

# VEHICLE MANAGEMENT SYSTEMS IMPLEMENTED THROUGH AN IMA ARCHITECTURE

R M Stewart / C P Ephraim  
Stewart Hughes Limited  
Chandlers Ford, Eastleigh, Hampshire, UK

## Abstract

Advances in avionic technology now permit the construction of integrated systems that bring together in one box processes of widely different certification level, from flight control at one extreme to simple data logging at the other. The first commercial system of this type appeared in the Boeing 777 as the Aircraft Integrated Maintenance System or AIMS. However, it is likely that such a system would be too expensive for helicopter use and so efforts are being made to find and develop radically different architectures primarily with this type of aircraft in mind. The focus for the moment is on what might be referred to as a Vehicle Management System or VMS. This would bring together a number of functions that share a great deal of expensive to gather and store data, and so bring together within one unit a whole variety of functions hitherto distributed among many individual boxes - carefree handling, usage monitoring, health monitoring, flight data recording, ground proximity warning, aircraft and obstacle warning, cockpit displays, power management, avionic diagnostics and digital aircraft-to-ground communications.

## TECHNICAL GOALS AND ECONOMIC OBSTACLES

By anyone's reckoning the helicopter is a difficult machine to fly, especially at the edge of its envelope and manoeuvring near buildings, ships or other obstacles in adverse conditions. Secondly, it is a machine prone to catastrophic mechanical failure. Thirdly, it is an expensive machine to operate for reasons of high insurance and maintenance cost.

The high level technical goals must therefore include reduction in the aircraft's operating cost and improvement in safety, both of which can be achieved in a number of ways, from better design through tightening up of pilotage and maintenance procedures to onboard avionic systems geared to the production and presentation of better information for aircrew and maintainers. This paper obviously concentrates on the latter.

## Safety

Table 1 shows some data culled from a recent *Flight* magazine article on small helicopter accidents. Like all statistics the data is open to different interpretations, but clearly it shows that the two biggest causes of accidents to be wire strikes and loss of pilot awareness in a range of situations, both inside and outside of the cockpit. Engine failure is high on the list because many small helicopters are single engined. In many instances engine failure leads to a badly conducted autorotation. Tail rotor failure is significant, as it is on all helicopters. Finally, there are rotor tip strike and load management, e.g. flying with underslung loads, loads that are too heavy, loads that move the CG either too far forward or too far aft, loads in Kg thought by the pilot to be measured in lbs, cargo boxes full of material thought by the pilot to be empty.

The situation for large helicopters is not too different. Flight into power lines and engine failure may be less important and mechanical failure, e.g. of tail rotors, more important.

Table 1 : Causes of Small Helicopter Accidents

Lloyds, All Accident Types, 1994	
Flight into power lines	28
Loss of control during an emergency (eg low NR) or loss of situational awareness and CFIT	23
Engine failure	22
Rotor strike	13
Tail rotor fail	9
Load management	8
Others (8)	28

Source: *Flight*, Sept 1995

## Operating Cost

Table 2 shows some data culled from *Vertiflite* on US helicopter direct operating costs. This shows the very high percentage costs of insurance and

maintenance (European insurance costs are thought to be significantly below those of the US), in turn due to the helicopter's poor accident record and weight of onboard machinery. The balance in accidents between human, mechanical and environmental factors is shown in Table 3, heavily biased towards the human aspects of helicopter operation.

Table 2 : Helicopter Operating Costs (US Commercial)

Depreciation	36%	} (reducing with age) HUM-FDR-PA targets
Insurance*	29%	
Maintenance	24%	
Flight crew	6%	
Fuel	5%	
	100%	

Source: Vertiflite, Jan/Feb 1993

\*European rates are thought to be in the range 10-15%

Table 3 : Causes of Helicopter Accidents

Systems failure	33%	(transmission, engines rotors, flight controls)
Human error	63%	(e.g. turning tail into wind, powerline strike)
Environmental	4%	(e.g. lightning strikes)
	100%	

Source: US NTSB

### Technical Goals

At the core of the vehicle management systems being considered by the author's company lies the HUM, or the Health and Usage Monitoring System, on large machines typically integrated with the Flight Data Recording (FDR) system. These have been in fleet service now for some six years and are beginning to show the benefits promised of them. For what it is worth, recent pronouncements by the CAA claim they are now producing data that has a bearing on over 60% of all airworthiness incidents. Recent data published in *Flight* shows a fairly dramatic decline in European accident rates following initiatives precipitated by the BV-234 accident off Sumburgh. One of these initiatives was the mandatory fit of FDR and the oil company led decision to fit HUM at the same time. How much either of these contributed to the decline in accident rates is by no means clear because many other things were done at the same time, e.g. the

tightening up of operational procedures, more training etc.

Perhaps of equal importance is the fact that the most recently developed systems appear to be going to enter commercial service with a manufacturer-promised return of a 25% reduction in aircraft DOC, achieved through an almost equal mixture of health and usage monitoring. If this turns out to be true, HUM will have made a truly large impact on helicopter operation, far more than its originators, including the author, had thought possible. On the military front one manufacturer has recently announced the possibility of a 35% reduction in O&S cost with a HUM fitted to their specification.

The primary technical goal must therefore be to do for the human element of aircraft operation what HUM is apparently capable of doing for the mechanical part. Much can now be done, with the exception of the wire strike problem, which is technically feasible but still too expensive. The problem, as it was with HUM, lies in justifying the economic case.

### Economic Obstacles

The economic obstacles are two-fold. The first is at the operational level, and, as alluded to above, related to the business of cost justification where the benefit is what might be termed 'probabilistic', i.e. not tangible in terms of dollars *definitely* saved per flight hour. The second is at the system manufacturing level where the costs of design, flight trial and certification can be very high. Part of this paper addresses the latter, largely through discussing ways in which normally quite expensive functions such as carefree handling, helicopter-compatible ground proximity warning and rotor tip strike may be added at minimal incremental cost to a modern HUM/FDR.

### VEHICLE MANAGEMENT TECHNOLOGIES

We will define the term *Vehicle Management Technology* to mean anything onboard the aircraft, apart from primary flight, nav and comms instruments, that either *assists* the pilot fly the aircraft more efficiently and/or safely, or gathers information for use on the ground to *assist* management of the aircraft's health or usage. There could of course be an overlap with flight management technology, but that is a term generally restricted to flight planning. Finally, there is a definite overlap with systems like TCAS, and in fact there are elements within VMS technology that attempt to offer a lower cost alternative to TCAS.

The purpose of this section is therefore to review some of the technologies which, under the above definition, fall into this area.

### Carefree Handling

Carefree handling is the broad term used to describe any technology that allows the pilot to spend less time worrying about what torque or engine temperature he is 'pulling', what rotor stresses he is inducing, etc. In extreme terms, this means a system that can accept broad pilot commands to turn, dive, climb etc and subsequently turn these into efficient control actions. Fly-by-wire of course does much of this, but is very expensive.

Use of the term here is much more restricted. Essentially it means putting artificial feel into the controls to cue the pilot on the imminent possibility of an over-torque, rotor under-speed/over-speed, rotor stall, rotor stress, engine TET exceedance etc condition. Based on this cueing information the pilot may then decide either to carry on as before (e.g. because he is in a life threatening flight state) or back off on the appropriate control.

At the core of this technology are the predictive algorithms used to drive the cueing system. These can range from being very simple to highly sophisticated. It is essential that the cue communicates the appropriate 'back-off' direction.

Perhaps the simplest carefree handling system involves the collective stick and use of a feel system to assist the pilot avoid unnecessary over-torques and rotor underspeeds or overspeeds. Put

very simply, should the system predict that near term (typically within 1 sec) torque is heading for a value to be avoided it increases by a small amount the force required to raise the collective stick. Full motion simulator trials (Reference 1) have shown such a cue to be very effective, and more effective than a variety of both visual and aural cues.

There are several pieces of technology required to realise such a system, one of them being the algorithm for predicting torque. Shown in Figure 1 is a polynomial function network created by automated learning techniques to predict torque on the Sikorsky Blackhawk aircraft; the polynomial function in Figure 2 shows the polynomial function embedded in Node 7, and Figure 3 shows the average algorithm accuracy during the flight test program. There is little point in discussing here the form of the polynomial. It is generated by a statistical optimisation process that, for a given amount of training data, trades off network accuracy against network complexity, and is almost entirely dependent for its final form on the training data fed to it. It cannot be over-trained. Once however the function has been learnt, it is frozen. The learning procedure places limits both on the overall complexity of the network and the complexity of the function inside each node.

Very much more sophisticated systems are possible. Current research is looking at feel systems for the pedals and cyclic stick, the latter to assist the pilot fly with minimum fatigue damage. Other variants are targeted at more general control of the aircraft in the heave axis.

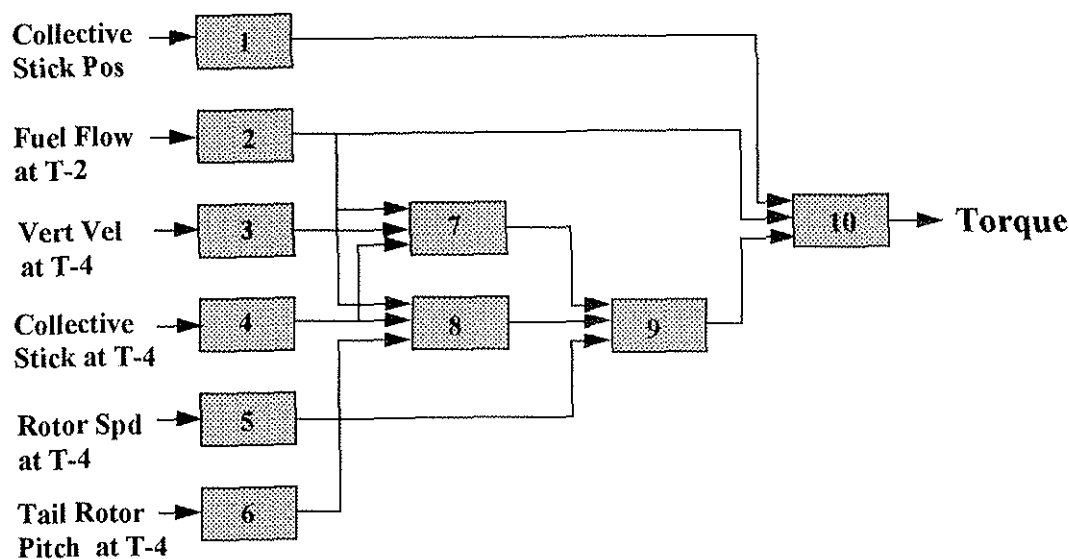


Figure 1 : Torque Prediction PFN (Blackhawk, 0.115 secs)

- $C_0$  = collective stick position at time  $t = 0$
- $F_{-2}$  = fuel flow at  $t-2$
- $V_{-4}$  = vertical velocity at  $t-4$
- $C_{-4}$  = collective stick at  $t-4$
- $N_{-4}$  = rotor speed at  $t-4$
- $T_{-4}$  = tail rotor pitch at  $t-4$

Node 7  
 output =  $24.6-434.1 * F_{-2} + 6338.4 * F_{-2}^2 - 10438.7 * F_{-2}^3$   
 $- 0.177 * V_{-4} + 5.11 * F_{-2} * V_{-4} - 25.46 * F_{-2}^2 * V_{-4}$   
 $- 0.008 * V_{-4}^3 - 0.75 * F_{-2} * C_{-4} - 28.74 * F_{-2}^2 * C_{-4}$   
 $- 0.013 * V_{-4} * C_{-4} + 0.036 * F_{-2} * V_{-4} * C_{-4}$   
 $+ 0.0001 * V_{-4}^2 * C_{-4} - 0.0028 * C_{-4}^2$   
 $- 0.09 * F_{-2} * C_{-4}^2 + 9.14e^{-5} * V_{-4} * C_{-4}^2$   
 $- 6.07e^{-5} * C_{-4}^3$

Figure 2

Carefree handling is but a small step on the way to full fly-by-wire. However, it is orders of magnitude less expensive, and so attractive to designers of small helicopter as well as expensive military ones. Perhaps more interestingly, the new technologies being used in the development of carefree handling show promise of significantly reducing the costs of fly-by-wire development.

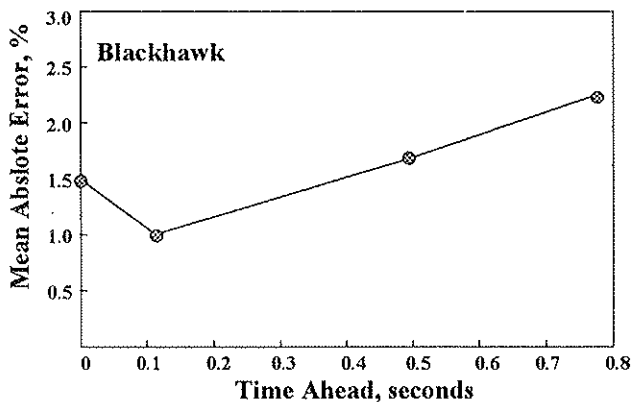


Figure 3 : Torque Prediction Accuracy using PFNs

Health Monitoring

This is primarily a ground-based function, but there are pilot assist requirements of some considerable importance, notably in the detection of tail rotor failure and the mode of that failure, i.e. whether it is within the drive line, control system or rotor itself.

Much of the ground-based North Sea health monitoring technology is too sophisticated in its

interpretative requirements for cockpit use. Quite serious faults might take several hours of a skilled interpreter's time to pin-point the possible cause and decide upon the appropriate remedial action. What is therefore required in the air is something much simpler that responds only to really threatening conditions, is able to pin-point the general area of the problem and is 'red/green light' in nature. There has been some talk about neural networks providing such a system, but these need training data in proportion to their size and that is generally just not available. Faults of the kind one would want to warn the pilot of seldom occur more than once. The best that neural nets could therefore offer is some kind of anomaly warning, but there again what would one tell the pilot apart from the fact that something or other has changed. In the author's view this is more a problem of physics than voodoo mathematics. What is needed are simple sensor systems that can detect the signs of catastrophic failure without being confused by normal machine activity. These fortunately are being worked on by various groups of researchers and results should soon be available. One such system attempts to sense, in a non-invasive way, the electrostatic activity generated by metal being ripped off the surface of a gear tooth. Others are working on acoustic emission devices for attachment to main rotor blade spindles.

The ground-based side of HUM has advanced by leaps and bounds over the past three years. Once the airborne systems became reliable (and the data they produced believable) productive effort could be funnelled into reducing false alarm rates and improving detection efficiencies. Today we are probably detecting something like three out of every five faults observable through vibration analysis, and false warning rates are running at about one per hundred flight hours. False gearbox pulls are much lower than that - probably about one per hundred thousand hours (a rare event) for a well supported system. Here the operative phrase is 'well supported' in the sense that the engineer on the flight line must be backed up by people with a thorough understanding of what the system is capable of doing.

The most important single problem to be tackled on the ground station is lowering even more the amount of effort expended by the skilled interpreter in investigating and cancelling false warnings. How this is to be done depends to a large extent on the purpose of the system. Within the North Sea environment that has so far been to lower the incidence of mechanically-induced accidents, and since these are generally rare the database on which to train any human or algorithmic system is very small. North Sea

systems are therefore generally used 'proactively' in the sense that the user is spending much of his or her time looking for evidence of new, never-before-seen faults likely to cause an accident. This is perhaps not what some military operators would like to use the system for as their complaint about North Sea HUM is that it requires too much in the way of operator skill. Their use of the system can therefore only be 'retroactive' in the sense that they are only interested in a system that can learn from experience, and so detect faults only after sufficient experience has been gathered to train a neural network, or some other such tool. Neither approach is to be criticised - it is just a matter of the realities of life. The problem arises when one tries to do the former via the latter.

### Usage Monitoring

Together with health monitoring this forms the bulk of what is in a current state-of-the-art HUM system. It can be almost trivial or very sophisticated in its implementation, depending almost entirely on the degree of involvement of the aircraft manufacturer. Within the R&D community there are five broad categories of usage monitoring technology being discussed, namely (1) simple measurement of time in flight, hover etc, (2) flight regime recognition, (3) manoeuvre recognition, (4) loads synthesis, and (5) direct strain measurement. Only one of those, the first, is currently in service, but at least one other (3) is about to enter service.

The central helicopter usage monitoring problem is typically the main rotor blade spindle and/or strap pack. Ideally, one would like to measure the strain on the component directly, in real time, and from that calculate life consumption via range pair counting, Miner's Law etc.

All the above mentioned methods have deficiencies of one sort or another. Working backwards, direct strain measurement is difficult to envisage working on the head without a significant advance in either slip ring or wireless data transmission technology. Loads synthesis, which involves the high accuracy calculation of rotor head strain from dynamic measurements made in the fuselage, has the problem of creating the estimation algorithm and ensuring its accuracy over the whole flight envelope. Manoeuvre recognition has the problem of assigning the aircraft's current flight state to some form of manoeuvre and then matching that to one or some combination of test manoeuvres (typically 20 to 30) originally flown to fatigue qualify the aircraft. Flight regime recognition has the problem of matching current flight state (typically computed in 2 sec timeframes) with component fatigue usage. Both (2) and (3) have been extensively flight tested.

The problem is that as one moves further and further away from the ideal (i.e. direct measurement) the usefulness of the technique diminishes in respect of the benefits that can be claimed. Everything must therefore be a compromise based on (a) system cost, (b) system maintainability, and (c) benefit in terms of life extensions gained. Where one ends up for any particular aircraft depends on factors such as the anticipated operational spectrum versus the design spectrum (if the two are close why bother with usage monitoring). Generally speaking, the largest gains are to be made from intensively flown commercial passenger transports designed originally to fly an arduous military mission.

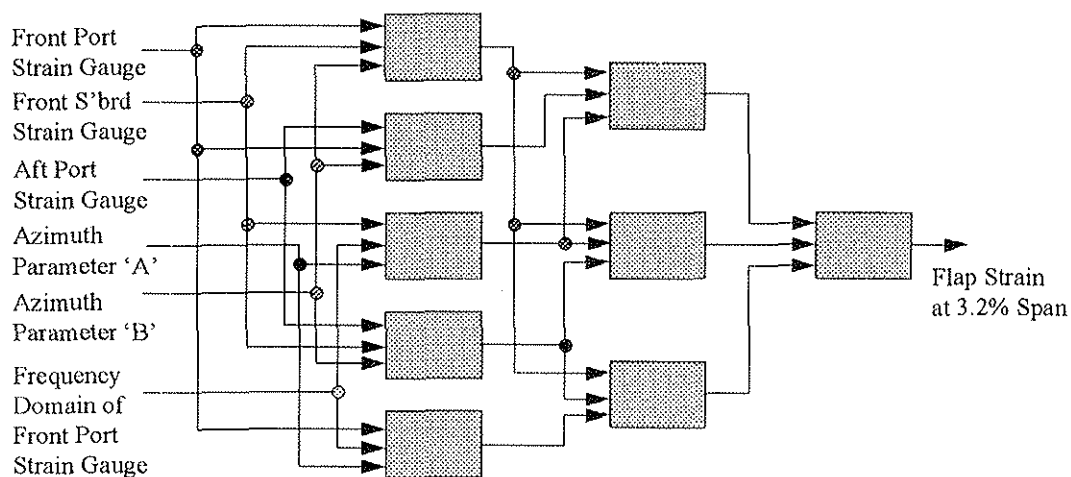


Figure 4 : Aircraft Usage Monitoring Using Polynomial Functions

A loads synthesis technology of considerable promise is that which employs polynomial function networks to estimate rotor strain from fuselage data. Shown in Figure 4 is one such network created for the Lynx aircraft and its dogbone flap strain. Figure 5 shows a comparison of estimated versus actual measured strain. The two have to be very similar in their major peak to peak values for the technology to be practical. The technical difficulties lie in (a) getting an adequate amount of blade harmonic data into the estimate, (b) achieving adequate coverage over the usable flight envelope, and (c) building in sufficient system self test.

### Tip Strike Warning

Rotor tip damage, and even catastrophic rotor tip damage, are important issues for helicopter pilots and maintainers alike. In the space of three weeks during May 1996 six aircraft were lost world-wide due to one aircraft colliding with another, four in broad daylight. Military aircraft in particular suffer due to the fact that they often have to land in wooded areas or fly at low speed through woods below treetop level. Helicopters landing or taking off from ships likewise have a hard time.

Both main and tail rotors are susceptible to tip strikes. However, the easier one to handle technically is probably the main rotor. Simple laser diode based sensors can be placed on the side of the aircraft's fuselage, or wheel sponsons, to look out up to 60 meters from the aircraft in a full 360 deg arc. These sensors are capable of detecting reflections down to 6% (a tree leaf will have a reflectivity of typically 15%) and give the pilot

something like a 6 second warning at a ground speed of 15 knots.

Most types of obstacle warning systems are prone to false warnings, and the laser diode tip strike system is no exception. Integration with other aircraft sensors and systems is therefore vital.

### Pilot Assist

Broadly speaking, several of the functions already described, e.g. tip strike warning, can be classified as falling under the heading of *Pilot Assist*, but in addition to those there is a group of similar functions that, on their own, are too small to merit a section. On the whole these concern aircraft performance issues such as what is its weight, where is its CG, what is its range, which engine has failed. Some of these may seem trivial, but assistance in determining them may save the pilot valuable seconds in an emergency.

A technology that has made a real impact here is that of the so-called *virtual sensor*. Here, data from sensors fitted to perform other functions is fused together by an algorithm to compute something entirely different. Examples have been given in previous sections, e.g. torque prediction as part of carefree handling, strain estimation as part of usage monitoring. Another to be mentioned is estimation of aircraft weight and CG position from air data, control positions, blade track data and a database of aircraft drag characteristics. Likewise, approach of the aircraft to the vortex ring state via the Wolkovitch (Reference 2) relationship.

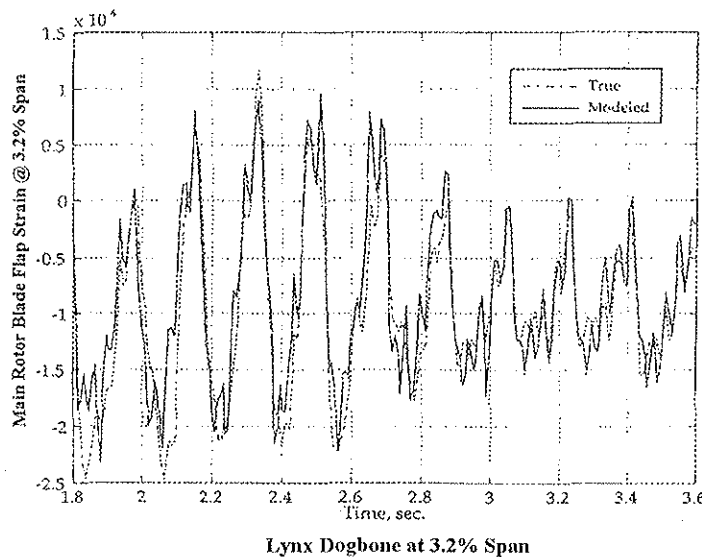


Figure 5 : Comparison of Estimated Versus Measured Strains

## Data Sharing

It is almost axiomatic that to get cost down, functions must share expensively won data to the maximum degree. Military avionic systems do this by putting shared data onto the 1553 bus, an expensive business and not particularly conducive to small system size and cost. As in any sharing system, compromises are also inevitable. Often it is in digitisation rate, one function wanting the shared data at a much higher rate than the other(s). Occasionally it is in data word length (accuracy) or freedom from noise. Finally, there is the certification level of the process to be considered and the fact that data at any time processed by a function at one of the lower levels *cannot* be passed for subsequent use by one at a higher level. However, the inverse does not hold - data gathered by a high level process *can* be passed on for use at lower levels. Certification level, digitisation rate and accuracy can therefore impose some quite stringent restrictions on the way data can be shared within an integrated system.

Table 4 shows the type of data sharing possible in a vehicle management system of moderate complexity. The data tends to be of two types - namely, low frequency data digitised at *control system rate*, typically 250 samples per second, and high frequency data digitised at anything up to hundreds of thousands of samples per second. It is only the low frequency data that is generally shared, though results *derived* from high frequency

data, e.g. obstacles in the path of the aircraft derived from gigahertz laser data, engine warnings from an electrostatic system etc, may appear in the low frequency streams. Data sharing of this extent is not of course unique - military 1553 bus systems have been doing it for years. The trick for commercial aircraft (and COTS military ones) is to do the same thing with much less hardware, and at much lower cost.

As mentioned above and now repeated, from a certification point of view it is important that data sharing take place on a 'cascade' basis. That is to say, data required for use at multiple levels of process certification must generally be brought into the system for processing at the highest level, used, and then passed down to the next lowest level that requires it.

## AVIONIC SYSTEMS

On their own probably none of the above technologies could make it into the typical helicopter on their own. The key is therefore an avionic processor system that can either accept signals from the function's appropriate sensor at marginal cost, or obtain its data by effectively riding on the back of some other function. This is first and foremost an issue of system design.

Table 4

Low Freq Parameter	Flight Data	Health	Usage	Gnd Prox	Avoid Curves	Tip Strikes	Carefree Handling	Weight CG
Roll rate	*	*	*	*	*	*		
Yaw rate	*	*	*	*	*	*		
Accel (X,Y,Z)	*	*	*	*	*		*	
Position	*			*				
Horiz. Vel.	*			*	*	*		*
Vert. Vel.	*	*	*	*	*		*	*
Pitch	*	*	*	*	*	*	*	*
Roll	*	*	*	*	*	*		
Heading	*			*				
Airspeed	*	*	*	*	*	*	*	*
Altitude	*	*	*	*	*	*	*	*
Gnd spd	*	*	*	*		*		
Torque	*	*	*				*	
Coll stick	*	*	*				*	*
Cyclic stick	*	*	*				*	*
Pedal	*	*	*				*	*
Fuel		*					*	*
TGT		*					*	
Wt on whls	*	*	*	*	*	*	*	*

A number of system design issues immediately arise. First, how is a system that incorporates tasks at several levels of qualification, different rates of evolutionary change and multiple levels of function 'duty factor' (i.e. operational all the time to operational only on pilot demand) going to be certificated for commercial use. That is easy for a multi-box solution - each is certificated to its own level. For the single box solution involving the sharing of data and processor resources it is much more difficult.

On top of the above we also have the increasingly important issue of electronic parts obsolescence, particularly the microprocessor. The typical lifetime of a processor, e.g. Intel Pentium or Motorola PowerPC is between 3-4 years, whereas that of the avionic unit may be as much as 25 years. Compound this with the shrinking size of the avionics market compared to games and cellular phones, add the increasing reluctance of electronic component suppliers to supply Mil-Spec or extended temperature range parts and the avionics manufacturer finds himself with a major problem of parts obsolescence. To some extent this is what has created the interest in COTS, or 'Commercial Off-The-Shelf' equipment.

#### Software Architectures

A radical solution being pursued by the author's company to both of the above problems involves moving towards the almost total separation of software from hardware. Why? Software development costs are typically running at about three times hardware costs, so any change in processor hardware must automatically have a large cost multiplier applied to it. To some extent this multiplier function can be reduced by use of an operating system, or a language such as Ada, but many problems still remain.

A route to almost total separation lies in application of the concept of the *virtual machine*. Here all code is compiled not to run on the target device, e.g. Intel 486, but rather a theoretical construct known as the *virtual processor*, the operation of which is thoroughly understood and checked by formal proof. The move to the target device is then accomplished by invoking another piece of software called the *translator*, which is used to convert the binary image created for the virtual machine into the equivalent image for the target device. The advantage to be gained lies in the very small size of the translator compared to the compiler, and the high confidence one can generate in the translator's accuracy once the translators for a variety of target devices can be cross-checked against one another.

The second step is to tackle of operating system and its kernel. Kernels able to support high level tasks are difficult to find. Ideally, for a VMS-type system not involving any sort of flight critical task one would perhaps like to see a Posix-compatible kernel. However, even for embedded systems these are typically still quite large in terms of Kbytes and so expensive to validate for avionic use. The goal is therefore a kernel that is small enough for formal proof to be a practicality. Kernel sizes of less than 10 Kbytes are the current goal.

The third and final step involves proving that the target system can meet all of its timing goals and that one task cannot affect another in any sort of adverse manner, e.g. by over-writing its memory space. The latter is probably best achieved through use of processors incorporating memory management, e.g. the PowerPC 603. The former is much more difficult.

A less than ideal solution to the proving of timing performance (i.e. showing that when a task is designed to run, it will be able to run) is formal proof via the Schuman-Pitt methodology. This involves first of all 'manual' timing of all of the tasks that must run, including those of the kernel. Secondly, definition of the task scheduler, e.g. first in/first out, rate monotonic. Finally, creation of the proof structure (typically written in Prolog). In theory, one could also use this to prove that one process could not corrupt the memory space of another, but that is perhaps pushing the technology too far for comfort (at least in the eyes of the certificating authority).

#### Hardware Architectures

With all software issues 'cleared up' as it were, the hardware side of life should become very easy. What in effect we want (and may now have gained) is total freedom to chose whatever processor, or mixture of processors, suites us in terms of environmental specification, cost, performance etc. The issue then becomes one of showing that we can produce a system suitable in terms of size, weight and cost for the helicopter.

The initial indications are that some extremely compact and low cost solutions are possible using the above technologies. Without even approaching the extravagances of multi-chip modules or special ASICs, we can probably now get all the processing power we would ever need for a full-up helicopter VMS system onto one 8 by 9 inch printed circuit card, with, if needs be, diversity of microprocessor hardware (i.e. a mixture of microprocessor types). For both maintainability and modularity reasons it is



possible that this board would be constructed as a mother board with daughter modules holding each of the processors and associated memory.

A combination of software and hardware such as outlined above would lead naturally to an integrated modular avionic architecture, or *IMA* system.

### CONCLUSIONS

The authors have presented a view of the future of HUM/FDR systems that, amongst other things, gives some of the system back to the pilot. He is the one that determines what the aircraft shall do, and therefore what its fatigue usage shall be. He is the one that suffers most in the event of a failure. He is the one that needs most help when the workload in the cockpit skyrockets because of some problem, often not of his making.

The paper also addresses the two important issues of system development and recurring cost. Software costs are in danger of becoming too large a fraction of aircraft cost. Component, particularly microprocessor, obsolescence and therefore high system sustaining cost is now a major problem. The certification cost of integrated systems is a major problem. For the helicopter to retain its market share the HUM manufacturer will have to play his part in addressing these problems.

HUM/FDR is still a relatively immature technology, but great strides have recently been made in raising its cost effectiveness. Manufacturers 'up to speed' with it are now talking about very significant reductions in aircraft direct operating and O&S cost. How far these will be achieved, and what is yet to come, remains to be seen. From a HUM system manufacturer's point of view it is encouraging to see all the large airframes now offering HUM, but much has still to be done in terms of achieving better fault coverage, producing more user-friendly ground-stations etc.

### REFERENCES

- 1 Padfield G D, 'Helicopter Flight Dynamics', Blackwell Science, 1996
- 2 Wolkovitch J, 'Analytic Prediction of Vortex-Ring Boundaries for Helicopters in Steep Descents', Journal of the American Helicopter Society, Vol 17, No 3, July 1972