

System Architectures and Safety for Mobile Construction Robots

D W Seward and S D Quayle

Department of Engineering, Lancaster University, Lancaster LA1 4YR, U.K.

ABSTRACT

This paper describes the most recent work on the development of the autonomous robot excavator - LUCIE 2 at Lancaster. Although the LUCIE project (Lancaster University Computerised Intelligent Excavator) has been running for about five years, in recent months it has undergone a radical change in both system hardware and software. This has come about as a result of research sponsored by the UK Safety Critical Systems Programme, which has highlighted the problems of producing a well argued safety case for intelligent robots in unstructured environments. Other changes have resulted from the automation of the vehicle tracks which means that LUCIE 2 is fully mobile for the first time. Work is also progressing on the problem of linking the robot excavator to project CAD drawings, and this is briefly discussed in the paper.

The new developments discussed in the paper include: the use of three individual ultra-compact PC104 computer systems, communication between computers using CAN bus, the use of a new scanning laser sensor to detect possible collisions, and the use of a satellite global positioning system for excavator positioning and navigation. Recent developments in GPS technology now mean that the vehicle can be reliably positioned to an accuracy of 25 mm.

1. INTRODUCTION

Previous work on the development of LUCIE was reported at previous ISARCs^{1,2}. A JCB 801 mini-tracked excavator has been automated so that it is capable of autonomously digging a good quality trench in highly variable ground. In the past the excavator was positioned manually before excavation could commence and no attempt was made to automate the tracks because of concerns for safety. This concern led to the development of a research programme - "Safe System Architectures for Large Mobile Robots" funded by the U.K. Engineering and Physical Sciences Research Council (EPSRC) under the DTI/EPSRC Safety Critical Systems Programme. This enabled the safety issues to be addressed and LUCIE became the target vehicle to demonstrate the concepts. The earlier version of the vehicle proved the concept of autonomous excavation, but is now regarded as an experimental rapid prototype and has now been extensively re-engineered in a more professional and robust form. This has consisted of a complete revision of the electro-hydraulic system and control architecture. In addition the tracks have now been automated in order to fully test the development of the safety argument.

Previous work showed the value of providing a clean modular separation between low level control of the excavator moving parts and higher level decision making. This distinction has been retained, and the top-level decomposition of the system modules is shown in figure 1. Figure 1 also shows the points where a human would intervene in the system. Defining such points is a valuable technique to arrive at an optimum decomposition. It provides, not only, for genuine human intervention but is also good software engineering practice to ensure minimum "coupling" between modules. It should be noted that the Safety manager is not shown connected to the system. This is because it has a unique status which is described later. Each of these modules will be discussed in more detail later. The hardware implementation of the system is centered around three on-board 486 PC computers. The first takes on the low-level control functions, the second the manager functions and the third is the independent safety manager. This arrangement is shown in figure 2. Ultra-compact PC104 cards have been used. These are described in more detail in another paper presented to this symposium³.

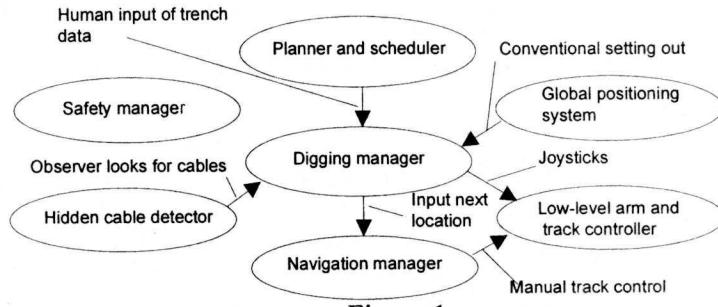


Figure 1
Top-level system decomposition

The hardware implementation of the system is centered around three on-board 486 PC computers. The first takes on the low-level control functions, the second the manager functions and the third is the independent safety manager. This arrangement is shown in figure 2. Ultra-compact PC104 cards have been used. These are described in more detail in another paper presented to this symposium³.

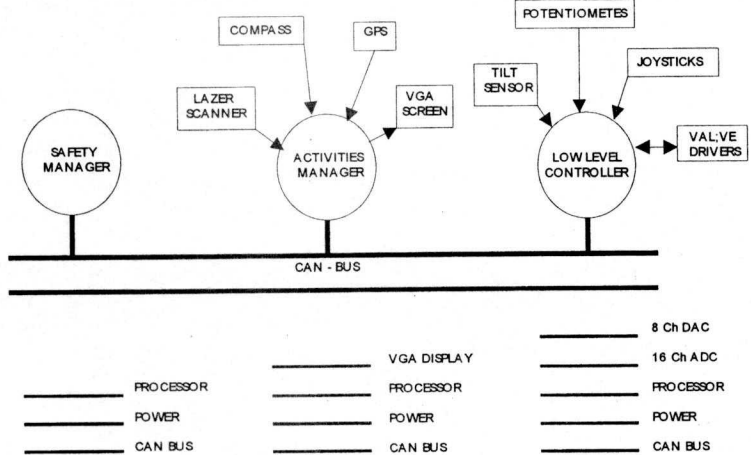


Figure 2
System architecture
plus contents of the three PC104)4 stacks

reduce the very complex wiring looms to just two signal wires and a power wire. The full CAN bus protocol is very fast with transfer rates of up to 1 MBaud, and has many built-in safety features such as:

- Up to fifteen different priorities for messages
- Error messages returned if a problem arises
- Continues to function (with some degradation of performance) with only one signal wire
- Message filtering so that processors do not waste time reading irrelevant messages

Currently only inter-PC communication is via CAN bus but it is the long-term intention to gradually move all sensors and actuators onto the bus. CAN microcontrollers can be used as nodes to feed clusters of sensors or actuators, but the technology is not trivial and there is an extensive learning curve and the need for software CAN analysis tools. Items such as rotary encoders are now coming on the market with embedded CAN bus interfaces.

4. SAFETY MANAGER

Construction robots 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 behavior in unstructured environments

In addition, the behavior 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 behavior 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 above factors produce conflict with accepted practice in safety critical systems design where the emphasis is on deterministic behavior and extensive testing.

This conflict is being addressed by means of a software **safety manager**. This is conceived as an independent distinct entity (like a robot conscience!), whose job it is to monitor the environment, and give permission for all behavior which could have a safety critical component. This is a **behaviorist approach** in that it is concerned with achieving safe behavior, but is not concerned with the processes that determine functional behavior. Clearly, in the interests of efficiency and reliability, the processes that control functional behavior should be rigorously designed using the best software engineering practices to maximise safe behavior. Ultimately the safety manager is, however, responsible, and will block all actions that might create a hazard. In an ideal world the safety manager would have a completely independent set of sensors with which to monitor the robot's working envelope, however this is probably not economically realistic for most construction robots. The LUCIE safety manager will therefore have no exclusive sensors but will have access to all relevant data over the CAN bus.

4.1 Rotoscan sensor

The principal sensor available for the detection of objects within the working envelope is a sophisticated scanning laser known as a Leuze Rotoscan RS3. It 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. See figure 4.

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

- (1) 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 2 (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.

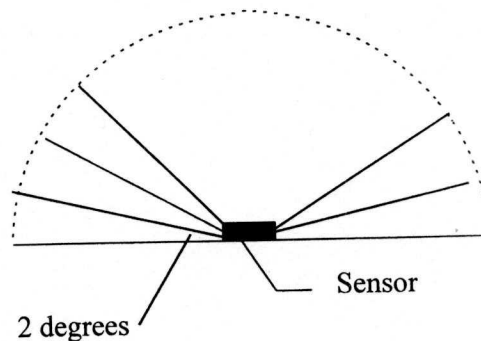


Figure 4
The Leuze Rotoscan sensor

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.

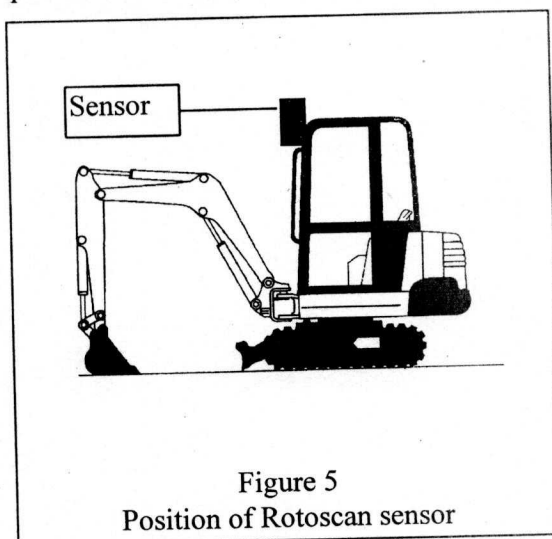


Figure 5
Position of Rotoscan sensor

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

5. GLOBAL POSITIONING SYSTEM AND NAVIGATION MANAGER

LUCIE is fitted with a Trimble 7400 Series satellite global positioning system (GPS) which provides centimetre-level accuracy in real-time (5 Hz update rate). The principles of accurate GPS are briefly explained in another

paper to this conference.³ Work is currently underway to design a navigation manager. The aims of this have been kept simple - "To move the vehicle to a designated Cartesian point and orientation". It is assumed that a higher level program will be concerned with route planning. The Rotoscan sensor will be operational during travel to prevent object collision.

6. LOW-LEVEL CONTROLLER

The development of the low-level controller for LUCIE 2 has so far been confined to that of arm operation only. The PC104 stack for the low level controller consists of:

- Cyrix 486slc2 33 MHz processor with 80387sx floating point coprocessor.
- 8 channel DAC card with 24TTL digital I/O lines.
- 16 channel ADC card with 16 TTL digital I/O lines.

- VGA display card.
- Power Card.
- CAN Bus card.

Software has been written in Turbo C, and allows various modes of arm operation. A software structure has been created containing all information relevant to the excavator itself (arm lengths, valve deadband settings, joint angles etc.) thus acting as a "Black Board" of real time information. Once CAN Bus has been fully installed the need for a majority of this information (current pot values etc.) will become redundant as each stack will have direct access to sensors (providing it has the authorisation) thus releasing reliance on the low level controller to provide the data, and reducing the likelihood of all three modules receiving corrupt data.

Current modes of operation are:

- Direct Mode - Accepts joystick inputs and maps the output directly to the proportional control valves.
- Soft Stop Mode - As direct mode but monitors the arm position via the analogue pots mounted on the arm joints. These pots are read twenty times each program cycle by the ADC and averaged. If the pot value exceeds a preset limit then the output to the valve is proportionally reduced thus slowing the movement of the arm as it approaches its mechanical limit prolonging the life of the equipment and nerves of the operator!
- XY Mode - A single joystick movement is used to control the bucket pin in an orthogonal manner moving the arm in either a horizontal or vertical motion keeping the bucket angle, relative to the ground constant. This approach differs slightly from that previously described⁴ as the equations are simplified for control of a two link mechanism rather than three links as was previously the case.

6.1 Arm Kinematics

Forward transformation:

$$X = L_1 \cos A_1 + L_2 \cos (A_1 + A_2)$$

$$Y = L_1 \sin A_1 + L_2 \sin (A_1 + A_2)$$

Reverse transformation:

$$\cos A_2 = \frac{(X^2 + Y^2 + L_1^2 + L_2^2)}{2 L_1 L_2} * -1$$

$$\tan A_1 = \frac{[Y(L_1 + L_2 \cos A_2) - X(L_2 \sin A_2)]}{[X(L_1 + L_2 \cos A_2) + Y(L_2 \sin A_2)]}$$

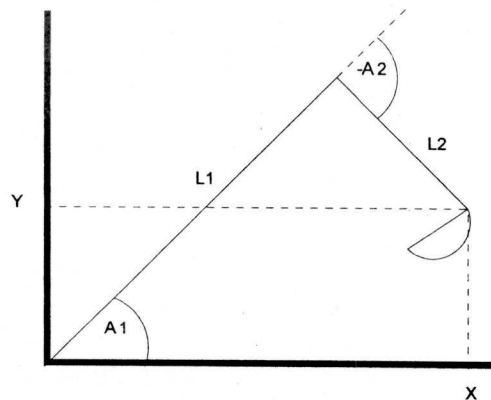


Figure 6
Arm kinematics

XY motion is achieved by creating a target point offset from the current bucket pin position. (The precise distance offset depends on the demand input either from joystick or higher level program). These new XY target coordinates are then put through the above reverse transformation equation to create target angles. The error between the current angles and target angles forms the basis for the proportional plus integral (PI) controller to develop the required valve output for each joint. By monitoring the angle errors it is possible for the controller to recognise such things as underground obstructions or similar thus making higher level automatic digging routines possible.

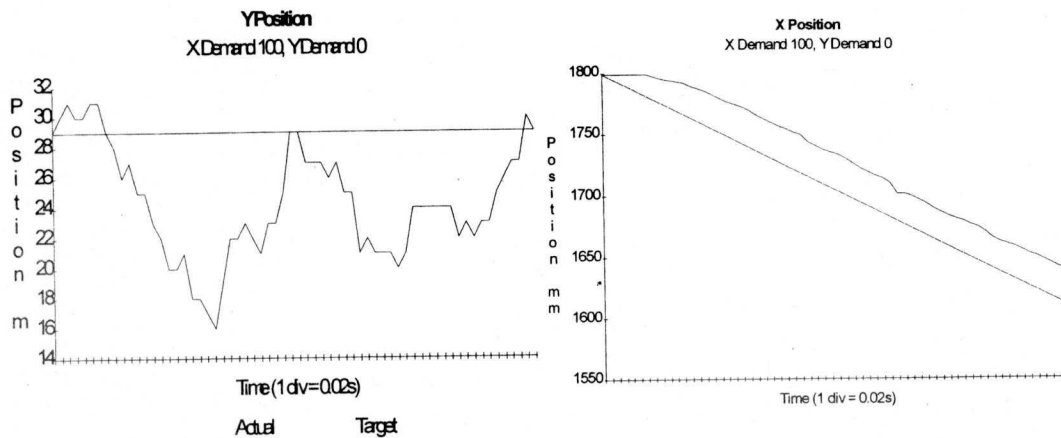


Figure 7
Accuracy of movement in X and Y directions

The PI controller values have been determined by studying the open loop characteristics of the system and relating this to relevant theory. Figure 7 shows the accuracy of movement of the bucket pin in both X and Y directions. It should be noted that the Boom response to the input is considerably slower. This is mainly due to the extra mass which has to be moved along with varying valve characteristics.

The bucket angle is kept at a constant angle to the ground simply by keeping the sum of the three angles equal to a constant, hence if the boom and dipper angles both change by 0.1 rad then the bucket angle must change by -0.2 rad.

A simple "automatic excavation" routine has also been written to write the name LUCIE on a whiteboard with a spring loaded dry wipe pen attached to the bucket. This has proved to work well at low speeds keeping the bucket within the 25mm tolerance, however once the speed is increased the normally legible writing becomes a "doctors scrawl". At present each joint is controlled relative to its own angle error and ignores the performance of the other angle. This can be a problem because the Boom is slower to respond than the dipper, and if digging it is possible that one joint may get temporarily jammed or lack sufficient hydraulic pressure to move at the required angular velocity, thus the straight line turns into an arc until the slow arm catches up with its target angle. To overcome this the two target angles need to be tied together so that one cannot exceed the other beyond an acceptable limit. (This has yet to be tried and tested but if successful should lead to an interesting demonstration with a 1.4 tonne excavator writing as smoothly and quickly as if it were a large pen plotter!)

7.0 DIGGING MANAGER

The language used to implement the Activities Manager was Ada. Programs written using high level languages are both easier to understand and maintain. They also provide a modular approach, information hiding and meaning-full data structures, which are all necessary for a maintainable system.

The Activities Manager involves a real-time component which was one of the critical design issues. A trade off exists between, on the one hand maintainability and flexibility, and on the other hand real-time performance. The strict real-time option in Ada is called **tasking**, but this was rejected for three main reasons:

- Firstly the complexity of tasking makes it unsuitable for a flexible, rapid prototyping process, although it may be appropriate for the final production version where performance needs optimising.
- Secondly the future developers of the Activities Manager could potentially be from an engineering rather than a computing background, and it is unrealistic to expect them to be familiar with tasking.
- Thirdly, the tasks run concurrently so it would be very hard to analyse how the Activities Manager had made its decisions. This makes a safety argument difficult to develop.

For these reasons the strict real-time option was not selected. Instead, simplicity, efficiency and a very fast processor were used to provide an 'as fast as possible' solution. This solution does not compromise the high flexibility and maintainability characteristics of the system.

The next problem was how to embed knowledge in the Activities Manager. A well known technique is to use a **production system**. This consists of three components:

- a list of rules of the form - IF (condition true) THEN (perform action),
- a working memory containing the current value of all system data,
- an inference engine which cycles through the rules, checks the conditions and instigates actions.

With complex systems this can result in having to check hundreds of rules at each cycle, and this can be a penalty in real-time systems. The solution was to place a separate production system in each of the states. In order to maintain performance, the production systems were kept small and efficient. If a production system became too big, the state was split into two or more sub-states. If the most important and most likely rules are given the opportunity to fire first, then each iteration of the production system rules is performance optimised. The addition, modification and deletion of the rule is simple and therefore provides a flexible and maintainable knowledge base.

The production systems need data in order to work. The data structures required must be flexible and meaningful, in order to be maintainable. The data structures provided by high level languages such as Ada are ideal for this purpose, as they provide flexibility, information hiding, logical structuring and meaningful naming.

Another feature of the design platform was the removal of all constants from the source code. Constants are inflexible and were therefore replaced by variables which can be changed, as required by the prototyping process, without having to re-compile the source code. The 'variable constants' can be changed by using a user interface, to edit external files. These

external files contain the values that the 'variable constants' should have. They are read by the Activities Manager and placed into internal data structures. This method provides a certain amount of flexibility without having to re-compile the source code, which is a great advantage in a rapid prototyping process.

8. PLANNER/SCHEDULER AND FURTHER WORK

A key element in making construction robots acceptable to industry is their proper integration into the design/build process. This requires a seamless interface between the CAD drawings and the robotic operation. Work has started on the post-processing of AutoCAD .DXF files to convert them into digging instructions for LUCIE 2. The production of a series of X, Y & Z coordinates representing the ends of foundation trenches, for example, is a fairly trivial task. The real challenge lies in converting these into meaningful digging instructions. It is necessary to introduce expert-system type rules, based on practical site observations. For example a skilled driver never digs himself into a corner, so that the excavator has to be driven across a previously dug trench to escape. Clearly planning is not a real-time activity, so consideration is being given to genetic algorithms or some other optimisation method to determine the best order of events.

REFERENCES

1. 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).
2. D W Seward, "LUCIE - The autonomous excavator". Industrial Robot International Quarterly Vol 19 No 1 pp 14 - 18, MCB University Press (March 1992).
3. D W Seward, J R Ward, J D Findlay and H Kinniburgh, "The Automation of Piling Rig Positioning using Satellite GPS", 13th Int. Symp. on Robotics in Construction, Tokyo, (June 1996).
4. D Seward, D Bradley and R Bracewell "The Development of research models for automatic excavation.", 5th Int. Symp. on Robotics in Construction, Tokyo, Vol 2, pp 703 - 708 (June 1988).