In-Flight Simulation of High Agility through Active Control
-Taming Complexity by Design-

Gareth D Padfield
Defence Research Agency
Bedford, England

Roy Bradley
The Caledonian University
Glasgow, Scotland

Abstract

The motivation for research into helicopter agility stems from the realisation that marked improvements relative to current operational types are possible, yet there is a dearth of useful criteria for flying qualities at high performance levels. Several research laboratories are currently investing resources in developing second generation airborne rotorcraft simulators. The UK's focus has been the exploitation of agility through active control technology (ACT); this paper reviews the results of studies conducted to date. The conflict between safety and performance in flight research is highlighted and the various forms of safety net to protect against system failures are described. The role of the safety pilot, and the use of actuator and flight envelope limiting are discussed. It is argued that the deep complexity of a research ACT system can only be tamed through a requirement specification assembled using design principles and cast in an operational simulation form. Work along these lines conducted at DRA is described, including the use of the Jackson System Development method and associated Ada simulation.

Introduction

The central issue when setting requirements for in-flight simulation involves the trade-off between performance and safety. The integrity of the experiment, from the very concept being tested through to its implementation in software and hardware, determines the achievable flight performance level. The greater the uncertainty in the behaviour of the simulated aircraft, then the greater the risk of misbehaviour; likewise, the lower the reliability of the experimental system, then the greater the risk of failure and consequent misbehaviour. It follows that the higher the inherent performance of the aircraft and its experimental system, the higher is the risk that misbehaviour will lead to an accident. Operational constraints and regulations usually dictate that this dilemma is resolved in favour of safety, hence compromising performance, or making it very expensive to achieve. These ideas are not new of course, and have featured large in the aircraft systems field for many years; the disciplines of modern design, test and implementation methods now ensure a degree of confidence in solutions to well defined problems. The compounding dilemma is that research into new and improved flying qualities contains the problem definition itself, and defining the flying qualities boundaries requires gathering data with Level 2 and 3 configurations.

The development of full authority, flight critical, active control technology (ACT) for helicopters has been proceeding apace for more than ten years with nine experimental aircraft in the form of research and technology demonstrators having flown in the western world. In the search for the quantum change in helicopter flying quality, a variety of solutions to the performance/safety tradeoff have been employed, including constrained experimental flight envelopes, multiple redundant hardware and limited performance actuation systems. All experimental systems have employed a Safety Pilot whose cockpit controls are back-driven, providing the primary cue on the behaviour of the system; experience has shown that the Safety Pilot is the most critical safety element. Along with ground-based simulators, these first generation variable-stability, active control helicopters have been used extensively to explore novel control methods and to build the database from which the ADS33C flying qualities criteria have been developed and substantiated.

Several Nations are now looking forward and planning the development of second generation ACT helicopters with a range of new research objectives in mind, centred on the need for greater levels of automation;

i) to extend operations in degraded visual cue environments,

ii) through the provision of carefree handling, enabling safe exploitation of the full operational flight envelope (OFE),

iii) through the integration of flight with fire control, engine control and mission systems to provide greater concurrency and hence operational effectiveness.

Research into these aspects of helicopter ACT needs to deliver solutions that will increase performance and safety in harmony. Ironically, as noted above, when exploring a new idea in flight, performance and safety attributes can conflict, and there is a potential problem that development of ACT and its operational benefits will be hindered by this dilemma. Recognition that a certain level of risk is inevitable is the first step towards resolving this problem; establishing well formulated operating procedures that contain the risks during the exploration of new concepts is the second. Adopting an approach to specification and design, that tames the complexity of the integration of the flight control system with the vehicle, its subsystems and

the pilot, is the important third step in this process and will feature as one of the key themes of this paper.

The paper reviews the UK DRA (formerly RAE) programme to define the requirements for and to build a high performance flight research system, designated ACT Lynx. Taking the performance/safety tradeoff as a starting point, a number of topics are addressed.

1) The performance requirements and the driving research objectives will be outlined; the emphasis from the outset has been to achieve high agility at low pilot workload.

2) The safety constraints and how they reflect on system architecture and airframe health will be addressed. The role of the safety pilot will be described and issues surrounding intervention times following failures will be addressed, drawing on results from an exploratory ground-based simulation conducted at DRA. Experience with other experimental ACT helicopters are discussed and (non-attributed) examples of the kind of failures that safety pilots have had to cope with in the past will be highlighted.

3) A vital key to confidence that an experimental flight control system will perform as required lies in the development of the functional requirements as an integral part of the system design. This has been achieved in the ACT Lynx project by the incremental development of an Ada simulation of the triplex redundant system using the Jackson System Development (JSD) methodology. The approach focusses attention on the interface of the experimental system with the outside world, eg operations at the pilot vehicle interface (PVI), the actuation system, sensor system etc. The behaviour of the system is considered from a constructional/design, rather than a hierarchical/descriptive, viewpoint. This distinction is crucial at an early stage to capture all the nuances of the intended behaviour. In addition, many of the human factors issues at the pilot/vehicle interface can be examined in detail through simulation. This approach is described.

4) The methodology for control law design and assessment is described. An important concern is the validation of the behaviour of the implemented control law; early in its life it will be immature and made up of several, limited flight-envelope, un-integrated functions. The development towards continuous, full flight envelope, agility enhancing control functions involves a gradual expansion of the envelope and actuator authority, using ground based simulation to pave the way for the flight tests. The philosophy will be described, including the role of the curtain limiter, a device for moderating the control inputs to the experimental actuators.

The UK programme is currently at a hiatus due to funding limitations, but sufficient ground has been covered to provide some clear messages for others striving for similar goals. The UK continues to collaborate with the key players in the research field - US Army/NASA, NRC and DLR - and this paper presents the opportunity to stimulate discussion, with the wider manufacturing and research community, on some of the trade-offs in this important area.

Harmonising Safety and Performance

Research Objectives

A companion paper at this Conference (Ref 1) has highlighted situations where current operational helicopters lack agility, such that when operated at high performance levels, flying qualities deteriorate and lead to high piloting workload. Figure 1 reflects this through the variation in pilot handling qualities ratings (HQR) with Agility Factor - the ratio of ideal task time to actual task time in a mission task element (MTE). As the pilot increases performance, the degradation from level 1 to poor level 2/level 3 ratings is rapid, making the use of high performance potentially quite dangerous.

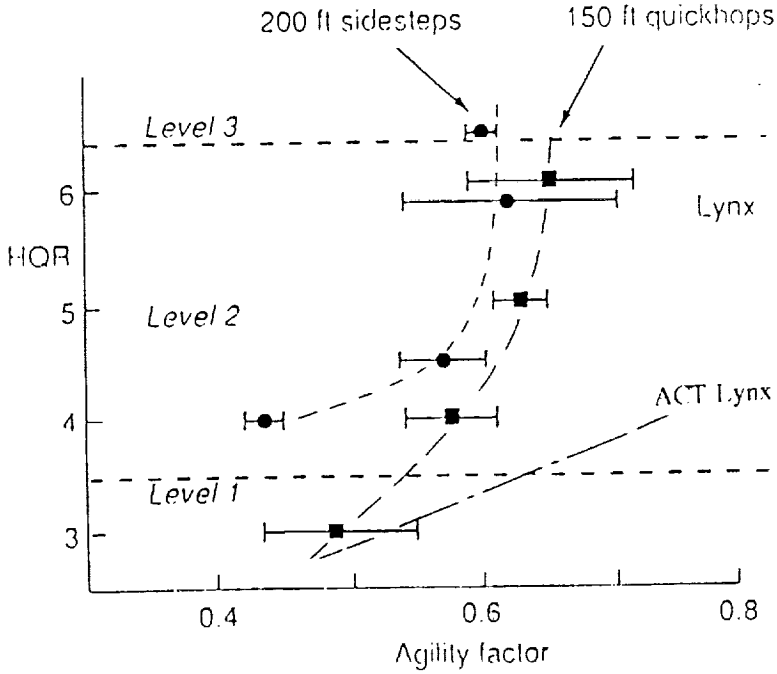

Fig 1 Pilot Handling Qualities Ratings vs Agility Factor for Lynx

The results shown in Figure 1 were gathered on the research Lynx at DRA Bedford, flown at much lighter weights than in normal operational Service, to simulate the higher performance margins expected of future types; the results are considered to be typical of all current Service aircraft and indicate a clear goal for research into improved flying qualities. A primary objective of ACT Lynx was therefore aimed at demonstrating the achievement of Level 1/2 flying qualities at high agility factors as shown in Fig 1. This and other key research objectives are summarised in question form as follows;

1) Can level 1 flying qualities be achieved at high agility factors? Research to answer this question would produce a database from which carefree handling functions could be defined and potential upper flying qualities boundaries identified.

2) Can multi-axis sidesticks be used effectively in such circumstances and what level of automation is required to facilitate their use? This research would address the ergonomic aspects of sidesticks and define the optimum feel characteristics and sensitivities; it would also address the use of such controllers with reversionary, less well augmented, modes.

3) Can high performance be achieved in the presence of strong disturbances? Disturbance rejection and ride-control functions can be designed to operate effectively at considerably higher bandwidths than handling-control functions and this research would define those control functions and associated sensor requirements.

4) What are the critical control augmentation/display trade-offs in degraded visual conditions? Research would address the integration aspects of displays and response types for different usable cue environments (UCE), blending issues and identify critical parameters in the controls/displays trade-off.

5) How can ACT be exploited to enhance functional integration between the flight control system and mission systems eg fire, engine, navigation? This question would direct research towards maximising concurrency between the flight and mission management systems, leading ultimately to the potential for fully automated flight.

Objectives 1, 2 and 3 require the high-fidelity environment of an in-flight simulator, able to operate in realistic scenarios close to the visual-cue-rich environment of natural terrain and cover, whereas considerable progress towards Objectives 4 and 5 can be made with ground-based simulation. In addition, the displays and integration research require considerably more on-board equipment. Hence the initial foci of ACT Lynx were to be the three high performance objectives.

Performance & Safety - The Conflict

The operational flight envelope for the Lynx Mk 7 represents the baseline ACT Lynx envelope. Key features are given in Table 1. The high values of attitude quickness and bandwidth stem from the hingeless rotor on the Lynx with its 13% effective flap hinge offset. The rotor provides a high natural damping and control moment capability enabling higher levels of agility to be exploited than with articulated rotor helicopters. Figure 2 illustrates the envelopes of roll and pitch quickness achieved in the Lynx for Sidestep and Quickhop re-positioning MTEs (Ref 2). The envelope covers the full attitude range to illustrate the high bandwidth (low amplitude) and control powers (high amplitude) achieved even in these, non-tracking, MTEs.

Table 1 ACT Lynx Performance Characteristics

Performance Aspect	Lynx Mk 7 Flight Envelope for ACT Lynx
hover thrust margin	> 20% (sea level, 20 deg C)
roll, pitch, yaw control power	> 100deg/s, 60deg/s, 60deg/s
quickness for 10deg attitude change	> 4 rad/s (roll), 2 rad/s (pitch)
attitude bandwidth in hover	> 5 rad/s (roll), 3 rad/s (pitch)
low speed side velocity envelope	30 kn
load factor	> 2 g, 0 g
Vmax	> 140 kn (sea level, 20 deg C)

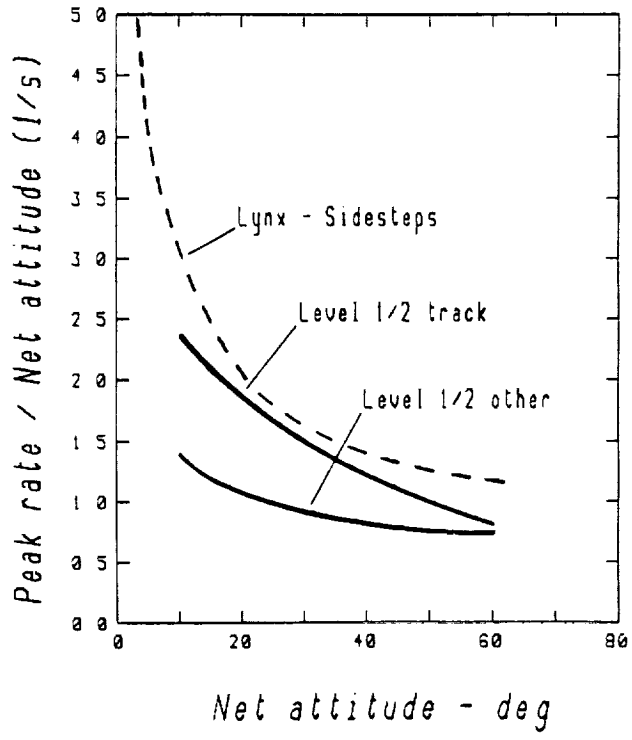

Fig 2a Roll Attitude Quickness

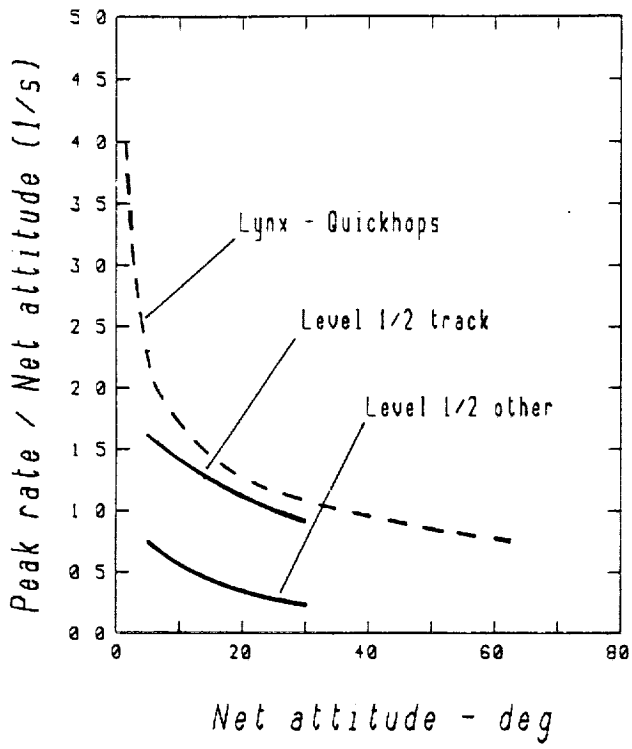

Fig 2b Pitch Attitude Quickness

The quickness is a direct measure of agility, closely related to the time to achieve an attitude change. At the two amplitude extremes the achieved quickness values are well above the ADS33C Level 1 requirements for bandwidth and control power and there is a generous margin in the moderate amplitude range, even relative to the tracking MTE boundary. Combined with a moderate hover thrust margin, maximum 'g' capability and wide speed envelope, these performance characteristics make Lynx well suited as an ACT testbed. But the performance is only useful if control laws are able to exploit fully the OFE and this raises fundamental safety issues concerning the aircraft behaviour following ACT system failures.

System failure can be loosely classified under two categories;

i) hardware failures; these are usually assumed to be random in nature, hence only predictable in a statistical sense, eg one failure expected within 10^n operating hours. The usual method of protecting against such failures is to build in hardware redundancy together with comparators and monitors, effectively to increase n.

ii) software failures; two ways that a software implementation can ***'fail'*** or misbehave follow from either the correct programming of the wrong reaction or failure to take certain situations into account. It is sometimes claimed that the probability of a software error occuring can be related statistically to the degree of testing carried out, but this does not appear to have a sound theoretical foundation. In reality, both the above software failures are deterministic and context dependent and unless the testing happens to include the particular conditions, the error is likely be missed.

The Safety Pilot

Failures in both categories can be expected to occur throughout the life of an ACT research vehicle and give rise to a variety of different behaviour including fast/slow hardover, oscillatory or frozen actuator demands. Acknowledging this, the next set of questions relate to the integrity of the system, the related tolerance to failures and the means of protection. All ACT research helicopters operated over the last ten years have included one principal element in common in this regard - they have all had a **Safety Pilot,** whose controls are back driven by the research actuators. The latter have either been special purpose, dual mode (electro-mechanical) type (Refs 3, 4, 5) or connected in parallel with existing power control units (Refs 6, 7, 8). All types have been full authority, high rate actuators. The safety pilot, with his backdriven controls providing an immediate and instinctive cue as to the health of the system and the experiment, is generally regarded as the most important and vital safety element. A well trained safety pilot will be able to identify misbehaviour through the motions of his backdriven controls, and can take rapid action to preserve flight safety. However, very special skills are required to make a good safety pilot, among which is the ability to judge when, and when not, to disengage and how to recover to a safe flight condition. It is a very demanding role and any help that the system can provide will reduce the workload and lessen the risk of a loss of control.

Help can be provided in the form of a fail-safe or fail-operate system configuration. Fail-safe normally relies on a monitor system running concurrently with the flight control system, either sampling and comparing dual channels or comparing the signals in a single lane with that from a model. If the comparator detects a difference, outside a defined threshold, the system will be tripped out and control will be returned to the safety pilot with appropriate alert signals. Fail-operate signifies that the system can continue operation following one or more failure; through monitoring and voting, faults can be detected and isolated. The remaining healthy system components continue to function as normal, but the crew is alerted to the fault. For a single fail-operate system, the system degrades to fail-safe following a failure. Operational fly-by-wire fixed wing aircraft are normally designed with a two fail-safe capability with respect to hardware failures to achieve the necessary overall system integrity. This requires a triplex-monitored or quadraplex system architecture. The research helicopters operated over the last ten years have a variety of different solutions implemented. The NRC's Bell 205 (Ref 3) and DLR's BO105 (Ref 6) are both single string systems with a limited fail-safe capability centred on the fly-by-wire actuator input/output relationship. Rotor flapping is monitored in the 205 and hub moment in the 105 with

both having limits which, if exceeded, trips the systems out. The ADOCS demonstrator (Ref 4) included a triplex fly-by-light hardware configuration and an independent (analogue) monitor. The latter was designed to model the behaviour of the primary flight control system (PFCS), hence automatic flight control system (AFCS) inputs were signalled as errors by the comparator; the thresholds were set to allow moderately aggressive flying. This, so-called DOCS monitor, was designed to catch software and other common mode 'failures'. The AV05 research aircraft (Ref 8) comprised a dual-duplex architecture providing, in principal, a two-fail operate capability. The concept included flight envelope limiting features within the control system. Most of these aircraft also featured a trip when the engine/rotor system torque exceeded a prescribed value.

From this very brief review of some of the current designs it is clear that help can be provided to the safety pilot in a multitude of ways; it is also clear that current wisdom suggests that he does need help, particularly in the detection of rapid, potentially rotor damaging, control inputs. The dilemma comes from trying to distinguish between a system failure and a genuine ACT system command; both can look very similar at the actuation stage. Failures from hardware faults can be detected and isolated through fail-safe or fail-operate architectures; software failures are considerably more difficult to detect. As noted above, software errors in both the categories discussed above are likely to be a regular occurance in the development of a control law. Examples (non-attributed) of software failures that have occured on ACT helicopters include,

1) 3-axis hardover caused by divide by zero - excursions of 20 deg pitch, 35 deg roll and 20 ft height loss during recovery,

2) control modes not referencing to correct flight condition, leading to position error and roll into turn,

3) integrators not inhibited at control stops, leading to time delay in response to following input,

4) no priority given when engage/disengage pressed simultaneously

All led to a transfer of control to the safety pilot, although there was inevitably some delay in recovery due to failure recognition problems. It should be stressed that no accidents have occured on ACT research helicopters to date.

Safety Pilot Simulation

To gain a better understanding of the kind of behaviour that Lynx would exhibit in response to failures and the resulting safety pilot reaction, an exploratory simulation trial was carried out on the Advanced Flight Simulator at DRA (Ref 9), using the small motion system. A Lynx, augmented with an ACT system, providing Level 1 flying qualities, was flown through a range of mission task elements. The safety pilot occupied the cockpit on the motion base, with the 'evaluation' pilot flying from the control desk.

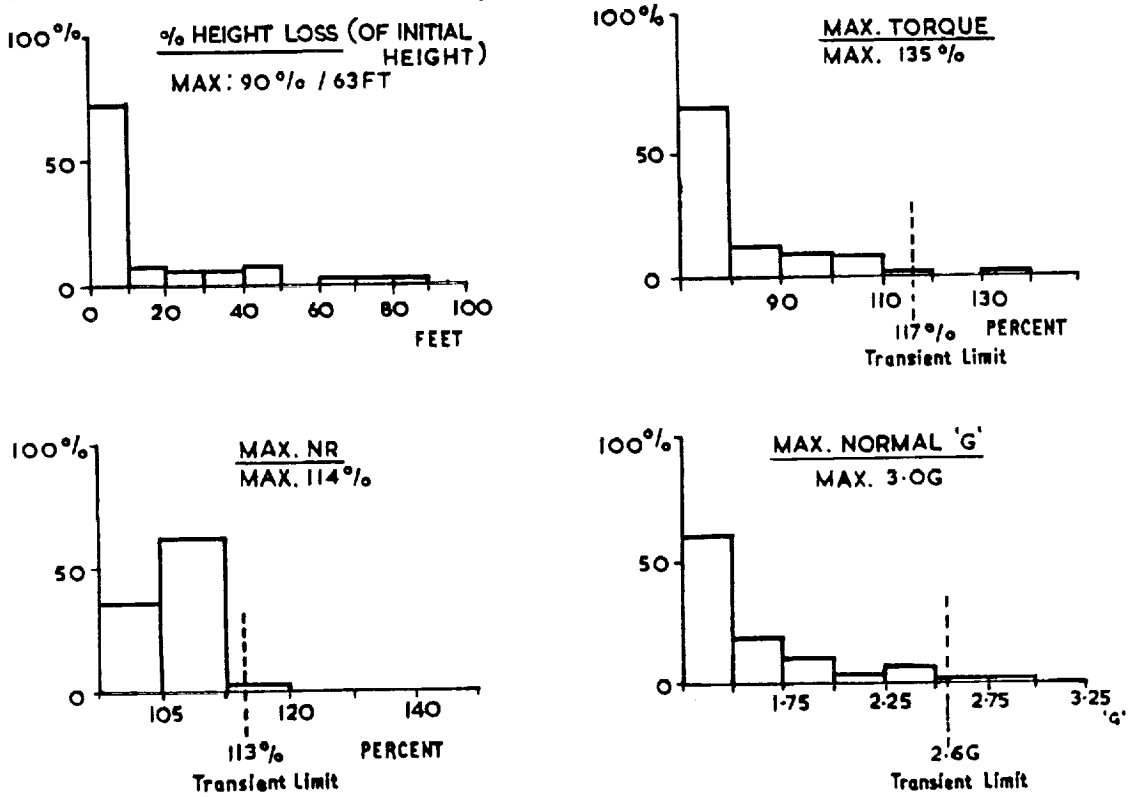

Fig 3 Statistical Summary of Excursions During Failures and Safety Pilot Recovery

Hardover failures were injected in combinations of axes at various points during the flying, and the safety pilot's task was to disengage the ACT system and recover the aircraft without exceeding limits and, of course, avoid the ground and obstacles. Following disengage, the aircraft configuration was Lynx with limited authority stability augmentation, as envisaged for ACT Lynx. This initial investigation had several related objectives including an evaluation of alternate disengage and alert mechanisms. A total of 61 failure events were flown with three evaluation pilots. With the preferred 'force' disconnect system, all disconnects were achieved in less than 0.3 second; 'button' disconnects resulted in longer times, up to 1 second. Figure 3 shows a statistical summary of the peak values of critical aircraft states recorded during recovery relative to the flight envelope limitations, including the height loss. The 3g peak occurred following a right cyclic/pedal runaway in a right turn, when a height loss of 63 feet was also recorded. The load factor limit was exceeded on this occasion to avoid hitting the ground. The main rotor torque and rotorspeed limits were both exceeded once, the former following a sympathetic positive collective failure in a bob-up. The results of the work reported in Reference 9 are tentative. The AFS simulation cues were limited and the Lynx aircraft model has known deficiencies particularly in the off-axis responses and in hard turns. Also, worst cases may not have been evaluated and instinctive, trigger disengage mechanisms were not evaluated. Nevertheless, the potential for very rapid flight envelope exceedances during failures, when operating close to limits, was demonstrated and the dangers of vertical flight-path excursions during recovery were highlighted.

Protection Devices

Protection against such occurrances needs to take into account that responses to failures can be similar to the response to an aggressive pilot-input applied to maximise agility. An approach used in the past has been to restrict the inputs to the rotor through employing both limited authority series actuators (as normally found in a conventional SCAS) and parallel actuators with reduced rates. Figure 4 illustrates the roll kinematics and pilot's lateral cyclic command during a sidestep manoeuvre on a phase plane. The shaded areas correspond to the excluded region if series/parallel, frequency-splitting, actuation had been used with typical 20% (20%/s) authority. The manoeuvre would have been severely compromised. Fig 5 illustrates the control/actuation quickness or 'attack' for the Lynx sidesteps showing values up to the PFCU bandwidth of 15 rad/s at small amplitude and quite high values extending out to large control inputs. The superimposed lines correspond to boundaries set by different actuation rates. The Lynx actuation system is able to achieve values greater than 200%.sec in single lanes. Any actuation rate limiting below this would clearly deprive the pilot of performance, but no systematic investigation of this aspect was carried out. Actuation limiting in such a crude manner can be effective but needs to be implemented in software if the limits are to be extended as confidence grows in the behaviour of a control law. This is effectively what happens with ADOCS, although in that implementation (Ref 4) the DOCS monitor tripped the ACT system out if rates and amplitudes from the AFCS were too high. For ACT Lynx, a scheme based on this approach was suggested, illustrated conceptually in Fig 6. The so-called 'Curtain Functions' would be defined in the software that limited the actuator inputs as shown in Fig 6. Initially, for a new control law, the curtain would be well closed, offering maximum protection following failures. As the control law developed and confidence grew in its behaviour, the curtains would be opened incrementally, until full performance was available. The concept has yet to be evaluated in simulation but potentially offers a safe route through to high agility.

As noted earlier, the ACT helicopters that have been operated over the last 10 years have adopted many different approaches to this protection question. It is believed that three main factors contribute to the 100% safety record in the operation of research ACT helicopters.

a) the reliance on an experienced, well trained and highly skilled safety pilot

b) the adoption of operating procedures that emphasise flight safety

c) the use of flight envelope monitors or restrictions that inhibit agility, particularly in low level trials.

For ACT Lynx, it was always considered that the practices in categories a) and b) developed by organisations like DLR, NRC and NASA would be fully adopted. The focus on agility research, however, meant that issues associated with c) had to be faced squarely and an alternate strategy developed that enabled a way forward. A fail-operate/fail safe (FOFS) architecture was selected to provide full protection against hardware failures, with the argument that in safety critical situations, even the safety pilot may not have sufficient time to recover with only a fail-safe system. Methodologies that ensure comprehensive verification and validation of the software system elements would be vigorously pursued. It was recognised that there would be two components to the embedded software, a high integrity 'core', including consolidation, monitoring, voting and actuator drive functions that would remain essentially fixed during the development of a control law, and the control law itself and its attendant curtain function, that would regularly change in structure and data input. The control law was envisaged as the most appropriate place for the envelope limiting to be incorporated, in the form of carefree handling functions. Ultimately, the control law would need to function without independent monitoring, to enable the high agility testing to be realised. For both kinds of software it was considered that a high investment in the requirements capture and definition process would pay off in high system integrity; these issues are developed further in later sections.

Airframe Fatigue Usage

Before discussing these aspects, there is one additional consideration regarding safety that was addressed with ACT

Lynx - the question of the impact of ACT flying on airframe fatigue. It was always recognised that an agility research aircraft would spend a greater proportion of flight time in high fatigue-usage manoeuvres, than its operational counterparts. Also, the effects of the ACT control functions on control linkage and rotor loads was relatively unknown. A third issue stemmed from the recognition that the existing aircraft's OFE was defined with a margin relative to the safe flight envelope and that carefree handling functions would, in principle, allow some of this additional performance to be used with safety. Some form of load monitoring in this regime would be essential. The critical structural areas were identified by the manufacturer and comprised components on the main/tail rotor hub and blades, control links, fuselage frame and gearbox, tail cone and fin. These components have since been strain-gauged for non-ACT purposes and are undergoing in-flight calibrations at the time of writing. The data from the strain gauges are processed in two different ways. First, via a telemetry link to a ground station to enable real-time monitoring of loads and, second, to the on-board recorder system for post-flight analysis and fatigue usage calculations. From a safety standpoint, the fatigue usage monitoring task was seen as an integral part of the comprehensive approach taken with the ACT Lynx concept.

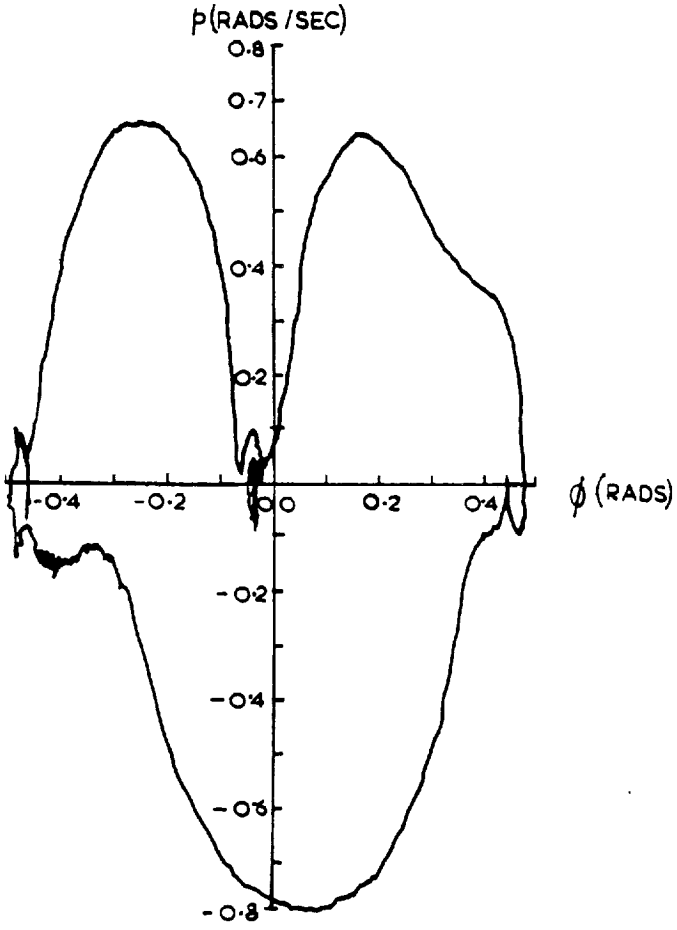

Fig 4a Lynx Roll Attitude Phase Plane Portrait in a Sidestep MTE

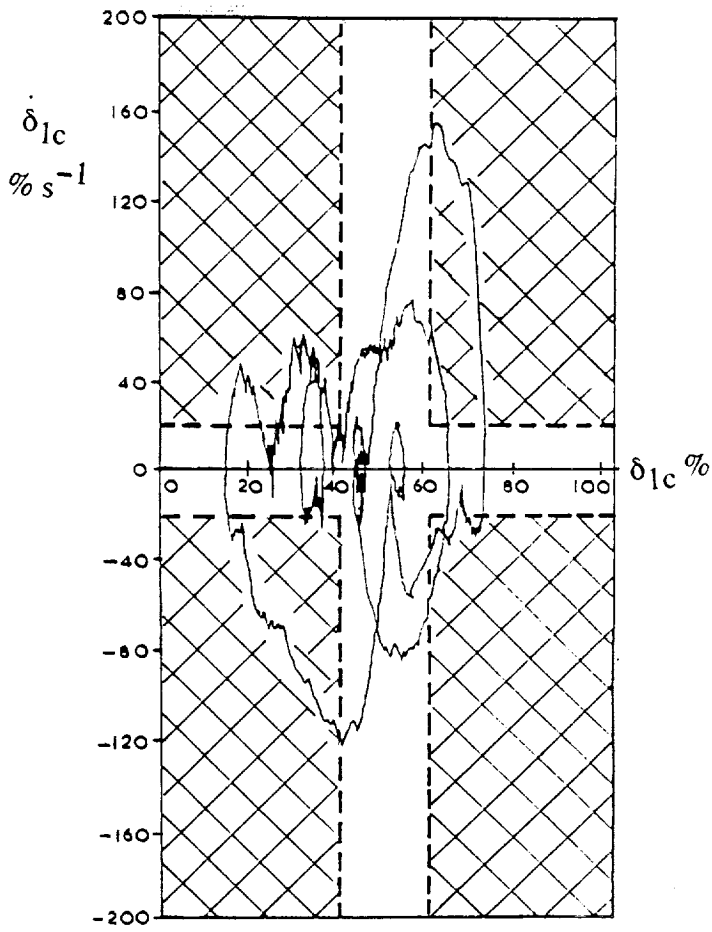

Fig 4b Lynx Roll Actuator Phase Plane Portrait in a Sidestep with Frequency Splitting

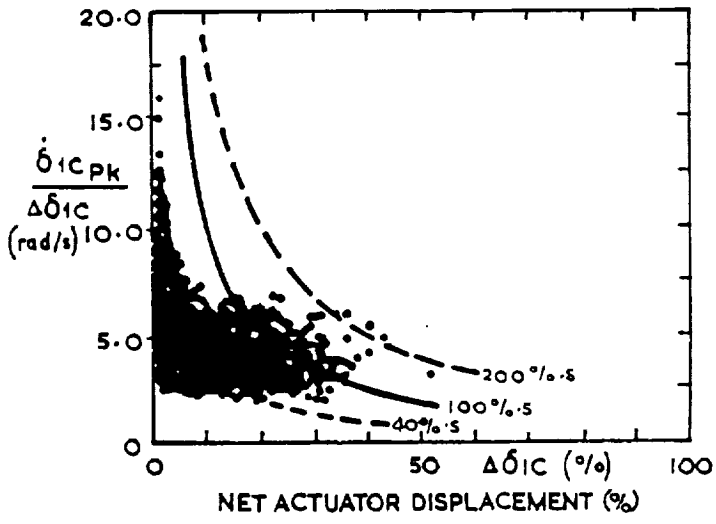

Fig 5 Lynx Control Quickness in Sidestep MTEs Showing Effects of Rate Limits

C-3

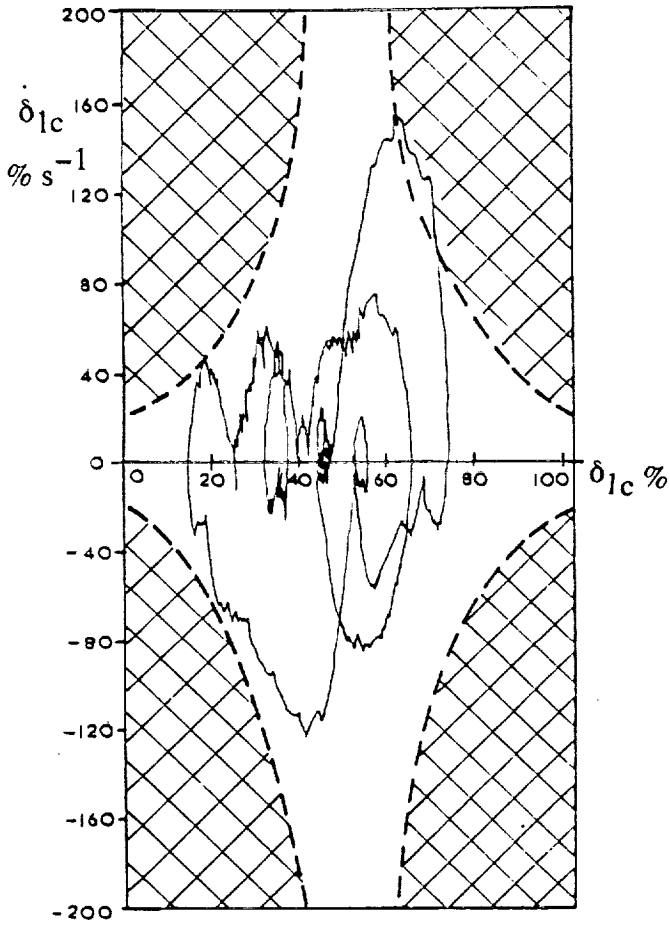

Fig 6 Actuator Phase Plane Portrait with Curtain Function

As a bonus, much valuable data on the different airframe load spectra experienced with the ACT system would be gathered and, ultimately, the load measurements would be available to the ACT system itself in the pursuit of envelope-expanding carefree handling functions.

In summary, the achievement of high performance with ACT Lynx was to be enabled through the incorporation of several layers of '**safety net**'. The hardware would be designed to exhibit fail-operate/fail safe reliability. The 'fixed' software would be designed and tested to be fault free. The control law software would operate within the constraints of the actuator curtain and be developed to a fault free state for testing in flight critical regimes. The safety pilot would be the ultimate protection against damaging flight path excursions and limit exceedances. Fatigue monitoring and accounting would protect against the consequences on airframe health of unconventional manoeuvres and control activity and provide a check for greater than usual fatigue life consumption. These safety nets were autonomous by design, yet it was recognised that only through their proper integration into the ACT Lynx concept would the performance targets be achievable. A comprehensive requirement specification was needed for the total system, developed through simulation, that defined the range of interacting functions and their operations.

Requirement Specification & Incremental Simulation

Preliminary Design Evaluation

The ACT Lynx design concept evolved from a number of preliminary studies which carefully explored the feasibility of modifying the DRA Research Lynx into a variable stability, active control, research helicopter.

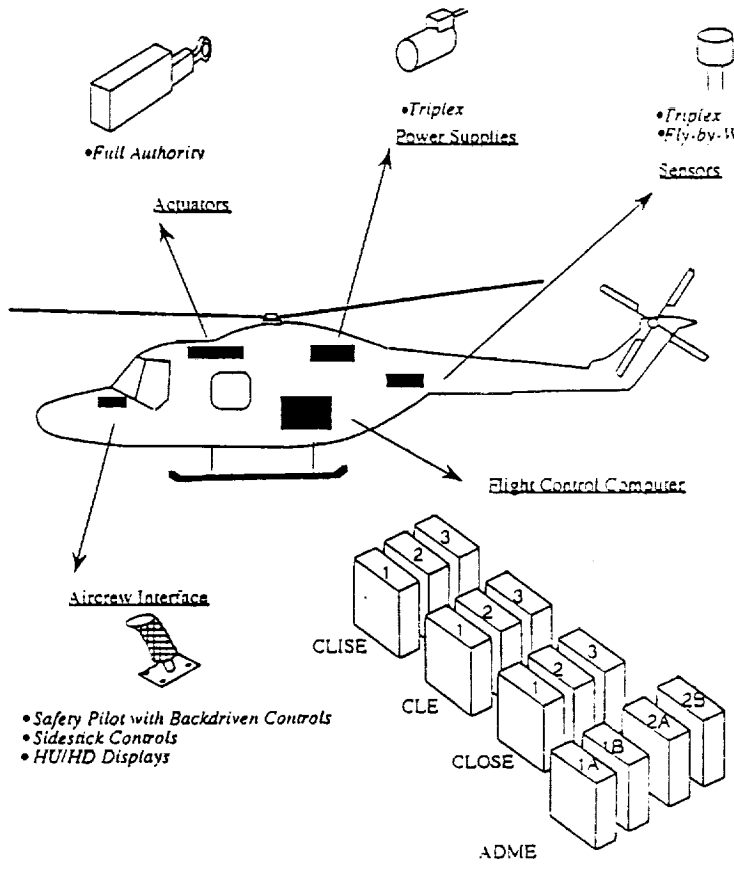

Fig 7 The ACT Lynx Concept

Practical issues addressed in these initial studies included a confirmation that the installed power and actuator system were sufficient to test to the limits of the desired ACT Lynx flight envelope, and that the mechanical linkages could be modified to allow backdriving by a set of high performance parallel actuators. Additional equipment such as sidestick controllers and advanced sensors were specified and an outline of the system architecture proposed in terms of a triplex flight control computer and a dual duplex actuator drive and monitoring unit. An entirely triplex architecture would have satisfied the fail-operate/fail-safe

requirement but, in the case of the ACT Lynx, a final component having a dual duplex arrangement was deemed to be more appropriate to connect harmoniously with the duplex hydraulic systems and primary flight control units (PFCUs). The ACT Lynx concept is illustrated in Figure 7.

A further aspect that received some preliminary design consideration was the nature of the pilot interface - that is, the displays, switches, buttons etc, that the pilot would require in order to engage and operate the facilities of the new system. These items were analysed and their likely functionality and appearance described in outline. When, in the light of these preliminary studies, the prospects for the ACT Lynx project seemed favourable, attention turned to developing a high quality specification (Refs 10, 11).

Specification Structure

In the design team there was a genuine commitment to avoid the pitfalls of many other projects and leave nothing to chance in the specification of the new system. In particular, there was a determination that the requirements specification must solve all of the significant design issues. That is, it must be correct, it must be complete, and it must be validated.

Such considerations placed a considerable challenge upon the team in the preparation of the requirements specification since there had to be sufficient detail to be totally unambiguous; that is, the implementation had to be clear, while at the same time there had to remain a high level of visibility of the design concepts and what the system was trying to do and why. These requirements are often incompatible since the very accumulation of a morass of detail imparts a complexity that militates against understanding. It is such complexity which needed to be tamed by an appropriate design and specification method, and which led to the decision to use modern software design methods for application to the whole diverse system. It was also recognised that hierarchically organised descriptions could be an effective technique for reducing complexity and in this case a decomposition of the system into its major functional elements seemed to be the most natural. This decomposition was the only one that was imposed on the system *a priori*. The outcome is shown in Figure 8, where the square and rectangular components are those relevant to the specification exercise. The bold rectangles are referred to as processing elements to be embodied in a Flight Control Computer (FCC), although such terminology was not used in the written specification. The elements of the system are described in the order of the primary flow of the signal information as illustrated by the arrows in Fig 8.

(i) Sensor Element (SE). This leading element contains the aircraft motion sensors - attitude and rate gyros and accelerometers, and also the air data units for obtaining velocity components, pressure and temperature information.

(ii) Crew Station Element (CSE). The other leading element incorporates the conventional controls for the safety pilot and a versatile sidestick controller facility for the evaluation pilot. For convenience these inceptor components were grouped together as an Inceptor Element (IE). The CSE also contains the various interfaces for the pilot to engage, operate and be cued by the ACT system as follows:

(a) Pilots Control Panel (PCP) - used by the Evaluation Pilot for engagement and disengagement and also for conducting the system-test sequence. Engage and Disengage operations would normally be performed using switches on the pilot's controls.

(b) Repeater Panel (RP) - provides a copy of the displays for the Safety Pilot.

(c) Menu Panel (MP) - provides other ACT interactions, such as selecting one of the available control laws and sets of parameter values. The same panel provides the interface for injecting preprogrammed disturbances into the system, as part of a flight-test facility used, for example, in gathering data for the validation of the helicopter simulation models and in demonstrating compliance with flying qualities requirements of new control laws.

(d) Mode Select Panel (MSP) - available for in-flight selection of control modes, for example, height-hold, speed-hold, hover hold.

(iii) Control Law Input Support Element (CLISE). This element has the main purpose of processing and managing the information from the Crew Station and Sensor Elements. It also contains the function for scheduling of a comprehensive system test.

(iv) Control Law Element (CLE). This element is supplied with inceptor, sensor, mode selection and related information by the CLISE. The CLE is the raison d'etre of the ACT Lynx since it hosts the experimental control laws which are to be evaluated. It is this element that the user of the ACT Lynx, the flying qualities engineer, will interact with. Carefully verified and validated control law software will be plugged into and unplugged from this element. Typically, six control laws will be selectable by the experimental pilot with an additional choice of up to six sets of parameters within each law.

(v) Control Law Output Support Element (CLOSE). The element following the CLE interfaces the demands produced to the remainder of the system. It also provides a selectable limiter on the demands produced by the control law as additional protection against immature software.

(vi) Actuator Drive and Monitoring Element (ADME). The final element to provide processing takes the demands from the CLOSE and produces drive signals for the parallel actuators resident in the Actuator Element, and the series actuators in the PFCU. The ADME also manages the

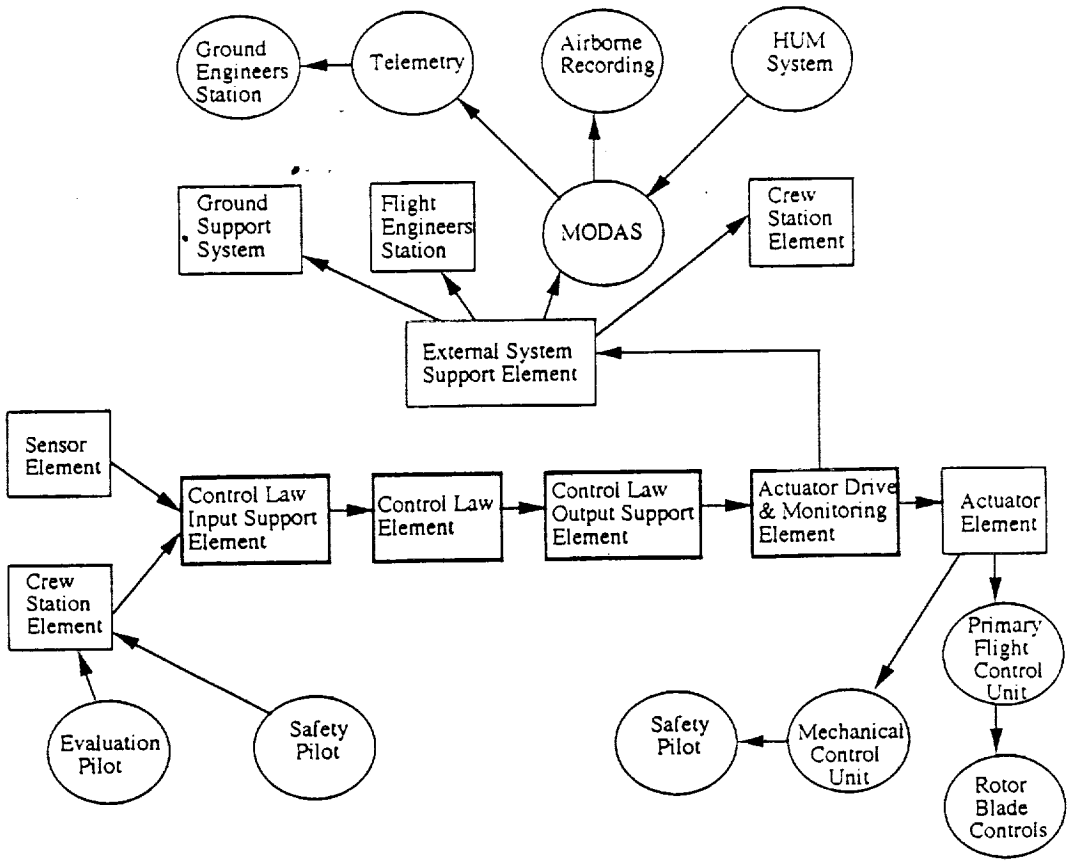

Fig 8 The ACT Lynx Logical Elements

engagement of the ACT system through the energising of the parallel actuators, and supplies a normal autostabilisation function when the ACT system is not engaged.

(vii) Actuator Element. The parallel actuator system is last in the sequence. The parallel actuators are connected to the conventional control runs from the safety pilot; when the actuators are engaged (hydraulically powered), the controls are back driven to provide the safety pilot with essential control position cues and to aid in recoveries, and forward driven to the Lynx PFCUs.

(viii) External System Support Element (ESSE). In support of the above network of elements is an element which essentially provides a catchment for all of the significant data in the system. It interfaces with the on-board data acquisition system and also with the experimental helmet mounted or head down displays. A record of all system related events such as engagement, disengagement and diagnostic messages is retained in a System Journal.

The specification takes each of the elements identified above and provides a detailed description. Each element is described in detail under the headings Type, Function, Operation, Performance, Inputs & outputs, Interfaces, Testing and Failure reporting & recovery. Where a particular element is composed of replicated units, so that several **units** together comprise an **element**, the replication of units in the element is stated and the unit itself is described under the same headings. For example, the CLISE is a triplex element composed of three identical CLISUs (Control Law Input Support Units).

In the event, this primary decomposition harmonised with the subsequently developed techniques for coping with the system's complexity. Hierarchies can lose their simplifying property if the structures become too deep; for the ACT Lynx project only three levels were employed, with quite different specification techniques and associated tools at each level:

(i) The top level is the written, structured text. It is manipulated and maintained by commercial text processing software.

(ii) The middle level is the capture of the specification in a Jackson System Development (JSD - Refs 12, 13) design, using CASE tools such as Speedbuilder (Refs 14, 15).

(iii) The lowest level is the Ada code. It is generated automatically from the JSD design using a CASE tool such as Adacode, and is acted on by a conventional compiler. The simulation so produced is an ideal vehicle for validation of the specification.

Thus each level has its own formalism and there is no decomposition from one level to another. The first consideration, as in many design problems, is deciding where to start: one advantage of the Jackson JSD approach is that the starting point is well defined; one must use the narrative text of the specification to begin the *modelling* phase.

Jackson System Development

Jackson System Development is a method of analysing a written specification for a computer system to produce a formally executable specification. The method was jointly developed by Michael Jackson and John Cameron in the early 1980s (Refs 12, 13). It consists of three stages: modelling, network and implementation. There is considerable emphasis placed on the modelling stage in order to establish, unequivocally, the information available from the world outside the system being designed, with which the system interacts.

Modelling and Entities: A model, in JSD, is a description of the real world as it appears to the system. Entities are objects in the real world which have to be modelled by the system, and of particular interest in the modelling activity are those entities which perform discrete actions. For example, a press of the ARM button by the evaluation pilot is an action to which the system must respond. The modelling phase requires that the actions be allocated to specific entities, and the main task is to identify viable entities and allocate the relevant actions to them. For each entity, the time ordering of the actions must be then be specified and, conventionally, a tree diagram is used for this purpose. As an example, consider the truncated list of actions from the ACT Lynx system shown in Figure 9. Some of these are related to the pilot entity in his role of engaging the ACT System; they may be identified and their time-ordering expressed as a tree diagram using Jackson Structured Programming (JSP) notation (Ref 13). The diagram is shown in Figure 10 where the root is named after the entity which performs the actions, and the leaves (the lowest level boxes which are named rather than numbered) hold the names of the individual actions. The intermediate nodes or boxes describe the possible types of behaviour: sequence, selection (o) and iteration (*), as denoted by the symbol in the top right hand corner of the box. The numbers in the lowest level boxes refer to changes in the state of the object (entity) as shown in the table in Figure 10.

Thus Figure 10 expresses a model of the Pilot Engagement entity as a repetition of occurrences of Engagement Cycles. An Engagement Cycle can either be a Normal Cycle, composed of a sequence of Arm, Armed, Engage, Disengage, or alternatively an Early Disengage, composed of only part of the normal sequence followed by a Disengage. The appropriate changes of state are indicated by the numbered operations for each action, and it can be seen that, prior to any action, the engagement state is initialised to DISENGAGED by operation 13.

Action	Summary	Attributes
ARM	The pilot requests that the system be armed.	
ARMED	The actuator positions and the control law demands are in harmony	
ARM_DEFAULT_MODE	The initial arming of a default control mode.	ID: MODE_ID_TYPE
CANCEL_SYSTEM_TEST	A request to cancel the system test.	
CAPTURE	This is the signal to mode to go from ARM to ARM_AND_IN_CAP	ID: MODE_ID_TYPE
COMPLETED_SYSTEM_TEST	All tests of the system test have been successfuly completed	
CONTINUE_SYSTEM_TEST	Indication that the current test of the system test has been successfully completed.	
DISENGAGE	The system has been disengaged. This may happen before engagement (1) by the pilot pressing the disengage button or (2) by the system failing to get into the ARMED or ENGAGED state. It may happen whilst ENGAGED on receipt of a signal from an actuator relaying the fact that it has become disengaged	
DOWN_DISTURBANCE_REQUEST	The pilot wishes to be offered the previous valid disturbance, that is the first disturbance with a lower index number (ID) This is equivalent to the pilot presssing the DOWN button	
ENGAGE	The pilot requests (successfully) that the system be engaged.	
FAIL_TEST_STAGE	The current 'automatic' stage of the system test has not been successfully completed.	

Fig 9 Typical List of Actions

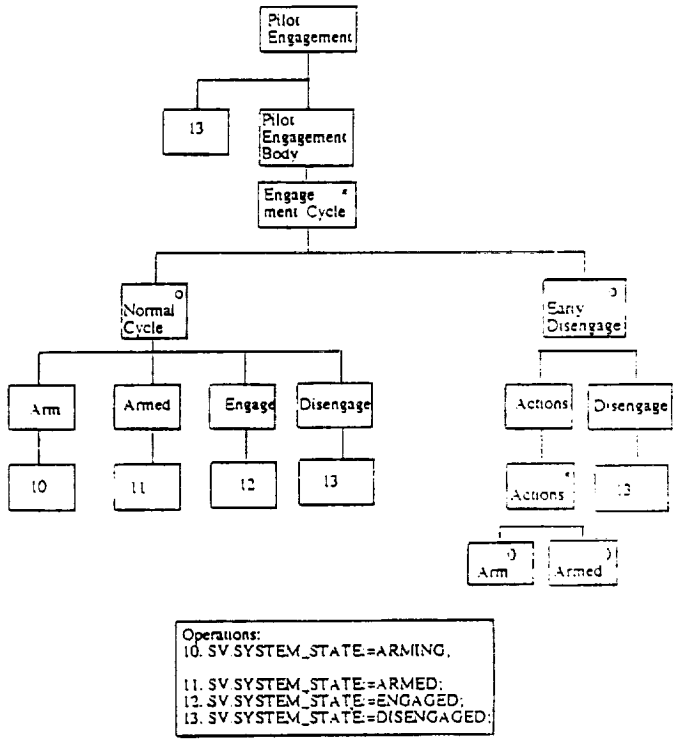

Fig 10 Pilot Engagement Cycle in JSP Form

The same type of modelling is applied to the other components of the CSE such as the activities associated with the Menu Panel and the Mode Control Panel, to obtain the full complement of model descriptions. In a completed modelling exercise, the total set of tree diagrams describes all of the time orderings of the actions plus the changes in system state. In real-time systems it is often only the activities at the man-machine interface which require this type of modelling and much of the real world is modelled simply by polling sensor information. In the ACT Lynx application, for example, the Lynx helicopter is modelled kinematically by polling data from inertial and air motion sensors. The tree diagrams in these cases are simply iterations of polling actions. The inceptor displacements are conveniently treated in this manner too.

It is fundamental to the JSD method that the model structure in Figure 10 can be used as a program structure for a process to control the engagement of the ACT system. Once operations have been added to read incoming action-messages then all that is required is for the operations to be expressed in the required language. The iterations can be expressed as loops and the selections as conditional statements with appropriate conditions. The result is that the tree diagram can be converted to code mechanically either by hand or, as in the current work, automatically.

Network and Implementation Following on from the modelling stage is the development of the *network* . Processes derived from the entities defined in the modelling stage are called model processes. Other processes are needed to make use of the data stored by the model processes in order to generate the outputs which provide the required functionality of the system. More details can be found in Reference 11.

JSD Summary The principal aim of the JSD method is to create a specification which can be usefully viewed from both above and below. The modelling stage is an object oriented analysis of the real world which produces a description which users can readily grasp, because the result is described in terms of objects familiar to the user. The tree diagrams of the method also provide important detail about the model of the real world. The network stage uses two descriptions: (a) Data flows, which can be presented to the user to indicate the architecture of the system and (b) Tree diagrams, which the analyst can use to express the design of a particular function. The resulting specification can be viewed by the user from above in terms of the interface with the real world and, simultaneously, the specification contains enough detail for the implementers below to perform their task. It is this general property that makes JSD particularly attractive and encouraged a determined assault on the difficulties associated with the application of JSD to the complete ACT Lynx System.

Specification Structure

Even with the brief review of the JSD method contained above, it should be clear that the envisaged application to the ACT Lynx presented substantial technical challenges. The primary difficulty was how to adapt the method to a system which had a diversity of types of component. For example, how was a hydraulic actuator to be specified using JSD and, in this context, what was the interpretation of data streams and state vector inspections - the JSD inter-process communication methods? A further complication was how to include the replication associated with the embedded redundancy without the occurrence of a comensurate increase in complexity. It was clear that the JSD method itself, although offering a desirable development route, did not, on its own, offer the reduction of complexity which was considered essential for the ACT Lynx requirements specification. As a compositional method, JSD eschews a top-down approach to system development. The rationale is argued at length by its proponents and a convincing case can be made for it in software development; however for more general systems, the physical architecture can impose a natural decomposition. This decomposition may then be harnessed and used to guide the development of those enhancements to JSD which are necessary to reduce the complexity of the system specification. This recourse to a decomposition based on the underlying hardware was adopted for the ACT Lynx system and led directly to a significant conceptual and practical reduction in the descriptive complexity.

JSD enhancements The next step in resolving the complexity of the system is to recognise that each identifiable element can be viewed as an independent system communicating in a limited way with other elements. For elements which are composed of replicated units, each unit is treated as independent.

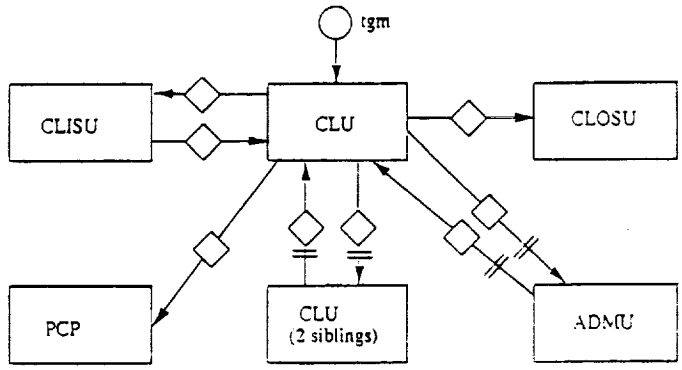

Fig 11 Top Down View of Control Law Unit

Figure 11 shows an example of such a top down view. The datastream into the CLU is a frame time-grain-marker, and the only inter-unit connections are state vector inspections. Each box represents a unit and JSD is applied in a conventional manner to that unit. The limitation of inter-unit communication to state vector inspections is crucial to the exploitation of the decomposition into elements. The absence of data-stream connections means that there are no inter-unit messages and consequently there

is no requirement to design complex process structures to handle the incoming and outgoing messages. Therefore, the complexity of a unit is determined solely by its internal functionality and, moreover, the effects of any redesign has limited impact on the rest of the system. The simplification which results from this is so significant that it justifies additional terminology, and the term *JSD unit* has been adopted.

The problem of the diversity of the system is resolved by transferring the specification to the software context. For those aspects of the system which are not expected to be digital, such as the actuator element, a simulation of that element is specified using the methods described above. Naturally, care has to taken to ensure that all of the relevant functional properties of the real element are included in the simulation specification with due authenticity. The integrity of replacing the real element in a specification by a simulation depends not only on an authentic duplication of its relevant functions, but also on ensuring that the remainder of the system only has access to that data which the real system can provide. In the case of the actuator element, for example, the actuator positions are not directly available to the ADMUs; one of the four simulated position pick-off signals for each control lane which must be used. Another example is the engagement state of the actuator; signals corresponding to appropriate sensors mounted on the actuator must be used to determine whether the actuator is hydraulically energised or not. As a consequence, the actuator entity must be modelled within the ADMU using JSD principles. The need for modelling one element within another is a natural consequence of the imposed decomposition into elements (Ref 16).

When system elements consist of replicated units, for example triplex or dual duplex, it is clearly undesirable to compose a JSD network diagram for each unit individually. At best, it duplicates effort, and at worst introduces errors caused by accidental differences in the individual networks. What is needed is to reflect the written specification and to describe a single unit in detail through a network diagram in the normal manner, supplemented by a formal description of the element in terms of its component units. Such a formal description is shown in Figure 12 (a) and (b) which depict descriptions of units of the Inceptor and Control Law Elements respectively as held on the CASE database. After some standard information (STD-INFO), consisting of its identifier and optional background detail, the MAIN-PART of the description includes a number of options such as:

(i) the type of unit - whether the unit is analogue or digital.

(ii) the number of units - here both are simplex units replicated three times.

(iii) Whether the units run synchronously or not.

To complete the description a list is required of all the JSD processes which belong to that unit, and thus need to be replicated; the final entry (UNIT-SID), being blank, shows that the name of the list on the database defaults to the name of the unit.

A similar format is provided for the description of the connections between elements as shown by the example in Figure 12 (c). The relevant fields are the source, destination and whether the connection is unit to unit individually (ONE-TO-ONE), or completely cross connected (BROADCAST). The connection description also holds some information relating to the fault tolerance implementation.

```
UNIT IE
   STD-INFO
      LONGNAME
      REFERENCE IE
      [*]CLASSIFICATION-SET
      [*]SUMMARY
      This unit is connected to the
      inceptors of the evaluation
      pilot.
      [o]NARRATIVE
         NO
MAIN-PART
   [o]TYPE
      ANALOGUE
   [o]BASE-REDUNDANCY
      SIMPLEX
   REPLICATION 3
   [o]UNIT-LVL-SYNCHRONISATION
      ASYNCHRONOUS
         FRAME-LAG
   [*]INTRA-UNIT-CONNECTIONS
UNIT-SID
```

(a) unit description (analogue)

```
UNIT CLE
  STD-INFO
     LONGNAME
     REFERENCE CLE
     [*]CLASSIFICATION-SET
     [*]SUMMARY
     This unit houses the control
     law algorithm and associated
     processing. It is the middle processor
     in a three processor "lane".
     [o]NARRATIVE
        NO
  MAIN-PART
     [o]TYPE
        DIGITAL
     [o]BASE-REDUNDANCY
        SIMPLEX
     REPLICATION 3
     [o]UNIT-LVL-SYNCHRONISATION
        ASYNCHRONOUS
           FRAME-LAG 10
     [*]INTRA-UNIT-CONNECTIONS
     UNIT-SID
```

(b) unit description (digital)

```
CONNECTION IE_CLISE
   STD-INFO
      LONGNAME
      REFERENCE IECLIS
      [*]CLASSIFICATION-SET
      [*]SUMMARY
      [o]NARRATIVE
         NO
   MAIN-PART
      SOURCE IE
      DESTINATION CLISE
      [o]DATA-TRANSMISSION
         BROADCAST
      [o]SPEC-INTERFACE
         NO
      [o]CONSOLIDATION
         YES
            HISTORY_LENGTH 3
      [o]SIBLING_ERROR_MONITORING
         YES
            HISTORY_LENGTH 3
```

(c) connection description

Fig 12 CASE Database Formal Descriptions

Incremental Implementation

The compositional, or "middle out", nature of the JSD method has the property that once a model has been built, every new function added to it provides a, potentially deliverable, working system. In fact, at any stage of the development of the network, it can be implemented. Incremental development takes advantage of this natural property of JSD and phases development of a system over a number of increments. The added functionality required from each increment is defined initially in outline and, as each increment is completed, it is reviewed and the contents of future increments re-examined in the light of any modifications or additions that have been found to be necessary. The development of a system is thus responsive to an evolving specification but at the same time allows the project to be managed on the basis of milestones actually achieved.

The ACT Lynx simulation was developed over six increments distributed as follows:

> Increment 1: A model of the pilot/system interaction including engagement of the ACT system and inceptor movement, the Repeater Panel and a display of the control run position.
>
> Increment 2: A model of the pilot/system interaction as regards System Test, Control Law Selection, Disturbance Selection, Mode Selection, Parameter Set Selection, the Menu Panel, Mode Control Panel and Pilot's Control Panel.
>
> Increment 3: A definition of a hardware description language for units and connections, and development of associated tools. The functionality of Increments 1 and 2 is based on the specified hardware, including fault tolerance. Provision for injection of errors.
>
> Increment 4: Completion of the Control Law Input Support Element, including the development of a tool for building a System Test process from a non-procedural definition. The Aircraft Motion Sensor and the Air Data Elements
>
> Increment 5: Completion of the Control Law Element and the Control Law Ouput Support Element.
>
> Increment 6: Completion of the Actuator Drive and Monitoring Element and the Actuator Element. Further development of the System Test Builder.

The simulation also includes a simple model of a Lynx helicopter to provide sensor data and the actuator displacements.

From the distribution of material in the six increments it can be seen that the primary concern was to establish an acceptable model of the pilot's interface. One of the early lessons was that different readers of a specification can place different meanings on the same words, and, for example, the sequencing of the lamps relating to engagement and system test on the Repeater Panel needed to be revised. The reference to system test in Increment 6 is indicative of the difficulties encountered in specifying a comprehensive test. The contribution from Increment 4 was not sufficient and more work had to be included in the final increment.

During the development of the simulation no fundamental flaw or omission has been discovered in the written specification. Nevertheless, a wealth of additional detail has been accumulated mainly to reinforce inadequate descriptions or to compensate for minor omissions. The most significant inadequacy was the omission of a description about how to apply the consolidation algorithm of Reference 17 to replicated units in a fault tolerant manner (additional voting was included).

Implementation of the Simulation

Ada was selected as the implementation language; its selection was determined by a number of considerations. First, Ada is a US DoD mandated language, and was also "highly recommended" by the UK MOD, which has resulted in a number of very high quality compilers being available. In addition, packages and tasks are language features which have been very important in implementing the system. Finally, the code generation tool Adacode, described below, was already available in prototype form to serve as a basis for the project.

Complexity Overview

The question of complexity has been addressed from several viewpoints. The JSD method incorporates in its modelling phase a powerful technique for grasping the fundamentals of system development and provides a solid platform for subsequent work. On its own, however, it is not sufficient for resolving the complexity of a diverse system which has inbuilt redundancies. It is necessary to introduce additional features, JSD units and connections, to reduce the complexity of whole system to a manageable size. These conceptual advances are of little practical use without associated support from CASE tools. A database must be able to accept and manipulate the unit definitions, and code generation tools must be able to access this information in order to build the final system. The whole JSD-unit based approach gives rise to a management of the complexity of the system to the extent that it may be considered tamed. The verification which is embedded in the various stages of the development of the specification, and its resultant validation through operation of the simulation ensure that any *rogue* aspects of the specification have been eliminated.

At the heart of this complex specification, the detailed requirement for the control law element was left blank; for initial clearance this would be a unity transfer function followed by a digital representation of the Lynx analogue

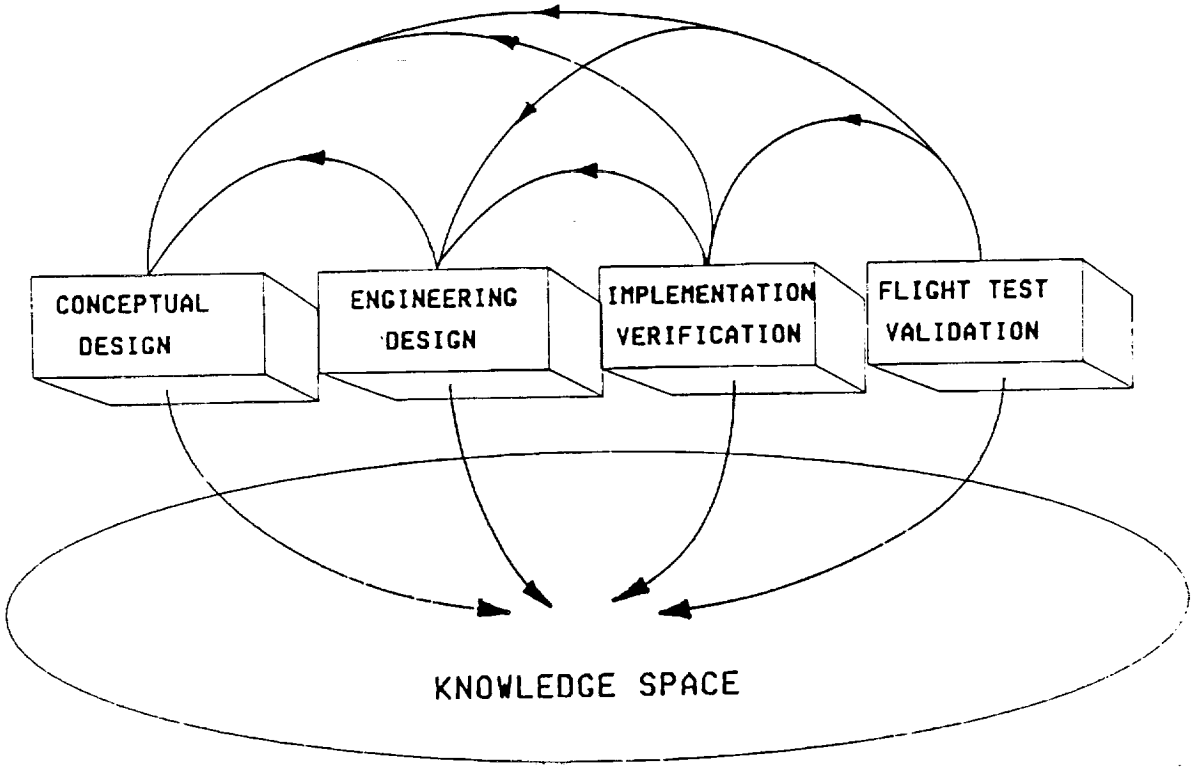

Fig 13 The Four Phases in a Control Law Life Cycle

AFCS. But the experimental control laws were the *raison d'etre* for the ACT Lynx project, and needed a different approach to their development.

Control Law/Concept Evaluation - Envelope Expansion

Recognising that immaturity would be a normal part of the development of ACT Lynx laws, a Control Law Life Cycle Model and associated working practices and procedures were developed at DRA to ensure a disciplined path to full control law validation (Ref 18, 19). The development cycle was formalised to ensure that when control laws were ultimately exercised in safety critical areas, there would be no possibility of them failing. Thus, along with the hardware redundancy, the system would have a truly comprehensive fail-operate capability. The cycle comprises four phases (Fig 13);

i) The Conceptual Phase (CP) evaluates basic concepts in a form that can capture the operational requirements. It includes simple modelling, design and analysis activities and pilot-in-the-loop simulation. Outputs from this phase include knowledge of the response types and system characteristics required to achieve the various Levels of flying quality.

ii) The Engineering Design Phase (EDP) takes results from the CP and involves full control law design with a representative vehicle model and includes refinements to control system architectures via detailed modelling and extensive piloted simulation.

iii) The Flight Clearance Phase (FCP) consolidates results from earlier stages and achieves a verified implementation for the target flight control computer. Validation of the design, including a loads and stability analysis, is a key activity in the Clearance phase. The techniques of 'Inverse Simulation' (Ref 20), with prescribed MTEs, offer a convenient and efficient method for exercising the control law in a wide range range of representative conditions prior to flight.

iv) The Flight Test Phase (FTP) evaluates the control system in full scale flight and appropriate operational MTEs. Experiments in this phase will be 'replicas' of tests conducted in ground-based simulation and changes to control laws would cover only those regimes mapped out in the Conceptual and Engineering Design phases. An incremental approach to safety critical, high risk, flight conditions would be normal practice.

The phases are sequential but also iterative, acknowledging that growth in knowledge can lead to a change in the requirement or criteria format, often the objective of the research itself. At all stages, the discovery of a fault, design error or uncertainty will generally require the return to a previous phase. Special care needs to be taken when 'imposing' a procedural discipline on research, that creativity is not inhibited, but the discipline needs to cut even deeper with well defined working practices and activities, if it is to have any real meaning as a safeguard against errors or faults being designed in. Fig 14, taken from Ref 18, illustrates a process structure diagram for the CP with the three principal tasks - problem expression,

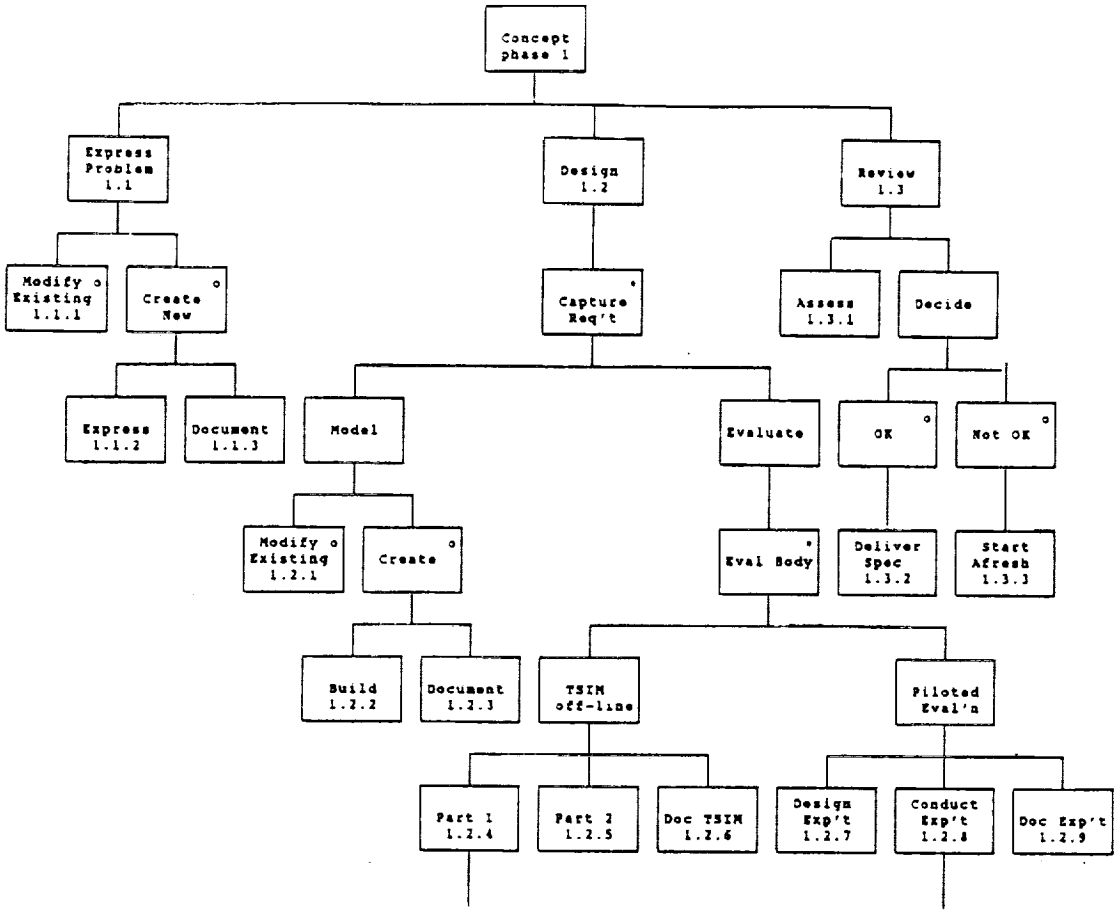

Fig 14 Structure Diagram for the Conceptual Phase

design and review. The JSD notation is again used, ie sequence, iteration (*) and selection (o), with the *activities* corresponding to the lowest level 'leaves' on each branch. Typically, documentation is required as each new piece of knowledge is accumulated and this is reflected in the right hand leaf of the branches.

Conceptual Phase

Examples of research in the Conceptual Phase can be found in References 21, 22 and 23. The archetypal DRA conceptual simulation model (CSM) was developed in Reference 21, which reported comparative results with different response types and autopilot modes. In Reference 22, the first conceptual results from the DRA/Westland research into carefree handling systems were published, indicating the significant benefits of direct intervention control laws. More recently, the first helicopter trials on the DRA Large Motion Simulator reported the achievement of Level 1 handling qualities for rate response types (Ref 23). Fig 15 shows one set of results from Reference 23, with pilot handling qualities ratings plotted against roll attitude bandwidth for a slalom task. The wide spread of ratings with each configuration illustrate the change in perceived handling as performance is increased, the poorest ratings generally corresponding to the highest levels of pilot aggressiveness.

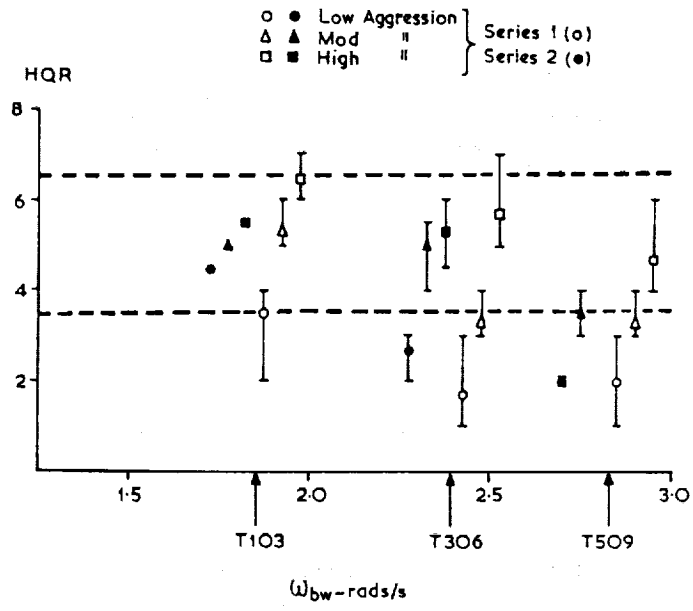

Fig 15 HQRs for Slalom MTE Flown With DRA Conceptual Simulation Model

The ADS33C Level1/2 bandwidth boundary for non-tracking tasks is 2 rad/s, corresponding with the lowest level of aggressiveness flown in the AFS trials. The degradation at higher performance levels is consistent with flight results (Ref 2), but pilots tend to be more senstitive to task cues and critical of simulator deficiencies as aggressiveness increases. Flying at large attitude angles near the ground is particularly demanding on the fidelity of the simulated visual cues; the limited vertical field of view and texture on the current AFS visual system must be a major factor in the inability to achieve Level 1 at high performance. This deficiency, along with modelling uncertainties, common to all ground-based simulators, is, of course, a primary reason for the vigorous pursuit of high performance in-flight simulators.

Engineering Design Phase

This phase consists of mapping the required characteristics from the CP onto the simulated target aircraft. As in the CP, problem expression, design and review cover activities in the Engineering Design phase. However, the level of detail will be considerably greater, including environmental constraints and robustness criteria. Internal control system loop performance requirements and stability of uncontrolled airframe modes will form parts of the problem expression. The design sub-phase contains the modelling and evaluation activities, as in the CP, but also includes significant new activities under the synthesis label (Ref 18). The desired flying qualities requirements, embodied in handling and ride quality functions, will be cast in functional form and the associated 'error' cost functions minimised with respect to control system gains and filter frequencies. This is the essence of the synthesis at the centre of control law design and a number of different techniques are available for working the optimisation, involving craft-like skills and trading performance and robustness to achieve the best controller. Examples of results from the Engineering Design phase are reported in Refs 24, 25 and 26.

Clearance and Flight Test Phases

Activities within this phase have not been well developed at DRA for the helicopter application. The clearance activities will include software verification and a degree of validation using more comprehensive models than in earlier 'real-time' evaluations, with the control law now embedded in the target hardware. Flight tests represent the ultimate research evaluation, although ironically, here there is little scope for design innovation and creativity. Flight test is essentially a knowledge gathering exercise, but there is considerable scope for innovation in experimental design. A procedure sequence in the evaluation of a control law might take the form;

i) engage ACT system when in required flight condition,

ii) build up task complexity and aggressiveness incrementally

iii) curtain function cleared for minimum flight envelope initially (low aggressiveness)

iv) open curtain incrementally as aggressiveness increased

v) test control law at safe altitude initially with representative task gain (eg using helmet display)

vi) test control law at low altitude with representative natural task cues

Throughout this process, regular reviews of the documented results with results from previous phases will be required. A fully developed control law, enabling Level 1 flying qualities at high agility levels, should never experience a software 'failure'. Hardware failures will be protected against to a high reliability through redundancy. Inadvertent excursions beyond flight envelope limits will be protected against with built-in carefree handling functions, working as an integral part of the control law.

Conclusions and Recommendations

With the aim of developing a high performance ACT research helicopter, the DRA has developed the ACT Lynx concept; focus has been on research at high agility levels to explore carefree handling concepts and the expansion of the helicopter's usable flight envelope. The inherent high agility of the Lynx, with its hingeless rotor, makes it an excellent airframe for establishing requirements for future types. This paper has reviewed this project from the standpoint of the conflict between safety and performance; we can see a way through but a number of concurrent safety nets need to be combined.

1) A highly skilled and motivated safety pilot with backdriven conventional controls is the most important safety net; exploratory simulation studies conducted at DRA have focussed on recoveries to common mode hardover failures. The results have highlighted recovery times generally consistent with past flight experience although torque, rotorspeed and 'g' limits can easily be exceeded.

2) System redundancy providing a fail-operate/fail-safe capability provides the strongest and most effective safety net against hardware failures.

3) A comprehensive requirement specification developed through simulation ensures that the integrated system is well understood and all functions and their operations are fully defined; this approach ensures that the 'fixed' software is coherent and fully validated, hence providing the most effective protection against common mode software failures.

4) Control laws developed within the framework of an iterative life-cycle, including ground based simulation, ensures protection against software errors during the early development stages of this critical element of the system. The four phases - conceptual, engineering, clearance and flight - have been briefly described.

5) Curtain functions, limiting the actuator drive signals, can also be used to protect against immaturity in the control laws and can be opened incrementally to allow more agility to be exploited.

6) A committment to carefree handling functions embedded within the control laws is considered to be an essential ingredient to ACT research if full agility is to be realised. Ultimately, together with the safety pilot and FOFS hardware, this should complete the triad of safety nets necessary for the synergy of performance and safety.

At the time of writing, the UK programme is at a hiatus due to funding limitations. In this paper the authors have attempted to provide a candid exposure of some of the issues sorrounding the safety/performance conflict, to stimulate a continuing debate with collaborative partners pursuing similar goals. It is believed that flight research at high agility levels will only be possible, with acceptable risk, if these issues are squarely faced.

Acknowledgement

The research reported in this paper was conducted as part of the UK MoD's Applied Research Programme - Package 3D (Tri-Service Helicopters).

References

1 Padfield, G.D., Lappos, N., Hodgkinson, J.; "The Impact of Flying Qualities on Helicopter Operational Agility"; AHS/NASA Conference on Flying Qualities and Human Factors of Vertical Flight Aircraft; San Fransisco, Jan 1993

2 Charlton, M. T., Padfield, G. D., Horton, R. I.; "Helicopter Agility in Low Speed Manoeuvres"; Proceedings of the 13th European Rotorcraft Forum, Arles, France, Sept 1987 (also RAE TM FM 22, April 1989)

3 Morgan, M.; "Airborne Simulation at the National Aeronautical Establishment of Canada", AGARD CP 408 'Flight Simulation', 1985

4 Hartman, L.J. et al; "Testing of the Advanced Digital Optical Control System" 43rd AHS Forum, St Louis, May 1987

5 Damotte, S. et al; "Evaluation of Advanced Control Laws with a Sidestick Controller on the Experimental Fly-by-Wire Dauphin Helicopter"; 18th European Rotorcraft Forum, Avignon, France, Sept 1992

6 Bouwer, G. et al; "ATTHeS - A Helicopter In-Flight Simulator with High Bandwidth Capability", 48th AHS Forum, Washington, June 1992

7 Hindson, William S.; "Past Applications and Future Potential of Variable Stability Research Helicopters", Helicopter Handling Qualities, NASA CP 2219, April 1982

8 Gupta, B.P. et al; "Design, Development and Flight Evaluation of an Advanced Digital Flight Control System", 43rd AHS Forum, St Louis, May 1987

9 Kimberley, A.M., Charlton, M.T.; "ACT Lynx Safety Pilot Simulation - Trial Runaway"; RAE Working Paper WP(89) 031, June 1989

10 Padfield, G.D., Bradley, R., Moore, A.; "The Development of a Requirement Specification for an Experimental Active Flight Control System for a Variable Stability Helicopter - an Ada Simulation in JSD"; AGARD CP 503, 'Software for Guidance and Control', Sept 1991

11 Padfield, G.D., Bradley, R.; "Creation of a Living Specification for an Experimental Helicopter Active Control System Through Incremental Simulation"; Proceedings of the 17th European Rotorcraft Forum, Berlin, Sept 1991

12 Jackson, M.; System Development. Prentice Hall, 1983.

13 Cameron, J. R.; JSP & JSD: The Jackson Approach to System Development. IEEE Computer Society Press, 1983.

14 Michael Jackson Systems Ltd., Version 3 of Speedbuilder for IBM PC/Compatibles: Installation Guide, MJSL, 1989.

15 Lawton J R & France N. "The Transformations of JSD Specifications in Ada". Ada User, Jan 1988.

16 Bradley, R., "A Method for Specifying Complex Systems with Application to an Experimental Variable Stability Helicopter", Ph.D. Thesis, Glasgow University, 1992.

17 Silva, A.; "Mode Synchronisation Algorithm for Asynchronous Autopilot", Fourteenth European Rotorcraft Forum, Milan, 1988.

18 Tomlinson, B.N., Padfield, G.D., Smith, P.R.; "Computer Aided Control Law Research - from Concept to Flight Test"; AGARD CP 473 'Computer Aided System Design and Simulation', August 1990

19 Padfield, G.D., Tomlinson, B.N., Smith, P.R.; "Management of Computer Aided Control System Design from Concept to Flight Test"; 'Safecomp'90', Safety of Computer Control Systems, IFAC Symposia Series, 1990, No 17

20 Bradley, R., Thomson, D.; "The Development and Potential of Inverse Simulation for the Quantitative Assessment of Helicopter Handling Qualities", AHS/NASA Conference on Flying Qualities and Human Factors of Vertical Flight Aircraft, San Fransisco, Jan 1993

21 Buckingham, S. L., Padfield, G. D., "Piloted Simulations to Explore Helicopter Advanced Control Systems"; RAE Tech Report 86022, April 1986

22 Massey, C., Wells, P.M.; "Helicopter Carefree Handling Systems"; Proceedings of RAeSoc Conference on Helicopter Handling Qualities and Control, London, Nov 1988

23 Padfield, G.D. et al, "Helicopter Flying Qualities in Critical Mission Task Elements"; 18th European Rotorcraft Forum, Avignon, France, Sept 1992

24 Yue, A., Postlethwaite, I., Padfield, G.D.; "H^{∞} Design and the Improvement of Helicopter Handling Qualities"; Proceedings of the 13th European Rotorcraft Forum, Arles, France, Sept 1987, Also Vertica, Vol 13 No 2, 1989

25 Manness, M.A., Murray-Smith, D.J.; "Aspects of Multi-Variable Flight Control Law Design for Helicopters using Eigenstructure Assignment", J. AHS, Vol 37, No 3, July 1992

26 Walker, D. et al; "Rotorcraft Flying Qualities Improvement Using Advanced Control"; AHS/NASA Conference on Flying Qualities and Human Factors of Vertical Flight Aircraft, San Fransisco, Jan 1993

Session 3

Modeling and Analysis Techniques