Proceedings of the 1996 IEEE
International Conference on Robotics and Automation
Minneapolis, Minnesota - April 1996

LUCIE the Robot Excavator - Design for System Safety

Derek Seward & Frank Margrave, Department of Engineering
Ian Sommerville & Richard Morrey, Department of Computing
Lancaster University, U K

1.0 Background

Staff and students at Lancaster University have, for the past five years, been involved in the development of an autonomous robot excavator - LUCIE - the **L**ancaster **U**niversity **C**omputerised **I**ntelligent **E**xcavator. An excavator provides a good opportunity for development, as it is basically a highly efficient and well developed four degree-of-freedom manipulator arm, but with the complete absence of automation or intelligence. The aim of the project is to add autonomy in order to produce a robot excavator with the following characteristics:

- It should concentrate on the task of **trenching**, and be able to produce a good quality and accurate smooth-bottomed trench.
- It should adapt to different soil types without human intervention.
- It should cope with obstructions, such as boulders in the trench.
- It should eventually be a self-contained system with no cables to external computers.

The first stage of the work was sponsored by the U.K. **Engineering and Physical Sciences research Council (EPSRC)** and involved site studies of the excavation process [Green 1990]. The techniques and strategies of skilled drivers were observed and analysed. A working hydraulic fifth-scale model was constructed - see figure 1 - and this enabled automated digging strategies to be developed in the laboratory.

A full-sized rapid prototype was then produced using largely off-the-shelf components The purpose of the prototype was to prove the concept and to further refine the system requirements. A hardware platform was provided by the JCB excavator company in the form of a JCB 801 tracked mini-excavator - see figure 2. An identical system architecture to the fifth-scale model was adopted which enabled software to be transferred easily from one to the other. The main processor used was a Harris RTX2000 communicating via a standard industrial STE (IEEE 1000) bus. This processor is a high speed device optimised for the FORTH computer language. For more details see [Bracewell 1990] & [Seward 1992]. Although the rapid prototype met the initial aims of the project listed above, the solution lacked compactness, robustness and was reasonably expensive in hardware terms. Neither was the rapid prototype mobile - i.e. the arm was under computer control but no attempt was made to control the tracks.

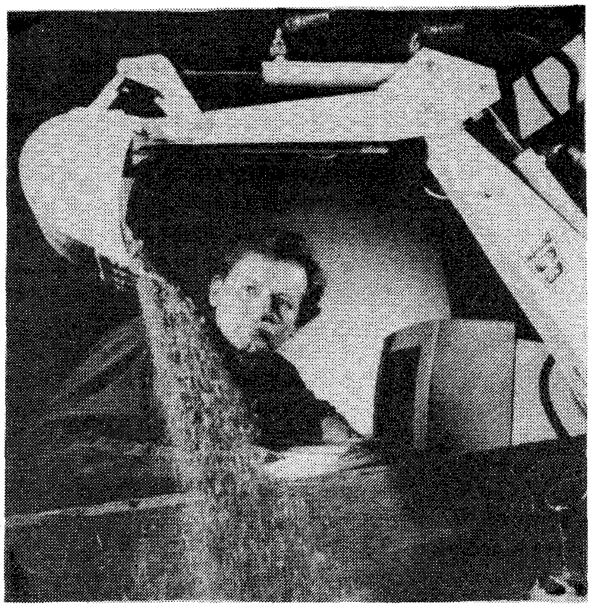

Figure 1 - Fifth scale hydraulic model

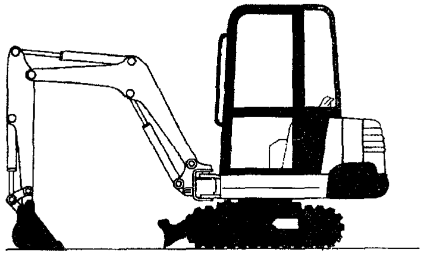

Figure 2 - The JCB 801 mini-excavator

Having shown that the initial aims are realistic it was decided to re-engineer LUCIE in a more robust and professional manner - the next stage being referred to as a **development prototype**. The hardware is described in

0-7803-2988-4/96 $4.00 © 1996 IEEE

section 3.0. It was also decided to make LUCIE mobile by extending automation to the tracks. This immediately emphasises the problems of safety with such large and powerful mobile robots. It is the approach to these safety problems that forms the bulk of the remainder of this paper.

2.0 Basic software architecture

One of the most important outcomes of the rapid prototype was the effective high level decomposition of the system into discrete modules. A useful guide to minimising coupling between modules is to consider the points where a human would intervene in the system in the event of a failure. This is shown in figure 3. The safety manager is not shown connected at this stage as it has a unique status. This is discussed in more detail later.

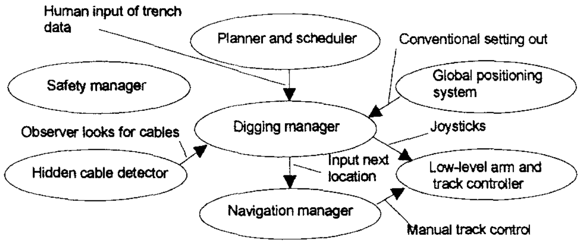

Figure 3 - Top-level decomposition with points of human intervention

Two of the above modules, the low-level arm controller and the digging manager will be considered in more detail.

The purpose of the low-level arm controller is to take movement commands from the digging manager and send the appropriate control signals to the electro-hydraulic valves. Experiments with the fifth-scale model revealed the desirability of a dual control strategy which is closely reflected in the human approach to digging.

When moving in air (i.e. tipping the spoil and positioning the bucket teeth) a **positional controller** is required. The commands from the activities manager thus instruct the bucket to move to specific X,Y,Z co-ordinates in space. Angle sensors were placed on the arm joints to provide closed-loop control.

When moving in soil a **velocity controller** is required. The commands from the activities manager instruct the tip of the bucket teeth to comply with a particular velocity vector (i.e. speed and direction). This strategy accepts the fact that movement in the ground needs to be highly adaptive as ground conditions change, and that there is little likelihood of reaching a specific point via a predetermined path. Error feedback is used by the activities manager to modify the velocity command in order to optimise performance. Thus if the excavator cannot achieve the demanded velocity because the ground is too hard, the activities manager will direct the low-level controller to attempt a shallower dig where the ground is expected to be softer. This approach has proved very effective in providing pseudo-force feedback without the need of additional force sensors. The low-level arm controller is currently implemented in "C".

Of the above high-level modules, it is most difficult to provide an early detailed requirements specification for the **digging manager**. The digging manager is the module that directs the digging process and has knowledge of the tactics required for efficient operation. To help the prototyping of the digging manager a design platform concept was used. The design platform allows the developers to try out and modify ideas, as well as reacting swiftly to requirements changes in other system components.

The aim of the design platform is to provide maximum flexibility without compromising on maintainability. Maintainability is essential not only because of the potentially fast and possibly radical prototyping process, but also because of the unstable nature of developing the system using students. The purpose is to produce a detailed and static specification of the activities manager module. This specification is then used to produce an optimised and well engineered software solution.

In order to construct a design platform, it is necessary to have at least a basic understanding of the robotic system and the high level goals of the control software. Most useful intelligent robots will be **finite state machines**. These are systems which are in one or other particular state of activity depending upon the stimuli received. These stimuli can be as a result of signals from sensors, timers, switches or work instructions from a higher level programme. The stimuli trigger the switch from one state to another. Figure 4 shows a state transition diagram for "digging within reach". The words inside the boxes describe particular states and the words in italics outside the boxes indicate the stimuli that triggers the transition from one state to another. The digging manager is implemented using the well known AI technique of a **production system** [Seward 1992] in ADA. The semi-formalism of this technique assists in making the safety case for the robot. About seventy production rules are required for excavation, but because the system is a finite state machine only a sub-set of the rules needs to be considered within any particular state.

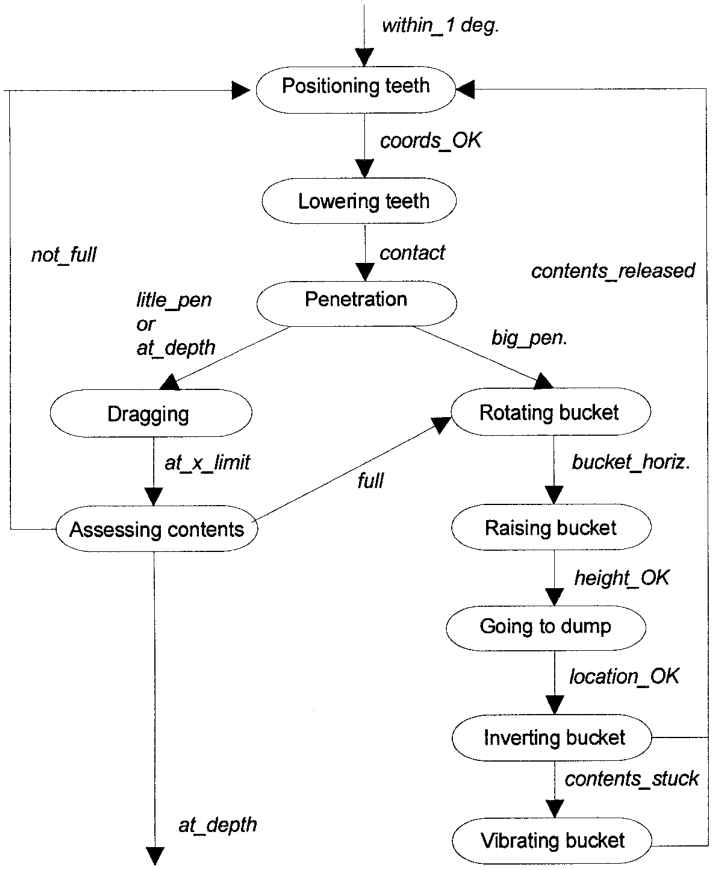

Figure 4 - finite state transition diagram

3.0 System hardware

The software architecture shown above leads naturally to a hardware architecture. A conventional Intel 486 based system has been adopted - initially based around three processors. One for the digging and navigation manager, one for the low-level controllers and one for the safety manager. The ultra-compact PC104 format is used with the processors communicating via a CAN bus.

The following sensors are provided:

- Four potentiometers on the joints for angle measurement
- A tilt sensor for reasons of safety and levelling the trench
- An obstacle detection sensor (see later)
- Bump sensors
- A Trimble 7400MSi series satellite GPS for location and navigation

These sensors are currently hard-wired, but it is the long-term intention to convert all sensors to "intelligent" sensors communicating via the CAN bus.

3.1 Obstacle detection sensor

There are potentially two types of sensor available to the project for the detection of surface obstacles:

(a) Standard 'bump' sensors, which can be fitted to the extremities of the vehicle and are activated only when they actually come into contact with an obstacle

(b) The Leuze RotoScan RS 3 optical distance sensor, which can detected obstacles up to a distance of 15 metres.

LUCIE's sensing capabilities will be based on a combination of these two types of sensor.

The Leuze RotoScan RS3 works by using two lasers which scan through 90° thus providing a semi-circle of coverage. Objects greater than 7 cm wide are detected up to a 25 m range. The area is swept at 10hz and the precise position of objects output in serial form to the computer

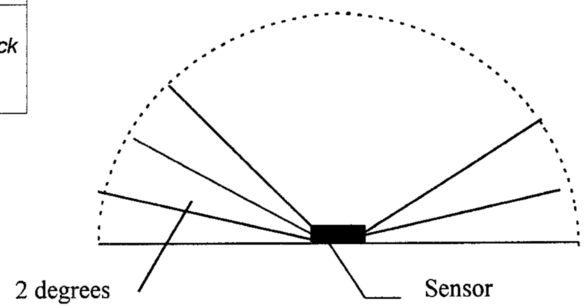

Figure 4 - The Leuze RotoScan Sensor

There are two important limitations which impinge on how the sensor can be deployed:

(I) The range in the horizontal plane is very narrow -- more-or-less negligible; so it is possible for objects lower than the sensor to be missed, as well as those entirely above it - overhanging branches, birds etc.

(2) The sensor only detects the obstacle closest to it. This means that if any part of LUCIE (the boom for example) cuts across the sensor's field of vision, temporary blind spots will be created beyond which any obstacles will remain undetected.

These limitations mean that the single RotoScan we are likely to have available must be positioned carefully. It will be placed on the top of the cab at the front so that it monitors the region in which the boom moves when the cab is stationary, and when the cab rotates, the 'leading edge' of the sensor range will monitor the space into which the cab is moving. See figure 5.

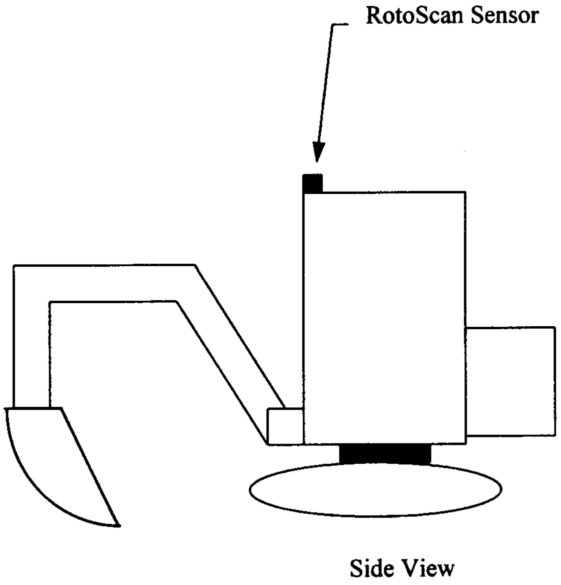

Figure 5 - Optimum sensor position

This overcomes the problem of the boom triggering the sensor for most applications.

4.0 The safety problem

There is currently a great deal of interest and active research throughout the world in the field of large mobile robots. Applications range from firefighting, handling of hazardous materials, nuclear de-commissioning and sub-sea activity to general construction robots. When the time comes for such technologies to reach the marketplace, safety will be a vital issue. Indeed unless considerable research effort is put into addressing the safety issues, it is conceivable that the future exploitation of such robots will be severely handicapped.

Robots in the above categories differ from conventional industrial robots in four key ways, all of which have very important implications for system safety:-

- Mobility
- Higher power to weight ratios
- More intelligence - to provide autonomy to tackle less well defined problems
- More external sensors - to determine appropriate behaviour in unstructured environments

In addition, the behaviour of these robots must be considered to be **non-deterministic** for the following reasons:-

1. The end-user may need the facility to modify the behaviour of the robot in order to 'train' it to carry out new tasks.
2. The use of heuristic rules is probably essential for flexible operation.
3. They will operate in unpredictable and unstructured environments.

The report "Safety and Standards for Advanced Robots - a First Exposition" [Advanced 1992] highlights the following fundamental dilemma facing advanced robot development:-

"Certain functions of an advanced robot i.e. its ability to interact with a dynamically changing world, cannot readily be achieved other than by the use of symbolic software representations. To mandate the use of formal methods is in effect to deny this functionality.... "

"The issue of artificial intelligence in safety critical applications causes concern and has been side-stepped in the existing standards committees, although they are aware of the problems. "

This conflict is being addressed by means of a software **safety manager**. This is conceived as an independent distinct entity, whose job it is to monitor the environment, and give permission for all behaviour which could have a safety critical component. This is a **behaviourist approach** in that it is concerned with achieving safe behaviour, but is not concerned with the processes that determine functional behaviour. Clearly, in the interests of efficiency and reliability, the processes that control functional behaviour should be rigorously designed using the best software engineering practices to maximise safe behaviour. Ultimately the safety manager is, however, responsible, and will block all actions that might create a hazard.

5.0 Developing the safety case

The process of developing safe systems is described in the "Safety Lifecycle Model" [IEC 1992], however European work [Redmill 1989] has produced a model which greatly expands the early steps that lead up to the creation of a suitable safe system requirements specification.

This is shown in modified form in figure 7. The starting point is the creation of five documents that contain the necessary data to carry out a safety analysis.

5.1. Robot physical characteristics

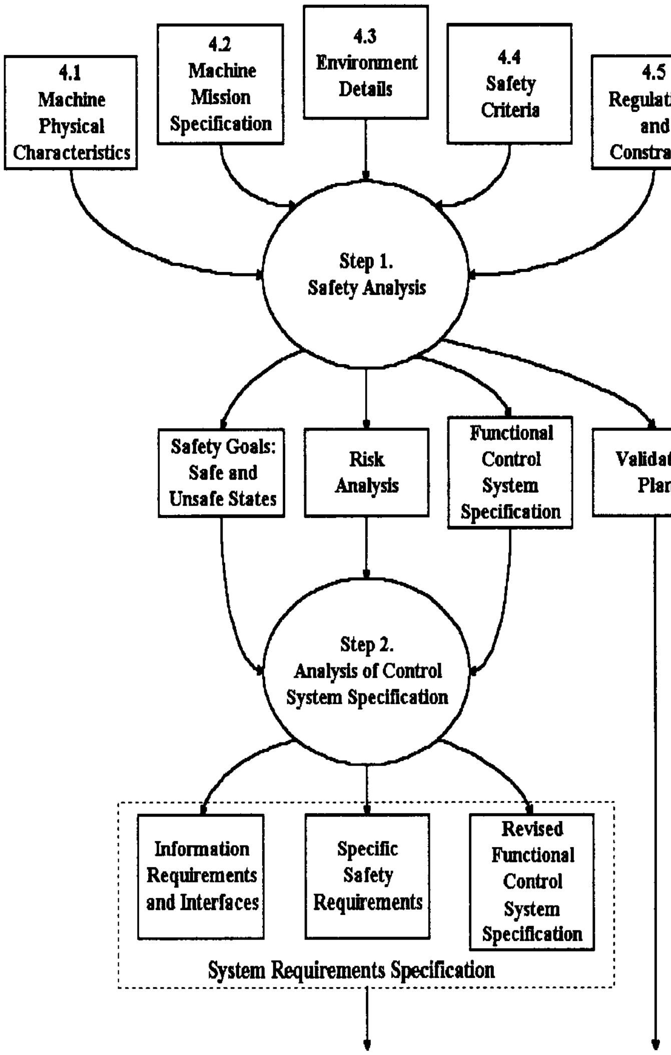
4.1 Machine Physical Characteristics
4.2 Machine Mission Specification
4.3 Environment Details
4.4 Safety Criteria
4.5 Regulations and Constraints
Step 1. Safety Analysis
Safety Goals: Safe and Unsafe States
Risk Analysis
Functional Control System Specification
Validation Plan
Step 2. Analysis of Control System Specification
Information Requirements and Interfaces
Specific Safety Requirements
Revised Functional Control System Specification
System Requirements Specification

Figure 7 Breakdown for requirements specification

This contains such details as the dimensions, power and speed of the proposed robot. Much of this information will be presented in diagrammatic or tabular form.

e.g. The excavator slewing mechanism can apply a torque of up to 30 kNm

5.2. Robot mission specification

This describes the range of tasks that the robot must actually perform. It is essentially the robot requirements specification minus the safety considerations. It is likely to be a substantial document and will contain both verbal high level descriptions of activities, as well as much more detailed information such as data-flow diagrams. If the robot is to handle hazardous materials, they must be clearly defined.

e.g. The excavator must deposit excavated material at the side of the trench by slewing the arm and cab.

The excavator may slew through a full 360° at a rate of up to 1.5 radians per second.

5.3. Environment details

A clear description of the working environment and conditions must be provided. This will contain details of such things as temperature ranges and noise levels. It will also contain information about the proximity of the robot to humans and other objects, particularly objects which can provoke significant secondary hazards such as power cables or pressure vessels.

e.g. The machine operates on a site which has a site boundary fence to prevent access by members of the public, but no physical barrier exists between itself and human workers.

5.4. Safety criteria

This contains the information which will form the basis for decision making concerning safety, reliability and availability.

It includes the required safety performance for the robot in terms of accident probabilities as well as listing requirements for self-test facilities and redundancy.

This data can be both difficult to acquire and have an important influence on the economic viability of the robot.

e.g. The robot must operate in such a manner that it will not cause a higher incidence of accidents than a similar manually operated machine.

The machine must demonstrate an availability of at least 75% in a 24 hour working day.

5.5. Regulations and constraints

Existing legislation concerning mobile robots is rare, despite the large number of organisations developing legislation. There has been a distinct shift away from prescriptive technical structures, and a move towards a more open format for implementation of safety issues throughout the design process. In particular this has been reinforced by the onus placed on designers, manufacturers and suppliers regarding their responsibilities in connection with product liability.

Procedures which relate to safety issues require clear identification of the possible hazards which exist within equipment and the associated risks which are present in its use. The situation is further complicated by conflicting regulations from various European and International organisations, despite much work that has been carried out to harmonise areas of conflict.

e.g. Machinery Directive 91/368/EEC[4]:- The obligations laid down by the essential health and safety requirements apply only when the corresponding hazard exists for the machinery in question when it is used under the conditions foreseen by the manufacturer.

6.0 The VORD requirements tool

Current analysis methods such as the Ward-Mellor approach [Kotonya 1992, 1995] for real-time systems are really software design rather than systems requirements engineering methods. System requirements engineering, particularly where safety considerations are concerned, needs input from multiple perspectives and different engineering disciplines. To support this, we have developed a notion of viewpoints, which represent system stakeholders or sub-systems, and which are used to capture their requirements. To support this we have developed a tool called VORD (Viewpoint-Oriented Requirements Definition) has been developed which covers the requirements engineering process from initial requirements discovery through to detailed system modelling. The tool has been extended to incorporate an explicit safety analysis activity.

The safety analysis process includes the identification of safety considerations, hazard identification, hazard analysis, risk analysis and the derivation, recording and checking of safety requirements. The hazard and risk analysis stages use any appropriate hazard and risk analysis techniques and are not tied to any particular method, however in this case fault-trees are used. VORD automatically computes statistical probabilities of hazards causing incidents.

Conclusions

The safety of powerful mobile robots in unstructured environments is a formidable problem, and some of the conclusions that have been reached so far are:

- Where possible safety related software should be isolated from functional software and given special consideration. An independent safety manager is the ultimate result of this philosophy.
- For reasons of both safety and reliability, the dependence on large numbers of sensors should be minimised. The move towards intelligent sensors which self-check and output high grade information should be encouraged.
- The issue of whether or not to adopt a communications bus such as CAN is a difficult one. Mistakes and faults from very complex wiring layouts are reduced but bus faults themselves become safety critical. Provided such faults can be detected the system can be shut down.
- The definition of safety requirements for new automated products is difficult as they are often being developed in advance of legislation. The emerging solution is to prepare a well-argued safety case to demonstrate that the new system is at least as safe as a comparable conventional manual system.

Acknowledgement

This work is part of the Safe-SAM project which is jointly sponsored by the UK Department of Trade and Industry (DTI) and Engineering and Physical Sciences Research Council (EPSRC) Safety Critical Systems Programme.

References

[Advanced 1992] Advanced Robotics Research Limited (1992), 'Safety and Standards for Advanced Robots - A First Exposition', *Report ARRL.92.009*, 1992.

[Bracewell 1990] R H Bracewell, D A Bradley, R V Chaplin and D W Seward, "Control Systems Design for Robotic Backhoe.", 7th Int. Symp. on Robotics in Construction pp 222 - 229, Bristol, (June 1990).

[IEC 1992] IEC/TC 65A(Secretariat) 123, May 1992, Draft. Functional safety of electrical/ electronic/ programmable electronic systems: Generic Aspects. Part 1: General Requirements.

[Green 1990] P Green, D W Seward and D A Bradley "Knowledge Acquisition for a Robot Excavator.", 7th Int. Symp. on Robotics in Construction pp 351 - 357, Bristol, (June 1990).

[Kotonya 1992] G. Kotonya. and I.Sommerville, “Viewpoints for requirements definition”, IEE/BCS Software Eng. J. 1992. 7(6) pp 375-87

[Kotonya 1995] G. Kotonya. and I. Sommerville, “Requirements Engineering with Viewpoints” IEE/BCS Software Eng. J., 1995. 10(6). To appear November 1995.

[Redmill 1989] Redmill, F.J. (Ed), 1989, Dependability of Critical Computer Systems 2, Elsevier Applied Science.

[Seward 1992] D W Seward, "LUCIE - The autonomous excavator". Industrial Robot International Quarterly Vol 19 No 1 pp 14 - 18, MCB University Press (March 1992).

[Seward 1992] D W Seward, D A Bradley, J E Mann, M R Goodwin, "Controlling an Intelligent Excavator for Autonomous Digging in Difficult Ground". 9th Int.