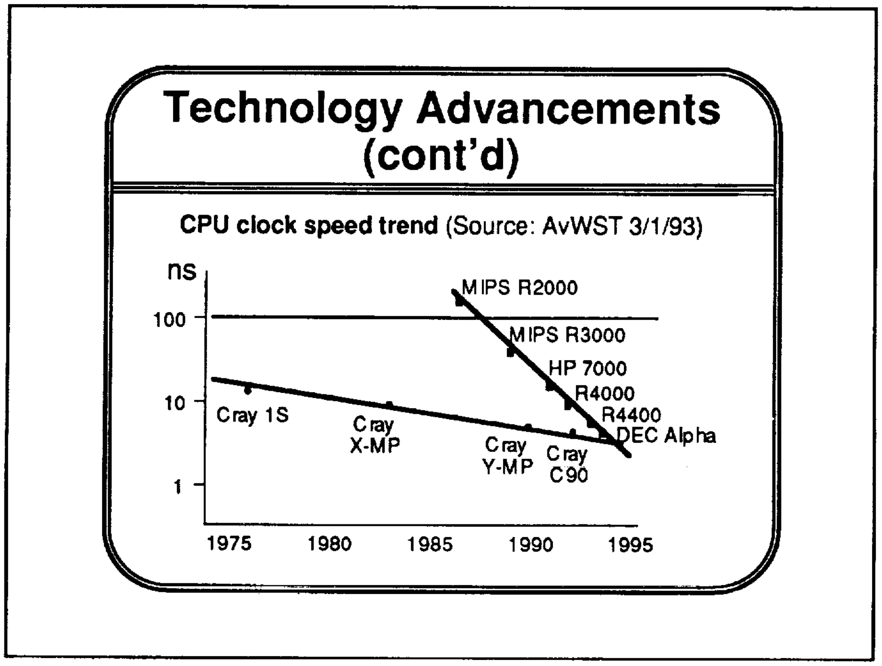
Technology Advancements

- CPU cost/performance improvements
- Data compression/interchange standards
- Internet access expansion
- Software engineering methods have matured

Driving the proliferation in simulation capability are rapid advances in computer technology. A laptop computer today has the power of yesterday's mainframe; a deskside or desktop computer of today outperforms last year's supercomputer. Desktop real-time high-fidelity simulation of rigid-body aircraft flight dynamics is now an actuality.

Moreover, the rapid interchange of large amounts of data, such as the aerodynamics model and dynamic check case data of a high-fidelity aircraft model, is common through wide area networks and data compression technology. Connections to world-wide pathways for data, in the form of the Internet, are growing at an ever-increasing rate. Same-day updates to simulation models are now possible, if the necessary standards for data exchange were in place.

Improvements in software design methods and languages - interface documentation, modular programming, object-oriented design, along with user friendly computer programming and execution environments - have improved the robustness and quality of most computer software. These modern software engineering methods are only now beginning to be applied to production real-time engineering simulation software.



This graph depicts the improvements in RISC technology computers, leading to an apparent capture of the supercomputer CPU performance benchmark - 10 nanosecond clock time, or 100 MHz CPU clock rate. In general, one can expect to be able to run real-time on anything faster than 10 MHz clock rate (10 million instructions per second, or MIPS).

Simulation Requirements

- Real-time execution @ 30 Hz requires ~10 MIPS
- "Large" memory storage for data (> 4MB)
- Pilot interface - display, controls
- Dynamic system models
- Dynamic system data interpolation
- Time history data for validation

To provide real-time simulation capability, a processor with at least 10 MIPS capability is needed, a specification that is exceeded by most RISC machines today. Another requirement, easily met in any modern computer, is at least 4 MB of memory, although lower amounts have been successfully used. The capability to perform 64-bit "double precision" floating-point operations is usually expected.

Some sort of pilot interface is needed, of course, since real-time operation implies a pilot is in the loop with the simulation. While a mouse and simple line grahics might represent the minimum capability for pilot controls and displays, some sort of quasi-realistic control stick, throttle, rudder pedals, and other controls are needed, as well as a realistic out-the-window and primary flight instrument displays. This requires the capability for four to eight channels of analog input and color shaded 3D graphics, executing at 30 Hz or faster. Cell texturing has been found to improve the realism of the visual scene as well. No more that 100 to 150 millisecond transport delay, in addition to model dynamics, can be considered adequate for a realistic visual cue.

The aircraft model, in order to be considered high-fidelity, must include a fairly detailed model of the vehicle flight systems - aerodynamics, propulsion, sensors, control system, weight and inertia model, and equations of motion software models are needed. If takeoff and landings are to be performed, a realistic landing gear model is also required.

Supporting these models are usually large tables of data, arranged by flight condition, that are interpolated in real-time. Check case data is needed as well.

Rehost Costs

- Human serves as interface between computers
- One man-year minimum to rehost and validate new aircraft simulation model
- One to six man-months to incorporate changes - basically a two-person job per simulation per site
- V-22: five simulation sites, five years: 50 man-years to maintain rigid body flight dynamics models alone

When a simulation model is transferred from one site to another, the most common scenario requires a simulation engineer to convert the software from the original format into one that is compatible with the receiving facilities architecture. The involves, as a minimum, "rewiring" the software modules (e.g., adding FORTRAN COMMAND and EQUIVALENCE statements or function/subroutine arguments such that the correct input and output variables are passed to and from each module); it usually implies considerable restructuring of the code to meet architecture needs - changing variable names, "sense", and units of measure (radians to degrees, for example). It almost always involves converting the typical table lookup data from one format to another and executing appropriate precompilers to generate function table routines or real-time data files.

Verifying proper implementation is tedious as well, due to dissimilar check case data formats. It is not uncommon to receive hardcopy plots of time responses in lieu of digital data; these must either be matched "by eye" or redigitized for overplotting purposes. Rigor and criteria in matching this data is left up to the interpretation of the receiving facility, in general. Each new release of data or models requires some element of this manual process.

The experience of the Navy's Manned Flight Simulator was to expect at least 12 man-months of labor to rehost a complete simulation, and usually one or two people were assigned full-time as "model managers" for a particular simulation. It is estimated that the V-22 simulation support staff, given the five entities involved (Bell, Boeing, Navy, NAS, and Hughes), approached 10 people just to keep up with changes in data releases during the DT/OT (dvelopment/operational test) period.

LaRC Issues

- Multiple real-time architectures
- Introduction of high performance workstation computers
- Language barriers
- Opportunity for new technology development
- Real-time data network in place

Due to historic reasons, Langley has three distinct simulation architectures running on two sets of host computers, leading to duplication of effort and cross-training of personnel. The equations of motion models are different, and have different variable names and units of measure.

Meanwhile, several user groups at Langley are developing independent real-time simulation capability with little or no commonality between them and the original real-time facility.

Language barriers exist: AGCB/FDB develops full vehicle simulations in Matrixx/Matlab; feeble autocode generators require nurture and constant attention to successfully generate real-time usable code. Hand generation of software from computer-generated wiring diagrams is common.

The opportunity to leapfrog into 21st century methods is here, if the needed resources are made available, resulting in potential industry benefit. Innovative cueing systems also being pursued by LaRC researchers.

NASA Issues

- Several different real-time architectures
- Proliferation of workstations
- Language barriers
- Real-time data network reqts

Langley is a microcosm of the simulation dissimilarity within NASA. Each NASA center has one or more simulation facilities, which are, by and large, dissimilar. Exchange of simulation models between any NASA facility requires manual rehosting.

National Issues

- Many real-time architectures
- Multiple host computers
- Language barriers

And the NASA problem is representative of the general industry problem: each simulation facility uses dissimilar architectures. Exchanging simulation models is not easily performed, with few exceptions.

Necessary Standards

- Standard data dictionary
  - Names
  - Sign convention
  - Units of measure
  - Precision (bytes)
- Equations of motion - standard inputs/outputs
- Standard partitioning of subsystem models
- Standard interfaces for other components e.g. cockpit display routines

What is needed are a set of standards for simulation data exchange. It is not anticipated that any existing facility will agree to adopt a simulation architecture developed elsewhere; too many resources have been expended in developing the existing architectures, and staff retraining is painful and expensive.

An evolutionary set of hierarchical standards would allow a gradual phase-in of the capability to exchange simulation models between facilities. The initial agreement would be on variable names, axis and sign conventions, and units of measure for commonly calculated variables, leading to a standard "data dictionary" that would be the basis for future simulation models, as well as an aid to translating to/from each facilities' variable name space. An agreement on where generic equations of motion and specific aircraft models would be delineated and how aircraft math models should be partitioned could lead to a standard set of inputs and outputs to/from the facility-supplied equations of motion and standard subsystem models (aero, engine, gear, controls, etc.) An agreement on headers for software modules would allow automated "wiring" of exchanged models into specific facility architectures; the ultimate would be to have a method of describing the math model that is not language specific.

To encourage commonality, a widely-accepted set of equations of motion that covers most forms of near-Earth flight could be made available to industry and academia that runs under most Unix platforms under X windows; these equations of motion would adhere to the standard, allowing easier mode interchange between existing simulation facilities and their support organizations and grantees.

Necessary Standards (cont'd)

- Dynamic system model interchange via ASCII
- Function data interchange standard
- Time history data
  - Large (>5 MB) files!
  - Should be self-documenting
  - Tied to flight test community/PID needs
- Memory mapping / real-time networking - SIMNET
- Automated validation - maneuver generator

The least common denominators in computer data interchange are 7-bit ASCII text files. An interchange standard for dynamic models and data should be based upon an agreement on how to encode and interprete dynamic systems in terms of ASCII characters. The resulting text file could be converted into facility-dependent real-time software or a number of block-based graphical editors.

Several attempts at this are underway to demonstrate this capability, including the Ames/Dryden SBIR contract with G & C Systems; at least one commerical control design software vendor has expressed an interest as well. Certainly a NASA-wide standard would be supported by major vendors of simulation and control design tools.

Necessary Standards (cont'd)

Wanted: Digital Aerospace Vehicle Exchange format:

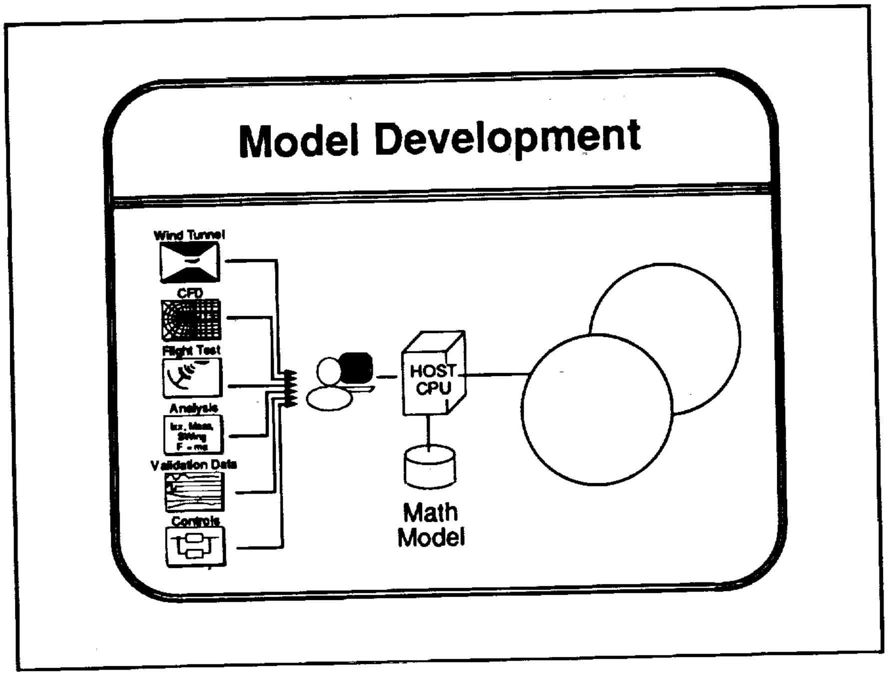
- Self-documented, complete data package
- Human readable documentation
- Subsystem data
- Subsystem models
- Validation data
- Hooks to include specialized data such as display formats

The ultimate goal of this standardization effort would be the capability to easily transport complete simulations across the Internet between dissimilar real-time simulation facilities, and successfully implement and validate the rehosted simulation with a minimum effort and time.

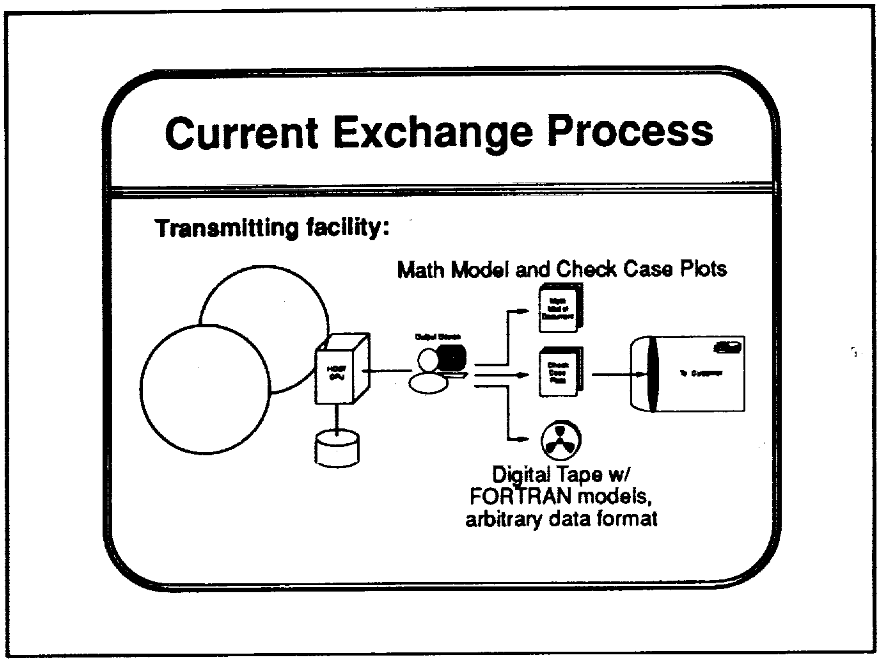
Benefits of Standardization

- Increase confidence in simulation predictions
- Improve configuration control
- Increase productivity by factor of ten

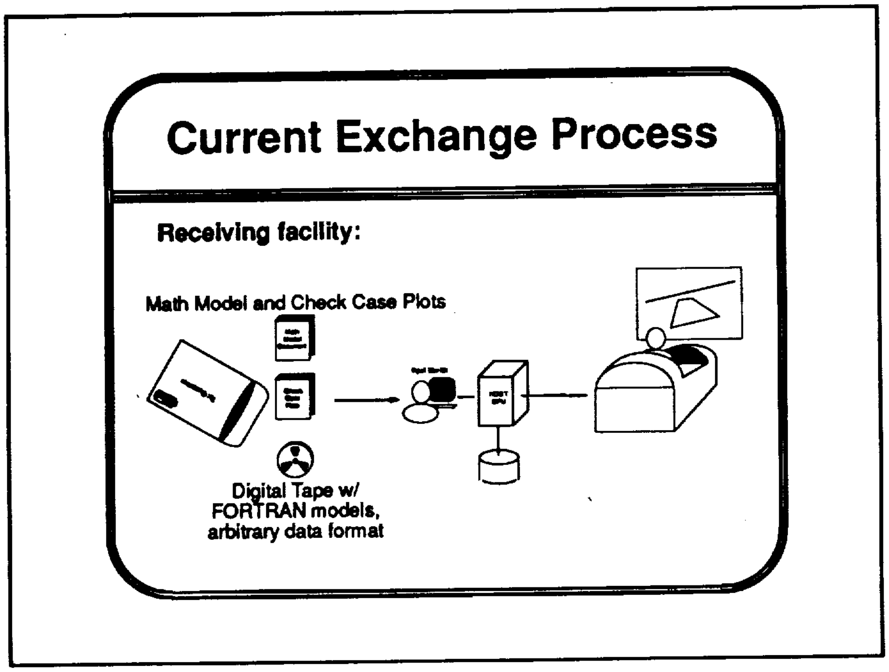
By reducing or eliminating the human element in the digital exchange of digital simulation models, an increase in productivity will result; in addition, configuration control of simulation models across facilities will be enhanced, reducing paperwork. As the inevitable difficulties are resolved and multiple successes are experienced, confidence in imported simulations will grow, making the sharing of complete simulation models commonplace. This will undoubtedly raise some security questions, however.



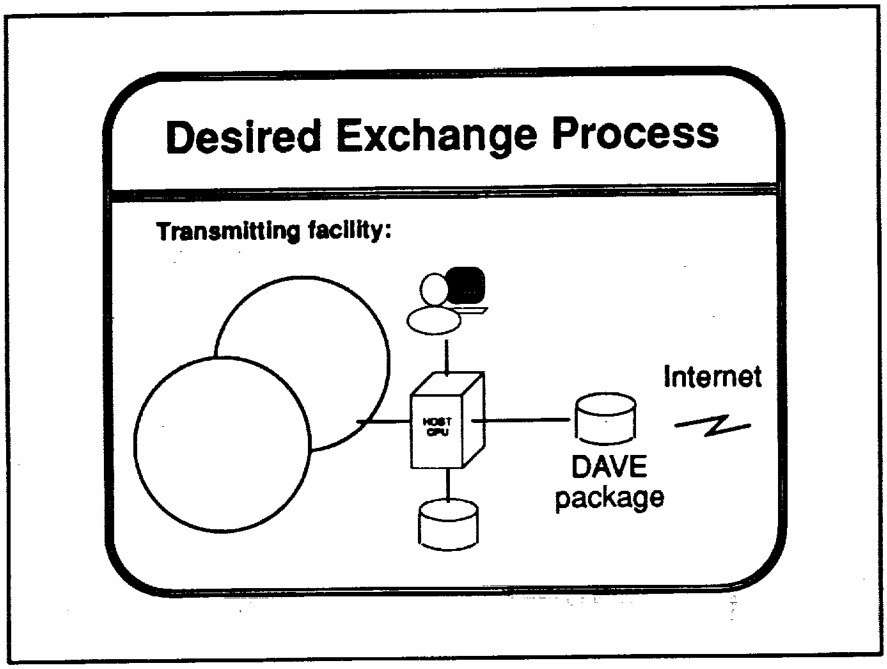
For reference, this is a simple-minded schematic of the current simulation development process, a very human-intensive operation. The only impact of the proposed standards would be to modify the end product to be amenable to exchange with other agencies.



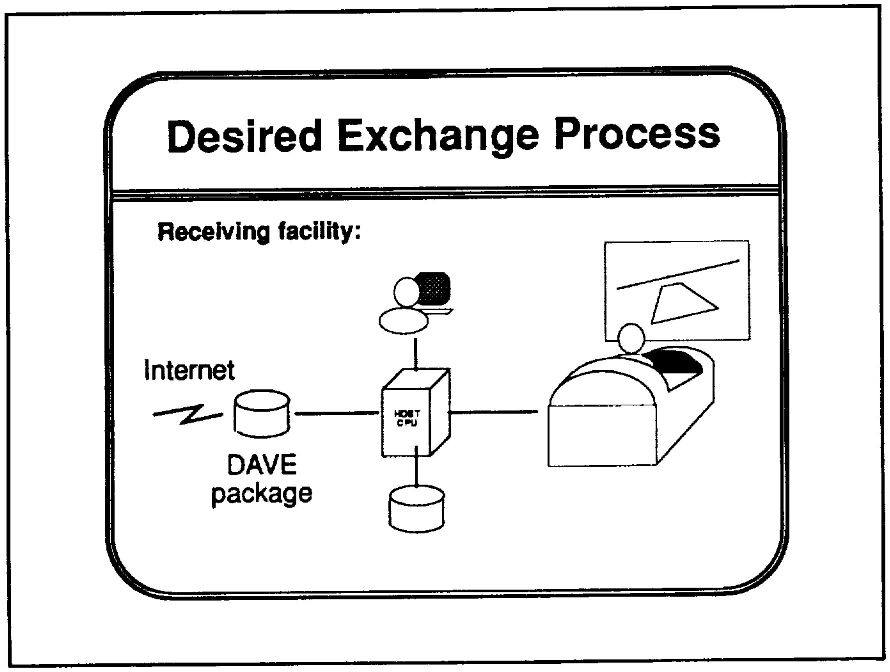
The current method of exchanging a simulation model is depicted in the next two figures. This shows the use of a human to generate, from the existing facility specific software, a set of listings, documentation, and a copy of the simulation "tape", although different media might be used. Dynamic check cases (time histories) are usually provided only in the paper documentation. Exchange of this data requires physical transport from one facility to another.



At the receiving end, another human is tasked with converting the software from the original facility architecture to that of the receiving facility, and validating the results. This is a six-to-twelve month process.



In the envisioned future, a post-processor converts the originating facility's model into a architecture-independent ASCII text file (or set of files). This package can be sent over the Internet to the receiving facility...



...where the package is run through another process to convert it into a model that can be run immediately on the new simulator facility. Some form of automated checkcase comparison should be a part of the exchanged data.

Implementation

- Langley community should develop Center-wide standards for data dictionary, wind tunnel data
- Solution for Langley could become NASA solution
- NASA solution would become industry solution
- AIAA adoption as national standard would follow
- Method would be to have each site write a translation program. This would NOT REQUIRE the redesign of existing simulation architectures!

Langley is in a perfect position to simultaneously improve its simulation architecture, resolve a Langley data exchange problem, and lead an effort to vastly improved simulation model exchange capabilities for the United States aerospace industry, with minimal impact on existing software and facilities.

Conclusions

- Traditional spending emphasis is on hardware components (COF)
- Real payoff will be from software improvements
- Simulation modeling standards would be valuable contribution to American aerospace industry
- Langley should take lead in standards development

PRECISION INFLIGHT MEASUREMENT OF TURBULENT AIR MOTION USING A SOLID-STATE COHERENT LIDAR

Phase I SBIR study NAS1-19872 by

Coherent Technologies, Inc.
Boulder, Colorado

prepared for

John Ritter
NASA-Langley Research Center
Hampton, Virginia

CTI

NAS802

PRECEDING PAGE BLANK NOT FILMED

OBJECTIVE:

Design an airborne, 2 micon pulsed, coherent solid-state lidar system that can accurately measure all three components of the turbulent velocity field at a point far enough in front of the aircraft to avoid its disturbance (i.e., 10 m or greater)

METHOD:

Use of a conical scan of the optical system focal point along a helical path traced out by the translation of the aircraft. Atmospheric aerosols or cloud droplets provide the desired air motion tracers.

APPROACH:

Design parameters will be determined using detailed lidar simulations. Doppler spectral analyses, time gating, and/or amplitude thresholding will be used to separate the desired returns from aerosols in the focal volume from interfering returns due to large hydrometeors (rain, snow, etc) which may not follow the local air motion.

PHASE I PROGRAM OBJECTIVES

- Performance and error analyses
- Airborne system requirements
- Preliminary design
- Experimental validation of error models

CTI

NAS802

REQUIREMENTS FOR TURBULENT FLUX PROBE

o Measure relative wind vector just ahead of aircraft

o Measure in airflow undisturbed by aircraft body

o Achieve high measurement accuracy (0.05 m/s)

o Measure of all three vector components

o Sample at rate adequate to resolve 5 meter turbulence scale

o Altitude requirement : 0 to 1 km; 0 to 10 km desired

CTI

NAS802

ERROR SOURCES

- photon shot noise
- aerosol inhomogeneities
- wind field inhomogeneities
- speckle
- platform vibration, flexing
- precipitation ' slip '
- backscatter from optics, distant objects

CTI

NAS802

SYSTEM PARAMETERS

Focal length

Aperture

Wavelength

Pulselength

Pulse energy, repetition frequency

Doppler algorithm - radial component

Cone angle, PRF, vector retrieval algorithm

CTI

NAS802

PULSE REPETITION FREQUENCY

o optimize to get best SNR consistent with minimum backscatter levels expected

o need at least 3 pulses per resolution element (5 meters at 100 m/sec ==> 60 Hz)

o Increase pulses per sample to verify wind homogeneity

o Increase pulses per sample to improve velocity precision

CTI

NAS802

RELATED AVIONICS APPLICATIONS

o Laser airdata probe to replace conventional pressure sensors
  - precision measurement of TAS, AOA, AOS
  - rapid update for automatic flight control systems

o Laser local wind shear sensor
  - extension of airdata probe to measure local wind shear <u>rates</u>
  - input to pilot and/or control system

o Laser local vortex wake sensor
  - sensing of nearby wake vortex
  - warning to pilot, input to automatic control system during takeoff and landing

CTI