UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2000

BEng Honours Degree in Computing Part III
BEng Honours Degree in Information Systems Engineering Part III
MEng Honours Degree in Information Systems Engineering Part III
MEng Honours Degrees in Computing Part IV
MSc in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute

PAPER C333=I3.25

ROBOTICS

Wednesday 3 May 2000, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions

1 a i) A variety of locomotion systems can be used for mobile robots. Legs, Tracks and Wheels being the most common. For each of these three, outline the advantages and disadvantages, in terms of hardware and software control, associated with using such locomotion systems on a small, autonomous robot.

ii) Wheels are the most common form of locomotion for robots and there are a variety of wheel arrangements. Differential, Synchro and Ackerman arrangements being the most common. Describe these three wheel arrangements and illustrate the associated control problems (both hardware and software) when they are used to drive a robot that has a sense of orientation as well as position.

iii) Chassis shape can also impact on the ability of a robot to negotiate corridors and small openings. Illustrate and describe an environment that highlights the problems of a rectangular chassis and note how the choice of wheel arrangements can impact on the software control.

b A certain behaviour for an autonomous robot is required. In general, the robot should move forward in a precise, straight line. On impact with an object, the robot must backup in such a way as to have some chance of avoiding the obstacle in the future. Having backed up, the robot should continue in a straight line again. This behaviour continues until the robot reaches a goal position. This goal position is indicated by a bright light.

i) Construct a Finite State Machine (FSM) that captures this behaviour.

ii) Outline a mixed analog and digital control system that would exhibit this behaviour. You may introduce any common sensor, timer and control blocks that would be realistically available.

iii) If the behaviour of the robot in part [ii] were to be altered such that it always veered toward the light rather than heading in a straight line, how would you alter your block diagram?

The two parts carry, respectively, 40% and 60% of the marks.

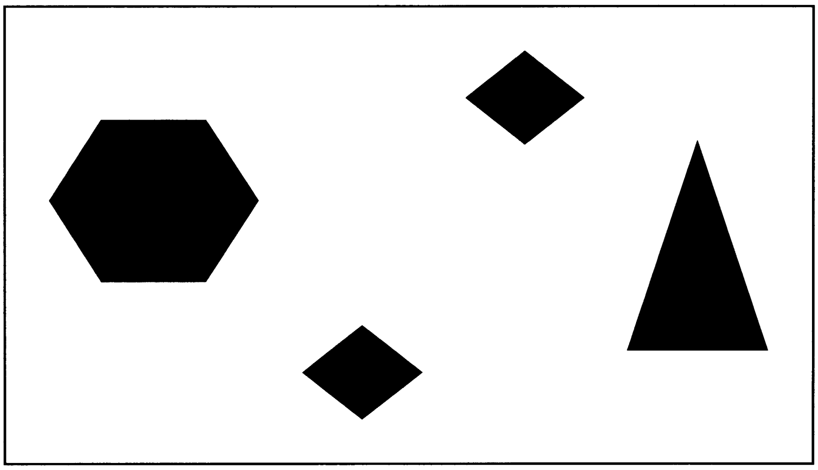
2 a i) For three significantly different types of sensor describe why the information gained from them may be incomplete, inaccurate or worse, ambiguous.

ii) How does sensor fusion attempt to overcome the potential conflict between the information gathered from two types of sensor?

iii) How does the use of a subsumption architecture side-step the issue of conflicting sensory information?

b Subsumption architectures rely on the ability of one behaviour to override another and will usually involve some form of prioritisation of those behaviours. A small robot uses a multi-processor distributed control architecture to implement a different behaviour in each of its processors. The processors are linked by the industry standard I^2C communication protocol, which is essentially priority free (except for the distinction between Master and Slave). Clearly explain how the I^2C protocol can be extended to allow a processor representing a dominant behaviour to gain priority control of the I^2C bus.

c i) Show an exact cell decomposition of the following diagram



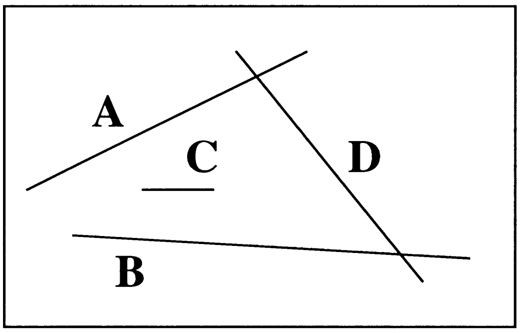
ii) Use this decomposition to construct a non-directed graph representing the possible paths for a mobile robot

The three parts carry, respectively, 30%, 40% and 30% of the marks.

3 a A common problem in motion planning is computing the intersections of lines. These lines may represent the edges of obstacles in a robot's domain or the paths along which a freeway has been computed.

i) If the intersections are computed by comparing every line segment with every other and the intersections thus calculated are then sorted, calculate the order of the processing time in terms of the number of line segments (n) and the number of actual intersections computed (c).

ii) A more efficient algorithm when c < n is the sweep line algorithm. The diagram below illustrates the paths a robot may travel along based upon a computation of the robot's freeways. Show the sequence of active line segments and events that would be generated by the sweep line algorithm if it were run against this diagram.



b i) Briefly describe the potential fields method for describing a two dimensional space in which a robot is to move. This space includes both goal locations it should reach as well as obstacles to be avoided. Illustrate your example with a sketch of the contours of a potential field for a space that includes one obstacle and one goal location.

ii) How might a computed potential field for an environment, of which the agent has a model, be used for path planning? How might a robot cope with obstacles that it encounters, which were not in its original world model, when following a planned path?

c i) Describe how the different primitive behaviours of a purely reactive robot, that has no prior model of its environment, can make use of the key idea of a potential field to define a motor reaction function appropriate to each behaviour.

ii) Illustrate this by defining a suitable reaction function for obstacle avoidance.

iii) How might the values returned by the different primitive behaviour reaction functions be combined?

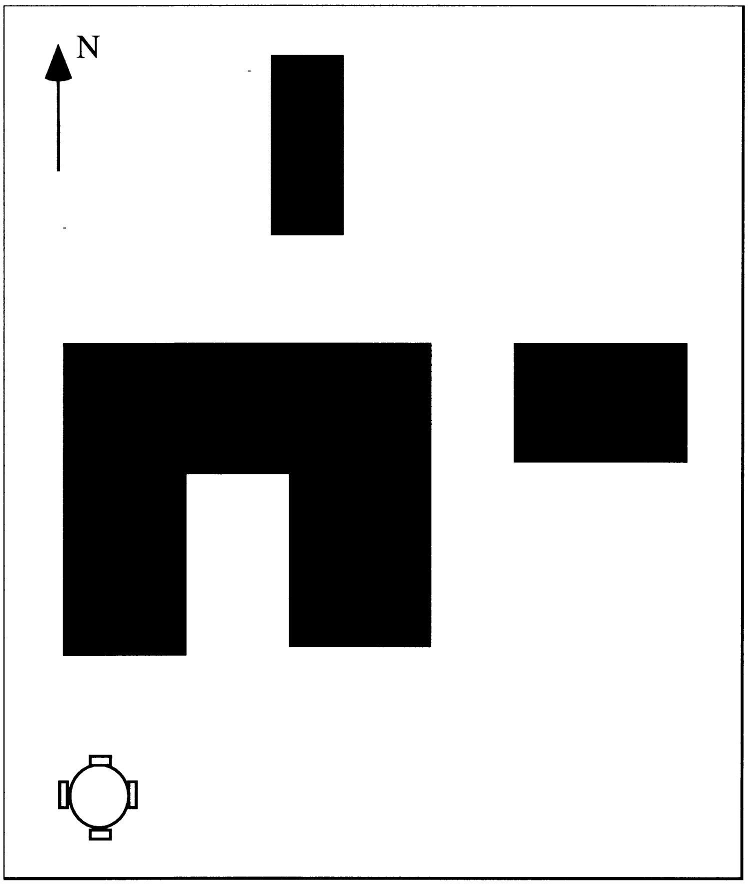
The three parts carry, respectively, 40%, 30% and 30% of the marks.

4 a i) What is a Teleo-Reactive (T-R) program?

ii) What does it mean to say that a T-R program is complete?

iii) Indicate how a T-R rule sequence might be implemented electronically.

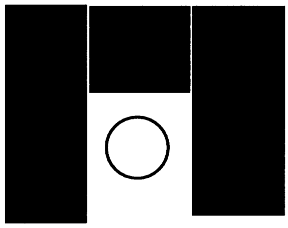
b Consider a small simulated robot walking over a two dimensional space containing straight sided obstructions as depicted below. The sides of the objects always run north-south or east-west.



The robot is round, and has four fixed infra-red sensors equally placed around its circumference. These can detect when an object is within one robot-diameter of the robot in the direction in which they are pointing.

Assume that the robot has `move_forward`, `move_left`, `move_right`, or `move_back` persistent actions that always move the robot in a straight line and that it has test routines, `object_in_front`, `object_behind`, `object_left`, `object_right` that immediately return true when the sensor pointing in that direction detects an object. The robot always points north.

i) Write a T-R program that will enable the robot to wander around its two dimensional space avoiding obstacles. Write it such that the robot prefers to move forward, left or right and only moves backwards if it gets into a cul-de-sac, that is it enters a space such as:



where its front, left and right sensors all indicate a obstacle. In this situation, the robot should move backwards until it does not detect an object on the right or the left. At that point it should move left or right - depending upon which direction is clear, and revert to its default behaviour of trying to move forward. [Hint: Use a state variable to remember when the robot is moving backwards.]

ii) Now assume that the robot also has an action `rotate(D)`, where D is `clockwise` or `anticlockwise`, which will change the orientation of the robot by rotating it 90 degrees either clockwise or anticlockwise. Also assume it has a clock which sends out a signal at regular intervals. It can detect when the signal is sent by a test, `clock_signal`.

Rewrite the T-R program given in answer to [bi] so that the robot gets out of cul-de-sacs but does not need to reverse. Also, give the robot some random behaviour by having it do some special action when the clock signal is detected and its front, left and right sensors do not detect an obstacle.

The two parts carry, respectively, 40% and 60% of the marks.

