The Effect of the Product Cost Factor on Error Handling in Industrial Robotics

N.W. Hardy* and **M.H. Lee**
Centre for Intelligent Systems
Department of Computer Science
University of Wales
Aberystwyth
(nwh@aber.ac.uk)

Abstract

The approaches to error handling in two contrasting industrial robot case studies are presented and discussed.

Introduction

We are interested in the role of advanced software tools and techniques in achieving robust operation of robotic systems in industrial and commercial applications. In this context, errors which halt production are costly and systems must be carefully designed to cope with any shortcomings during production in ways that are cost effective for the particular application.

We present two case studies from industry to illustrate our approach to error handling. The case studies show how production constraints can influence the priorities for solution of given robot manipulation problems.

The InFACT Assembly Machine

The project

The InFACT project ran from 1989 until 1991 with a short feasibility study during 1988. It had a budget of £6.8M and was funded under the multinational EUREKA/FAMOS program as Project 321. Commercial and academic collaborators from the UK, France, Italy and Austria were in the consortium [Loughlin, 1992]. Commercial exploitation of many aspects of the project is now under way.

The aim of the project was to develop a flexible assembly machine incorporating sensor systems, parts and components feeders, manipulators and assembly devices. The machine was to be suitable for the assembly of a wide range of electo-mechanical products of up to 300mm cubed in size. The design criteria laid down at the outset characterise the machine and are as follows [InFACT Group, 1992].

- **Economic viability**
  The ability to perform the basic tasks associated with the assembly of products at a cost which is competitive with existing labour rates.
- **Flexibility**
  A high degree of reusable hardware and software.
  Easy reconfiguration for different manufacturing requirements.
  The capability of dynamic reconfiguration allowing the changeover from assembling one product variant to another within one cycle time.
- **Reliability**
  Low equipment fault rates.
  The ability to deal with faults.
- **Meeting user requirements**
  A safe and easy to operate user interface which accommodates existing industrial skill levels.
  Ease of maintenance.

An important aspect to note is that the machine would be highly integrated, covering all aspects of the assembly process and that a reasonably high level of task programming would be required to allow non-specialist programmers to handle the diversity and number of devices in the machine and deliver robust performance. Coupled with this, i) economic viability in the mass assembly market is crucial and ii) errors must be dealt with (not necessarily corrected). The implication of these two points is that throughput rates of correctly assembled products would tend to be more important than the proportion of failed assemblies. The trade-off between rate of production and rate of failure would clearly be influenced by many factors but with reasonable failure rates and the typical cost of parts used in the intended product range, 100% success is unlikely to be an appropriate target.

The Hardware Configuration

The machine has three zones. The central or **main** zone houses a four degrees of freedom cartesian manipulator designed for rapid, light and precise operations. The **power operations** zone houses a three degrees of freedom manipulator designed for operations requiring high forces or the use of specialist tools. The **supply zone** houses a specially designed modular system allowing a number of stacks of pallets. This provides the entry point for parts pallets and the exit point for finished products and rejected parts and assemblies. To link the three zones a linear shuttle device runs beneath them. The shuttle is able to carry two pallets, raise them into positions in the floors of the two manipulator zones and later retrieve them. The main zone has five such positions and the power operations zone two. The shuttle can acquire and replace pallets at the base of stacks.

Up to 20 linear vibratory feeders with exchangeable tracks can be sited around the two manipulator zones. These provide an attractive alternative to pallet feeding for suitable small parts.

The end of arm tooling design for the main manipulator incorporated exchangeable tools and programmable compliance with six degrees of freedom. [Selke, 1992] The demonstration machine used a device with a subset of the original facilities. End of arm tooling on the power operations manipulator again benefits from a standard physical and control interface to the machine.

Many sensors are provided, both for local control of particular actuators and to make the sensor data are available to the assembly task programmer. The only standard devices provided for measurement operations are simple binary touch probes [Bell, 1992].

The design of the machine developed hand-in-hand with a particular policy for internal scheduling. Assembly takes place on specialised pallets known as fixtures. Assembly will be proceeding in parallel on three fixtures. One will be located in the power operations zone and one in the main manipulator zone. The third will be in the process of being repositioned in order that it be ready for the main manipulator to move on to it and not be idle while transportation takes place. The most cost-effective operation of the machine will be when a number of assemblies are produced on each fixture and the principle of adding one part to all the assemblies on one fixture before continuing to another part was adopted.

Task programming

The machine design criteria concerning the ease and robustness of task programming by production staff implied that an alternative to a language based task programming environment was required. Using on experience gained in earlier projects [Barnes *et al.*, 1983, Hardy and Barnes, 1992, Orgill *et al.*, 1989, Williams, 1985, Williams and Lill, 1987] it was based on so called generic tasks. These can be considered as highly parameterised subroutines embodying well designed and tested software, assembly expertise and best practice for exploiting the design and configuration of the assembly machine. Further and highly effective assistance can be given through the provision of an inventory. This is a database containing details on all elements of the machine the tools the components and the product. Data are added to the inventory in a systematic way and may be used repeatedly to provide parameters for generic tasks.

In addition to a coherent database, an assembly specification consists of a sequence of objects. These are instances of parts known to the database and have associated sequences of generic tasks with appropriate parameter values. The job of the task programmer is to create a sequence of objects which represents the assembly sequence and then to select and provide parameters for a sequence of task generics to specify the actions to be carried out with the part.

Error handling - initial considerations

From the earliest stages of the project the term resilience was used. This covered behaviour of the machine, in non-standard circumstances, intended to overcome the problems which would otherwise arise. Broadly three levels of such activity were recognised.

1. Halting the machine with good quality information for the operator.
2. Abandonment of the current assembly but continuation of production without operator intervention.
3. Salvaging of the assembly in question.

Level 1 was clearly undesirable within the InFACT objectives, though high quality diagnostics in an assembly context were seen as valuable if this eventuality arose. Design progressed on the basis of levels 2 and 3.

Six classes of technique to improve resilience were considered (it was recognised that the classes were not all independent).

1. Correction for deviation from nominal positions The variability in positioning and dimensions of fed parts must be handled. Three approaches were considered. In the light of experience with earlier machines it was considered that **active compliance** would not be a necessary or effective technique. A clear specification of the required **passive compliance** capabilities was established. The type and level of **measurement and correction techniques** was a matter for investigation during the project.

2. Explicitly programmed user specified conditions and responses This technique, which reflected the standard industrial approach was not considered appropriate. It requires conditional constructs within the task programming environment and typically requires significant analysis, design and coding effort by the task programmer. It fails to capitalise on available assembly expertise for the benefit of future assemblies and clashed with the fundamental InFACT philosophy of well engineered high level programming primitives.

3. Good design The mechanical and control components of the machine were designed from the outset with a clear set of functional requirements linked closely to the applications. It was anticipated that design for assembly would be employed wherever possible.

The design of activity within the cell must be good. This was seen at two levels. First, the individual generic tasks should be well designed. Second, sequences of these tasks need to be well designed to work at least in cooperation and preferably synergistically. This was expected to be achieved through guiding and constraining the user.

4. Pre-programmed generic responses A natural extension to a system based on a set of generic tasks is that tasks should include generic responses associated with particular conditions. This approach had been taken in earlier work and it was expected to play a part in InFACT. Two broad types of response may be considered. Hidden responses which are broadly implicit in the functional definition of the action (such as retries and searches) and more apparent responses where the effect of the action is altered (such as discarding suspected faulty components and obtaining new ones or re-grasping).

out of tolerance, i.e. $p_x > d_x + t_x$ or $p_x < d_x - t_x$ should be rejected. Other product may closely match the pattern, i.e. $p_x = d_x$ and should be marked for acquisition, while a third class will fall within the tolerance band, i.e. $p_x < d_x + t_x$ and $p_x > d_x - t_x$. These marginal cases will provide material for learning the scope of the grasping function — they will be processed for acquisition but the results will provide information that can be used to update the tolerance function. We notice a fourth possibility: any miss-feed of different product, such as might occur in mixed batches, may cause product of type P_y to be recognised during a run of P_x. This will be treated as a reject case and logged.

Acquisition (Grasping)

The present end-effectors consist of grippers with fixed-throw, two-state pneumatic fingers. This limits the outcome of grasping operations to three possibilities. Either the grasp is successful (product position being under full control), grasping fails during retraction from the conveyor surface, or the product position is lost during transport or delivery. The last failure is much more serious than failing to pick from the conveyor belt. Loss of positional control may mean either complete product loss (dropping the product from the gripper) or errors in location and/or orientation (partial gripping, twisted product, etc.)

Error Handling

We can now examine the scope for error detection and recovery. There are four cases where faulty product may cause errors: (a) the product is discovered to be faulty before acquisition is attempted, (b) the product is not picked correctly from the conveyor, (c) the product is lost during transport, and (d) the product is lost during packing. We notice that only case (a) can be detected without additional sensors.

The application factors which determine the form of appropriate error treatment are:

Low product cost The product has negligible cost. Any damaged, misshapen, oversized or undersized product should be rejected. We define the incoming product rate as F_{prod} and the reject rate as R_{prod}.

High production rates Compared with manufacturing applications, food processing batch sizes are very large. Benefits are only gained for low cost product when huge quantities are processed. Thus, it is essential for maintaining viability and competitiveness that very high production rates are achieved. This means errors must be handled at similarly rapid rates.

Low error rate in output packages Even when packed the value of the product is still small. However, any faulty packing could incur serious penalties if the product reaches the customer. It is important that no packed product emerges which has missing items, bad orientation, damaged items, etc. Hence the error rate of the outgoing product, E_{pack} must be minimised.

From the above it is seen that there is no scope for any significant processing during error handling. As the cost of any error treatment is bound to be higher than the cost of the product there is no justification for attempting to process any suspect product. Consequently the sole mechanism available for dealing with errors is rejection. In other words, error rates must be kept low, even at the expense of the reject rate, i.e 'reject on any hint of error'. It is usually found that there is a trade off between errors and rejects, as high quality can often be gained at the expense of severe rejection criteria. In this application E_{pack} must be minimised by using R_{prod} and R_{pack} to prevent errors reaching the output.

Returning to the above four classes of errors, suitable recovery actions are: for (a) ignore product (it will pass on to the end of the belt) — this increases R_{prod}; for (b) open gripper, ignore and try next product — this also increases R_{prod}; for (c) and (d) send signal to reject the current packing and start a new one — this increases R_{pack}.

From this analysis the following factors emerge as key design parameters for any proposed error handling processes:

Reject Propensity As the product has almost negligible value in its prepackaged state it can be rejected at the slightest hint of an error. With high volume there will be a non-trivial proportion of faulty product that is presented for handling. It is essential that this is rejected without engaging handling; recovery is a high risk operation and would be extremely costly. Fortunately detection and rejection of faulty product in the feed is easily accomplished by the vision sensor and conveyor combination.

Effective Grasping Once a decision to grasp the product with the robot has been made it is vital that the grasp is optimised and is as effective as possible. Some form of additional sensing is necessary if grasping errors are to be detected.

Product Variation Despite the controls inherent in mass production some uncertainty in product shape and size can still be manifest. While large variations will be treated as rejects, smaller changes should be accommodated. Grasping strategies must be able to respond to local variations and longer term trends in product dimensions.

Learning and Adaptation

There are two forms of learning that are appropriate to the task: continuous adaptation to small variation in product dimensions so as to optimise the success of grasping, and the utilisation of previous knowledge of gripper/product combinations so as to find effective grasp sites for new products. We are attempting to incorporate both forms of learning.

We are using a learning method known as the 'Collateral Architecture' to fine tune the grasping strategy for a given product; this is a method of adjusting the tolerance function, t_x, to capture the best fit from the cumulative experience gained during production runs. Learning methods are also being used to select grasping parameters for different product types. Different grasping strategies are selected by this process and learning is employed to capture and utilise experience of previous grasp performance. By this means a decision surface, d_x is selected for a given product type, P_x, using the database of gripper/product experience.

ics estimate corrections for this. All generics used after a measure automatically take the correction into account. [Nicholls and Hardy, 1992]

The expectation that well designed application specific jigs could be provided at acceptable cost. The provision of simple passive jigs is considered an acceptable cost in the context of the development of a new product. If an active jig or a jig incorporating sensing is required, this could significantly increase the cost. To reduce this a simple general purpose interface is provided at each fixture position.

Appropriate and programmable compliance in the gripper mechanism. The compliance devices designed and used in InFACT are described above.

General purpose sensing to detect incorrect assemblies provided as part of the task programming environment. A number of permanent monitors were provided to detect general problems such as major collisions. Beyond that each generic task explicitly uses appropriate sensors to check as far as possible that its intended purpose has been achieved. The types of checks performed are those which would be expected for standard current industrial applications.

The abandonment of an assembly where an error is detected and its removal using a parameterised routine. Any mis-match between a sensed value and the expected value causes the generic task to be abandoned and the associated assembly to be marked as such. A record of the stage at which the assembly was abandoned is also made.

The final stage in an assembly involves the transfer of completed assemblies from the fixture (which will begin the cycle again) to a pallet which will be taken to a product-out stack. At this stage any abandoned assemblies must be disposed of. To achieve this, each object in the assembly sequence may have a disposal generic associated with it, in addition to the generics specifying its normal manipulation. The disposal generic is a simple pick and place task specifying how to pick up that object and transfer it to a disposal pallet. This pallet is likely to be little more than a box, so accurate placement is not required. Each object in the sequence must have a disposal sequence defined. This is essentially a list of references to those objects whose disposal generics must be executed to remove all the parts from the fixture.

This approach appears to place a considerable burden on the task programmer. This burden is not unacceptable heavy for two reasons. First, the parameter data required are simple and probably already in use. Second, very rarely will it be necessary to move an abandoned assembly part by part. Many products have a base or enclosure which means that most parts can be removed by grasping it.

Robotic Food Handling

This problem originates in the requirements of a food manufacturer to automate the packing of frozen food products. It is being investigated under a project supported by the UK Science and Engineering Research Council.

Equipment Configuration

An Adept robot is stationed over a conveyor belt which supplies individual frozen product pieces. A vision system is placed upstream of the robot and uses standard vision algorithms to recognise features of the product shape. The robot then uses the coordinates of the product and the belt position to grasp the product and transport it to a packing location. Normally several product are packed into one box.

Implicit Constraints

The configuration of the work-cell is essentially fixed and all sensing and error recovery is to be carried out in the context of existing equipment. The viewing angle of the camera restricts vision to two dimensional overhead images, consequently, in all that follows, we treat product as laminae, i.e. flat plates of uniform thickness with variable shape outline.

Operational Requirements

The normal mode of operation of the handling cell requires high speed production and rapid changes of product batch, with fast gripper changes and minimum setup time. The controls and user interface are to be designed for absolutely minimal operator involvement.

The Robot Handling Task

There are a range of different product types, $P_1 \ldots P_n$, each of which can be handled by one of the available gripping tools, $G_1 \ldots G_k$, where $k \leq n$. Each batch run uses only one gripper and deals with a single product type. Gripper changes are made at setup time.

Analysis of the task identifies 7 stages:

FEED Present product to cell. This is achieved by the conveyor.

LOCATE The vision system gains information on product position, orientation, size and shape.

ACQUIRE Gain positional control over product. This is achieved by successful grasping.

ORIENT Align product with given reference frame. Achieved by Robot action.

TRANSPORT Move product to spatial location. Achieved by Robot action.

DELIVER Mate or place product at target location. Achieved by Robot action.

INSPECT Measure product/task parameters. None specified other than rejection of faulty product.

Of these operations only LOCATE and ACQUIRE provide opportunities for sensing and correcting errors. Let us examine the functions of these two stages.

Location (Vision)

Each product type P_x will have an associated decision surface, d_x in the space of measured parameters. Let there be an associated tolerance function, t_x. Then product that are

5. Flexible pre-programmed responses The generic response approach may be enhanced by the use of contextual data to select and parameterise generics leading to less rigid and more relevant responses. This method follows naturally from the concept of tasks generics and the inventory database. Suitably designed responses could be selected and executed to give a significant range of behaviour and this approach would be investigated and exploited in InFACT.

6. On-line recovery planning Techniques to plan corrective actions were considered. The cost of such planning would be important. From an early stage it was proposed that such techniques should not be used in InFACT for three reasons. First, they are relatively expensive to implement both at the systems level and for the task programmer who must assemble a large knowledge base to support them. Second, while certain aspects of the problem could be tackled in this way, others could not and a significant research component was implied in the approach. Most crucially, the value to InFACT in terms of enhanced performance would be relatively small.

Error handling - the development of understanding

The process of design of the final InFACT machine was complex one and involved the resolution of a large set of often contradictory pressures. Tracing in detail the steps by which initial ideas and understanding were transformed into the final decisions is a major task and beyond the scope of this paper. The following trends can however be recognised in retrospect.

The ease and economic viability of providing **physical restraint** within the machine compared to the problems and cost of handling positional variation by means of sensing the software techniques were increasingly recognised.

The **cost of rejection** of a possibly faulty part or abandonment of a failed assembly was taking increasingly into consideration. Many components are cheap and in addition such rejection has possible advantages for quality assurance. Manual correction and completion of assemblies can be a cost effective mechanism.

The range of possible errors is related to the **flexibility of the machine**. A machine capable of a wider range of activity will clearly be open to a wider range of errors.

In establishing a **range of activities** of which InFACT would be capable there was a strong tendency to reduce and simplify. Many proposed facilities were recognised to be of use only in a small proportion of real assembly tasks. Their omission would not significantly reduce the utility of the machine but would simplify implementation and reduce the scope for error.

The **probability of success of any corrective action** will obviously not be 100%. When the possibilities of incorrect diagnosis of the problem and of having a fundamentally insoluble problem at a given level of the assembly are considered success rates could be quite low. When combined with what should be a low basic error rate the achievable increase in throughput may be small.

The **cost of corrective action** was increasingly seen to be unacceptably high in many cases. The target time (achieved for a range of products) for the inclusion of one part in a single assembly was 3s. Any lengthening of this begins to challenge the economic viability of the machine. A doubling of the time (as might be caused by a complete and successful retry) for a small number of parts would not have a significant overall effect on throughput when spread across all the parts in all the assemblies. This however is not the real time cost of corrective actions which is related to the scheduling of the whole machine. High levels of activity can be achieved within the cell by relatively simple fixed sequencing of the activity of the manipulators and shuttle. Investigations revealed that deviation from that schedule would typically result in a large drop in throughput. If a parts pallet becomes empty during work on a fixture, the arm will be inactive while a new pallet is introduced. A policy of carrying on parts pallets an integer multiple of the number of parts required by a fixture was used and any strategy which exhausted a pallet prematurely would disrupt the schedule by delaying the arm. Secondary disruption is likely to result from the use of the shuttle for replenishment and from any attempt to use the arm for work on another fixture while more parts are fetched.

There are two alternatives to the above situation. First all parts pallets could carry a small excess calculated to handle the expected worst error rate for that part. This approach was not explored in detail but was expected to lead to a high rate of parts recycling and a loss of space within the machine. The second would be a dynamic scheduling regime. This could not be investigated in the scope of the project and its potential for the machine as configured is not known. It seems unlikely that such a system could achieve utilisation and throughput rates as high as those achieved by the fixed scheduling.

Error handling - the final design

The final InFACT design included the following features as a result of the above considerations.

High but reasonable levels of constraint on the quality and positioning of fed parts. Two parts feeding methods are provided. The linear vibratory feeders have replaceable tracks which can be automatically designed and machined starting with basic designs for the parts.[Dick and Lo, 1989] They are able to reliably and repeatably present small parts at a fixed location. The basic pallet can be enhanced by simple and cheap means such as vacuum formed inserts to provide accurate parts specific presentation. These part specific adaptations are considered unacceptable costs.

A careful characterisation of classes of remaining uncertainties in feed positions and the provision of sensory mechanisms to remove each class. The feeding mechanisms were recognised to leave positional uncertainties of a limited and simple set of types. These were matched one to one by a set of generic measurement tasks each designed to remove a particular uncertainty. All the tasks use the touch probes. Feed position errors result in deviations from nominal grasp configurations and the measurement gener-

It is clear that detection of successful grasping is vital for any treatment of those errors that affect E_{pack}. With vision as the only sensor it is not possible to detect any handling operation that fails. Hence the tolerance bands must be held very wide in order to reduce potential risk. This can lead to an unacceptably high reject rate.

By using a simple 'product present' sensor in the gripper it is possible to confirm correct gripping, detect faulty gripping and discover lost product during transport and delivery. Other more sensitive and sophisticated sensors provide more information (such as twisted or skewed grasping of the product) but they also incur more expense, complexity and computer processing power. These include: finger contact sensors, finger displacement detectors and finger force sensors.

It is also useful to consider the levels of actuation that are available for handling. Grasping and action strategies are constrained by the available motor functions. The existing gripper is a design based on fixed-throw, fixed-centre, binary-actuation fingers. This gives three control variables: x, y, and θ, for defining the position and orientation of the target grasp site on the conveyor. The only strategies for grasping consist of variations in approach by changing these values, (we note that re-grasping strategies may have some merit). We are currently considering more dexterous grippers which will allow more scope for flexible strategies. For example, control over the initial aperture of the fingers would allow more product ranges to be accommodated within one gripper, thus reducing the number of gripper changes. Some adaptation for product variation might also be facilitated, as minor aperture adjustments can cause corresponding changes in gripping force. This would offer four control variables: x, y, θ and r, where r is the initial opening radius for all fingers. As the throw, t, is fixed, all fingers close on a radius of $r - t$. Another design involves linear-actuation fingers. In this variant the closing of the fingers is performed by an electric motor or other drive such that the fingers can be driven to any position between fully open and fully closed. This gives five control variables: x, y, θ, r and t and much more opportunity for dexterous grasping strategies, owing to the enhanced performance and control of finger movement.

We are currently exploring the cost/benefit trade off between reduced error rates and additional sensory-actuator facilities. It has been shown that the task requirements provide constraints that may severely limit justifications for complicated error processing and the values of the various product cost factors are of great importance. In the present study we have argued that considerable potential benefit is offered by the addition of some simple sensing and learning procedures. We hope to report on experiments based on implementations of these ideas in the near future.

Acknowledgements

Although in a position of overall responsibility for the handling of errors in the InFACT machine the authors wish to make clear that the development of the InFACT design was undertaken by a large team of which they were a part. In particular we thank Tony Williams, the technical manager of the project and source of vital expertise and inspiration. We also acknowledge the contributions of the team on the food handling project:- Jem Rowland, Marcos Rodrigues and Youfu Li.

References

[Barnes *et al.*, 1983] D. P. Barnes, M. H. Lee, and N. W. Hardy. A control and monitoring system for multiple-sensor industrial robots. In *Proc. 3rd. Int. Conf. Robot Vision and Sensory Controls*, pages 471–479, Cambridge, MA. USA., 1983.

[Bell, 1992] Steve Bell. Touch probes: The vision argument - sophistication vs simplicity. In Clive Loughlin, editor, *InFACT: Project, Concepts, Machine.*, pages 46–47. MCP University Press Limited., Bradford, UK, 1992.

[Dick and Lo, 1989] M.D. Dick and E.K. Lo. Computer aided design and manufacture of a feeding system for robotic assembly. In *Proc. Int. Conf. Manufacturing Science and Technology*, Sweden, 1989.

[Hardy and Barnes, 1992] N.W. Hardy and B.P. Barnes. Affirm: Knowledge based robot supervision. Technical Report UCW-CS-TR-RRG-200-92, Department of Computer Science, University College of Wales, Aberystwyth, UK, April 1992.

[InFACT Group, 1992] InFACT Group. Infact. Integrated flexible assembly cell technology. Final report. Bristol Polytechnic., Coldharbour Lane, Bristol, UK., February 1992.

[Loughlin, 1992] Clive Loughlin, editor. *InFACT: Project, Concepts, Machine.*, Bradford, UK, 1992. MCP University Press Limited. 59 pages.

[Nicholls and Hardy, 1992] H.R. Nicholls and N.W. Hardy. Distributed touch sensing. In H.R. Nicholls, editor, *Advanced Tactile Sensing for Robotics*, chapter 6, pages 107–122. World Scientific, Singapore, 1992.

[Orgill *et al.*, 1989] C.H. Orgill, N.W. Hardy, M.H. Lee, and K.A.I. Sharpe. An application of a multiple agent system for flexible assemble tasks. In *Knowledge based envirnments for industrial applications including cooperating expert systems in control.* IEE London, 1989.

[Selke, 1992] Klaus Selke. End effector. In Clive Loughlin, editor, *InFACT: Project, Concepts, Machine.*, pages 35–39. MCP University Press Limited., Bradford, UK, 1992.

[Williams and Lill, 1987] A.M. Williams and B.H. Lill. Commercially available flexible assembly cell. In *Proc. BRA-10. Birmingham UK*, 1987.

[Williams, 1985] A.M. Williams. A flexible assembly cell. In *Proc.* 6^{th} *ICAA, Birmingham UK*, 1985.