25

The Mark 1.5 Edinburgh Robot Facility

H. G. Barrow and G. F. Crawford
Department of Machine Intelligence
University of Edinburgh

INTRODUCTION

In May 1971 the Mark 1.5 Edinburgh robot system went on-line as a complete hand-eye system. Two years earlier the Mark 1 device had been connected to the ICL 4130 computer of the Department of Machine Intelligence and Perception. The Mark 1 (Barrow and Salter 1970) had been little more than a semi-mobile T.V. camera, with coarse picture sampling (64 × 64 points, 16 levels), a limited range of movement over a three-foot diameter circular platform, and a pair of touch-sensitive bumpers. Within eighteen months we had developed suitable basic software, and a 'teachable' program capable of recognizing irregular objects via the T.V. camera (Barrow and Popplestone 1971). However, the restrictions upon movement, the limited range of actions which could modify the 'world', and the shortcomings of the video system, made more advanced work difficult. Plans were therefore laid for the construction of the Mark 2 device.

The Mark 2 robot system will possess moderately sophisticated eyes and a hand which can manipulate objects with a reasonable degree of precision. At the time of writing (May 1972) we have two fixed T.V. cameras and a hand with 6 degrees of freedom; the Mark 2 is intended to possess steerable, controllable T.V. cameras and a hand with 7 degrees of freedom and tactile feedback. The present equipment thus represents a useable system, not yet up to full Mark 2 specification, but considerably more useful than the Mark 1.

DESIGN CONCEPTS

It is important that the complete system should be as self-reliant as possible. If it depends much upon human assistance to pre-process information or to put things right when they go astray, it is all too easy in one's research to avoid the central issues of a problem, and produce a 'solution' which does not survive when confronted by real situations. We therefore wish that the hardware should not impose unreal restrictions upon our research: it should permit the system to extricate itself from as many predicaments as is reasonably possible.

We had a choice of a number of system configurations: a static hand-eye system, as the Stanford System (McCarthy, Earnest, Reddy, and Vicens 1968); a freely-moving robot, as the SRI device (Nilsson 1969); or even a distributed system with many sensors and effectors only some of which have human counterparts, as HAL 9000 (Clarke 1968).

In the Mark 1 device we implemented a suggestion from Derek Healy that when a robot is complicated and linked to a fixed computer, and its world is simple, it is better to keep the robot still and move the world. From its own point of view, the robot cannot tell whether it or the world moves, and one can simulate free movement in a restricted area. In principle the Mark 1 robot had three degrees of freedom over its plane world (two translations and one rotation). In practice it was limited by the inertia of the platform and the manner in which it was supported.

For the Mark 2, we chose a world platform about 2 metres square with two translational degrees of freedom, driven by servo-systems. Over the platform is a bridge from which are suspended a hand and an eye. Thus the system is like a man at a workbench, who can lean over it and move sideways, but not walk round the other side: the device is free-ranging but always constrained to face in the same direction.

If an object is dropped or knocked over, the hand must be able to restore it to its correct position and orientation. In general, this demands three translational degrees of freedom, three rotational ones, and grasp. If the object has suitable stable states when resting on the platform (for example, a brick), two-stage manipulation can be used to re-orient it with only two degrees of rotational freedom. Because the platform is moveable, two degrees of freedom are detached from the hand; only vertical motion, grasp and two rotations remain.

It is important that the hand possesses suitable tactile sensory equipment. It is valuable to know the force applied to the grasped object, if only to know whether it is grasped at all. Inevitably, errors of interpretation and guidance will result in attempts to push the hand or its contents through objects; such attempts must be detected to be corrected. Fitting objects together ultimately depends upon tactile feedback: the locating surfaces may be hidden from view and previous inspection and dead-reckoning are not sufficiently accurate.

The main sense for the system was chosen to be vision. From a practical point of view, television offers good resolution and there is a well-developed field of technology. We thus selected the T.V. camera as the principal sensory organ for our robot system.

The system possesses a large number of elements to which we wish to give commands or from which we wish to obtain data. There are also many occasions upon which it is desirable to give sequences of commands to two or more elements in conjunction, for example moving the platform in a straight line, in a particular direction, or closing the hand until resistance is felt. It was therefore decided that all sensors and effectors should be connected directly to a small satellite computer linked to the main time-sharing machine, and a Honeywell 316 was chosen for this purpose. The satellite is responsible for tactical control of the robot system, and it can perform certain low-level sensory analysis relieving the main machine of part of its burden. It is also a dedicated machine using machine-code programs which can run faster than their high-level language counterparts on the time-shared computer. Testing and installation of additional equipment can be performed using only the satellite.

The robot system is available as a standard peripheral of our Multi-POP time-sharing system, and is available to the user at any console. Several packages exist in the Program Library for using various parts of the robot hardware. There is an executive program which gives the user control over the satellite from a time-sharing console, as well as programs for driving the platform and hands, and for reading pictures from the T.V. camera.

THE PERIPHERALS

Figure 1 shows the configuration of the business end of the robot. There is a platform, or 'world', which moves bodily in two directions, and a bridge over it. From the bridge is suspended the 'hand' over the centre of the platform's area of action. At one end of the bridge is a cage which supports the oblique T.V. camera 'eye'. The overhead camera (not shown in this photograph) is normally suspended from the bridge.

Platform

The platform is about 2 m square, and made of light and rigid sandwich board. It is fixed to a carriage which has two degrees of freedom, each driven by a wire drive from a D.C. position servo-system. One motor drives it along the axis of the bridge, the other at right angles to this. The action resembles that of a giant X-Y plotter: the two motions are independent and may be easily programmed to reproduce any trajectory.

Because the position sensing is performed at one end of a drive wire and the load is at the other, there is inevitably some bounce when the table is moved. The overshoot is only a fraction of a centimetre and the settling time is very short. It causes us no problems and could be eliminated by improving frictional damping.

Positional accuracy is determined mainly by the accuracy of the sensing potentiometer ($\simeq$0.5%). When the command signal comes from a 12-bit D to A converter, positional increments are about 0.4 mm. When we measured repeatability, we found it to be better than 25 μm.

The speed of movement of the platform is approximately 250 mm per second in each direction, taking about 5 seconds to travel between extremes.

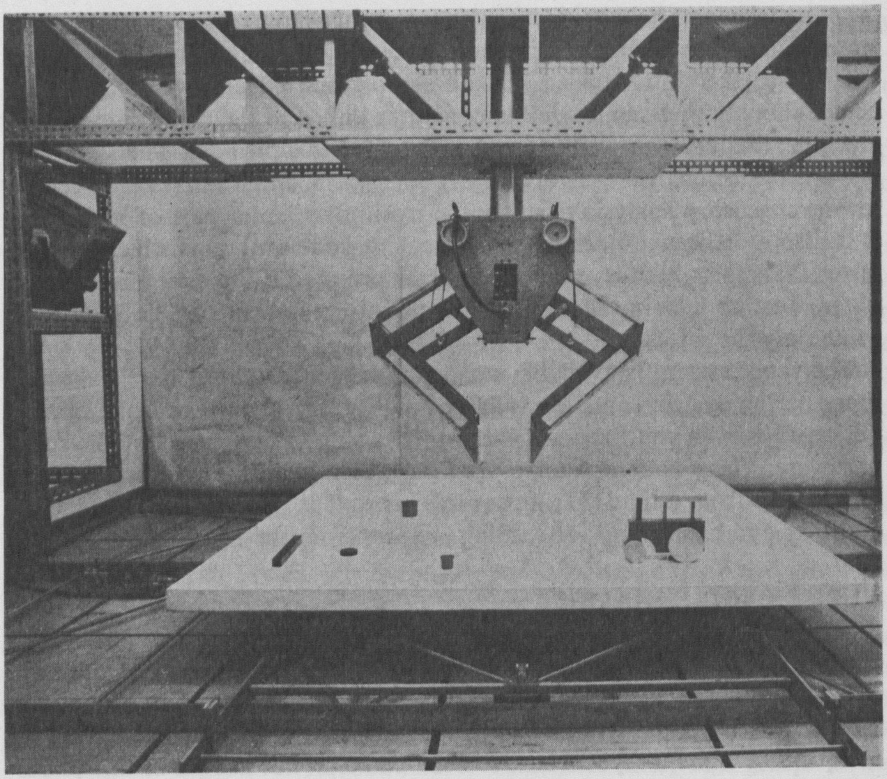

Figure 1

Its acceleration is of the order of 0.1 g so there are few problems of stability of objects piled on it.

Hand

The 'hand' resembles a pair of hands and arms. There are two vertical, parallel plates, or 'palms' which can be driven towards each other to grip an object, and raised together to lift it. Each palm is suspended below the motor housing by a linkage which permits vertical and horizontal movements to remain linear and independent, and which maintains the palm in a fixed orientation throughout its motion. There is a D.C. position servo-system for the horizontal movement of each palm, but one servo drives both palms vertically. The palms can be driven horizontally in opposite directions to grip or release an object, or in the same direction to shift it sideways.

The whole hand assembly is suspended from a column which can rotate about a vertical axis through nearly two turns, driven by a wire drive from a D.C. servo-system. Objects can thus be turned round, as well as lifted.

The forces which can be exerted are quite large, up to 20 kg, both gripping and lifting. We have therefore fitted controllable force limiting circuits to reduce this to about 5 kg. The palms can grasp objects up to 370 mm across, and be raised to a height of 270 mm. Corresponding positional increments are therefore 0.4 mm and 0.3 mm for 10-bit D to A converters. Rotational range is ±6 radians with increments of 0.006 rad. (about 0.7 degrees). Each movement takes about 5 seconds to traverse the full extent. The palm surfaces are about 85 mm by 140 mm and are covered with a layer of foam plastic. Objects only 2 mm in diameter have been picked up from the platform under computer control.

T.V. cameras

The oblique camera of the system is mounted in a frame hung below one end of the bridge. It is fixed in orientation and looks along the line of the bridge at the platform about 250 mm in front of the point directly under the hands. It is a commercial closed circuit T.V. camera with a 1″ *f*2.8 lens and the normal C.C.T.V. characteristics of 625 lines, 50 frames/sec. The camera can see an area of platform roughly 1 m deep by 0.5 m broad, that is, only a part of the whole.

A second camera is mounted near the middle of the bridge and looks, via a mirror, vertically downward. It sees an area of platform about 300 mm × 400 mm, and can be used for detailed examination of objects from its special viewpoint. It can also be used in conjunction with a projector which casts a stripe of light via a second mirror obliquely on to the platform. From the appearance of the stripe, the 3-dimensional properties of objects can be measured (Popplestone 1971).

A computer-controlled reed relay switch enables either camera to be selected and connected to the video sampler unit.

Video Sampler

The device which digitizes T.V. picture information is very simple. The computer sets the co-ordinates of a point in the picture; about 750 × 290 sample points can usefully be used. When the T.V. scan reaches that point, the video voltage waveform is sampled and digitized. The resulting number is read by the computer: it ranges from 0 (black) to 255 on a logarithmic scale.

The Z (vertical) position is found by counting the T.V. lines, and the X (horizontal) position by counting ticks of a clock. A normal T.V. picture is transmitted by sending all the odd lines in one field, and then all the even lines in the next. The two fields together comprise one frame of 625 lines and 25 frames are transmitted each second. We have not yet bothered to discriminate between odd and even fields, so the vertical resolution is limited to only 312 lines (of which some are outside the actual picture area). We normally consider only every other point across the picture, yielding a matrix of approximately 400 × 300 points. Because the aspect ratio of the T.V. picture is 4:3, they are distributed in a square array. It should be noted that the resolution of a good commercial camera is limited by its optics and electronics to about 400 lines.

When the T.V. scan reaches the selected point in the picture, the average amplitude of the video waveform over a 90 nano-second window is digitized on a logarithmic scale. Absolute light level is not important, but relationships between light levels are. Moreover, differences of brightness are affected by many variables, whereas ratios of brightness depend primarily upon the properties of the surfaces concerned, and less on their environment. Experience with the Mark 1, which had linear encoding, showed it had a paucity of discriminable levels in dark areas, and a surfeit in light areas. Logarithmic encoding was thus an attractive proposition.

The digitizing unit must be primed by the computer. When the scan reaches the selected point, the video signal amplitude is digitized and held in a register until the computer reads it. The action of reading also reprimes the sampler.

Consideration of the mode of use of the sampler, and the limitations of the computer to which it is connected, led us to conclude that the best compromise was to take only one sample per T.V. line. Accordingly, digitization is permitted to take up to 40 μs. The satellite computer can prime the sampler sufficiently rapidly to keep up with the T.V. scan, thus taking a complete column of samples each frame. To read 64 columns takes 1.28 seconds, too slow for some purposes but adequate for many robot activities, which are computer-bound anyway.

Some T.V. cameras have A.C. coupled outputs; it is therefore necessary to restore the waveform's D.C. component before passing it on to the sampler. This is done by a module which clamps the waveform to a reference level when it is known to represent black. Originally, this point was taken to be the 'back porch', just after the line synchronizing pulse, which is 'electronic black'. However, the camera used had automatic level adjustments which modified the 'real' black level. A strip of black felt was therefore stuck over the image plane to provide a strip of real black at one side of the picture, and the circuit adjusted to clamp on this real black level.

The pre-processing module also extracts line and frame synchronizing pulses from the video waveform and emits them as TTL logic levels for driving the X and Z co-ordinate units. From these in turn it receives appropriate pulses to mix with the original video waveform so that a pair of white cross-wires is generated to indicate on a T.V. monitor the picture point currently being sampled.

Other peripherals

In addition to the special hand and eye equipment described above, we have also connected a number of general purpose peripherals to the robot system. These include a Tektronix type 611 storage tube display, a 16-channel, 10-bit A to D converter, and reed relay inputs and outputs.

SATELLITE COMPUTER

Building the robot system round a small satellite computer was an attractive proposition because the satellite can fulfil tactical control functions. It can monitor sensors while performing actions, can co-ordinate several simultaneous movements, or it can perform low level picture processing. Such activities are time-consuming, but demand only simple programs: they are better performed by a small dedicated machine than a large time-sharing computer.

The machine we chose is a Honeywell 316, a 16-bit small computer, with 8K of memory. The decision was predominantly an economic one and, at the time it was made, the market was comparatively restricted. There is now a much wider choice among small computers, many of which have more exciting instruction sets, and are comparatively cheap.

All sensors and effectors of the robot are interfaced to the satellite machine, which is linked to the time-sharing machine. It was intended to give the satellite as much autonomy as possible, and to avoid making the relationship between the two machines heavily one-sided. The link is therefore two independent data paths which can operate simultaneously in opposite directions.

Interfacing

It was fairly clear that most peripheral devices would require some form of data buffering. Computer data transfers are usually synchronous; the information is only presented for a few fleeting micro-seconds. But devices need to transfer asynchronously: D to A converters need to maintain their outputs until they are told to change, and picture samplers must hold their data until the computer is ready to accept it. We also felt that in an experimental situation, it is desirable to keep one's devices as simple and general-purpose as possible, so that they may readily be modified or moved around.

We have based all our peripheral construction (including the link between the computers) upon two units, the input and output buffers. An output buffer contains a 16-bit register, and is connected as a normal peripheral to the I/O bus of the computer, which can transfer single words into its register. It presents a rather simple appearance at a socket into which the external device is connected. An input buffer unit is likewise a 16-bit register connected as a peripheral of the computer, which can read its contents. It too has a socket which presents simple interface conventions to the outside world.

Buffer interface conventions

In designing this interface we drew upon the ideas behind the British Standard Interface (British Standard 4421, 1969). Consider two units, *A* and *B* (figure 2), and a transfer of information between them from *A* to *B*. For example, *A* may be an output buffer and *B* a D to A converter, or *B* may be an input buffer and *A* a video sampler. The transfer takes place as follows:

Unit *A* places the data on the parallel data bus and then sets READY to be TRUE.

Unit *B* reads the data and then sets $\overline{\text{ACCEPTED}}$ to be FALSE

Unit *A* sets READY to be FALSE and removes the data

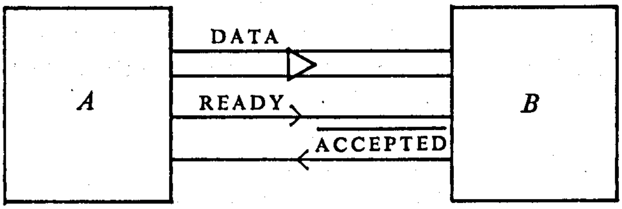

Figure 2

In addition to the READY and $\overline{\text{ACCEPTED}}$ lines, there is one additional RESET line for each unit. The RESET signal sent by *A* to *B* sets *B* into a state in which it is ready to receive data from *A*. The RESET signal from *B* to *A* sets *A* into the state in which it prepares the next word of data. *A* and/or *B* may be reset by external means (the computer or a push-button), and when this occurs a RESET signal is sent to the other unit.

Unlike the B.S. Interface, the data path is 16-bits wide and its signal levels are TTL logic levels: transfers can occur at full TTL speed.

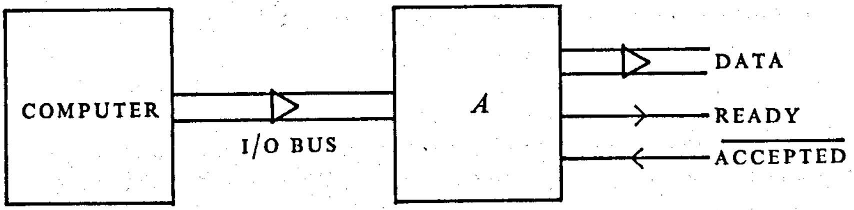

Figure 3

Consider now a single output buffer (figure 3). The unit contains a register to hold data, and a bistable which represents its state, EMPTY or FULL. When it is EMPTY, the unit ignores its mate and responds only to the computer. The computer can transfer a word to *A*'s register, which also sets its state to FULL. *A* then directs its attention to its mate and ignores the computer, which cannot now perform another transfer. READY becomes TRUE and unit *B* can read data from *A*. When $\overline{\text{ACCEPTED}}$ goes FALSE, *A*'s state becomes EMPTY and it can respond to the computer once more.

Consider an input buffer (figure 4). Unit *B* also possesses a register and a state bistable. When *B* is EMPTY it can participate in a transfer with *A*, and ignore the computer. On successful transfer, *B* sets $\overline{\text{ACCEPTED}}$ FALSE and becomes FULL. It now ignores *A* and can be read by the computer. When the computer reads the contents of *B*'s register, *B* becomes EMPTY once more.

At any stage an input or output buffer can be interrogated by the computer to determine whether it is FULL or EMPTY. A FULL input buffer or an

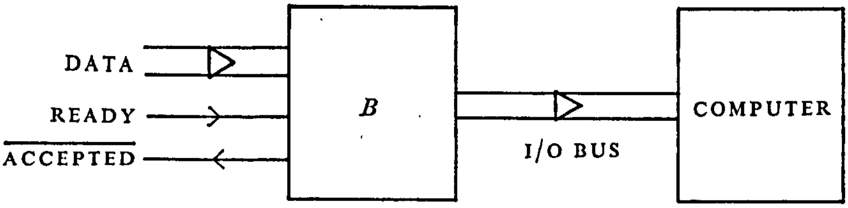

Figure 4

EMPTY output buffer can potentially interrupt the computer. Such interrupts can be permitted or inhibited for each buffer independently. Each buffer can also be reset by the computer to the EMPTY state.

Linking computers

Clearly, in accordance with the above description, any input buffer can be connected to any output buffer. If both are connected to the same computer, little is gained, but if they are connected to different computers, it becomes possible to transfer data from one machine to the other. Two such data paths in opposite directions give fully independent channels of communication.

We linked the H316 and ICL 4130 computers by designing input and output buffers for each, and then connecting them together. In each path are two registers: a specially built path need only contain one register, but it is much simpler to construct a link from standard buffer units.

Devices

Peripheral devices for the robot system have all been designed to conform to the standard interface conventions. They are connected to output and/or input buffers, and not directly to the I/O bus of the computer. This has several practical advantages; we have removed a complex piece of logic from the device, the design of which is simplified; buffer units can be mass-produced on printed circuits to improve reliability; devices and buffers can be readily swapped around to aid fault analysis; devices are machine-independent, only two units (the buffers) need to be re-designed if a different computer is to be used.

Modularity has also been maintained as far as devices are concerned. The same D to A converter module can be used to drive any servo-system, or the visual display, because we have endeavoured to keep any peculiarities confined as closely as possible to the equipment concerned. This philosophy has undoubtedly aided us on many occasions.

TOWARDS MARK 2

As we remarked earlier, in order to be reasonably self-reliant, the robot needs the ability to turn objects over, as well as to turn them round. We are already working upon the addition of such a facility. We intend to equip each palm with a motor-driven plate which will lie parallel to the present palm and will rotate about an axis normal to its surface. The two plates must turn in synchrony, with as little relative movement as possible to avoid twisting the object.

When it attains full Mark 2 specification, the hand will be able to hold most objects in any orientation at any point in its workspace, though it may be necessary to perform a two-stage manipulation to do so. Clearly, however, it will still not possess enough degrees of freedom to apply an arbitrary rotation to the object in its grasp, for example, turning a key in a normal keyhole, or a doorhandle. When we wish to tackle such tasks, we shall be thinking about a Mark 3 device.

The Mark 1.5 hand does not yet possess any tactile sense. At present we use vision to determine whether an object has been successfully picked up, but in cases of delicate manipulation, visual observation is not sufficient and touch is essential.

We intend to add force transducers to each wrist so that we can deduce such information as strength of grip, the weight of objects held, and whether the hand or its contents have collided with something. While being a good deal less sophisticated than that of the human hand, this tactile sensory system will enable us to study a variety of manipulative tasks, like putting rods into holes.

The robot's visual system is to be upgraded also. Despite the reasonably high resolution of the T.V. sampling system, it is ultimately limited by the resolution of the vidicon, and the focal length of the lens. We need a wide-angle lens in order to see a usefully large area of platform, but such a lens does not permit adequately detailed observation of a single object. We need to employ either a lens turret, or a zoom lens, under computer control. When a narrow angle lens is used, its small field of view demands pan and tilt movements of the camera (we cannot always rely on platform movements, for example, when the object is held) as well as focusing.

We need to have tighter control over the internal operation of the camera: automatic control circuits cause shifts of level as well as changes in gain. Tenenbaum (1970) has shown the advantages to be obtained from active control of the functions of the T.V. camera. He also worked on the analysis of colour. We hope to fit a colour wheel.

SOFTWARE

A number of program packages are available in the program library for using the equipment from a time-shared console attached to the main computer. The language used is POP-2 (Burstall, Collins and Popplestone 1971). Core occupancies refer to 24-bit words of ICL 4130 core.

LOADER

The most fundamental program for the H316 satellite is a 16-word loader which resides permanently in a write-protected area of store. It reads 8-bit bytes from the link and assembles them as 16-bit words, representing address or contents alternately. It stores the contents in the corresponding address repetitively until it reads zero as the address, in which case it jumps to the location specified by the following word.

LIB HONEY

This is a POP-2 program which assembles H316 symbolic machine code programs. It permits the user to define macros which have the full power of POP-2. Thus, it is possible to extend or re-define the assembly language at will. Facilities are also provided (when used in conjunction with [LIB LINK EXEC]) for accessing and modifying H316 core locations by symbolic names. (4K of POP-2)

LIB LINK EXEC

This library package provides communication between the satellite and main machines. From a Multi-POP console, the user can control the satellite, send data to it, read core locations, start up programs, check on their progress and debug them. The package consists of a set of POP-2 routines, and a set of H316 machine code routines to handle transfers between the two machines, and carry out the user's requests.

As far as possible, the capabilities of the two machines are equal. In particular, the two may operate autonomously, both being able to initiate requests and each responding to the other. The H316 executive is an interrupt handling program. A message from the ICL 4130 arriving via the link causes an interrupt entry to the executive, which saves the volatile registers on a pushdown stack and then interprets and obeys the message. Such interrupts can be nested. The user can employ the interrupt handling routines for his own purposes.

Multi-POP does not provide the user with interrupt facilities, so the POP-2 link executive routines are not interrupt driven. The user can check whether the H316 is attempting to send a message and if so read it and interpret it. The H316 has the ability to request the call of any POP-2 routine of a specified set (which can be as large as desired). Thus an error routine can be called, instead of the printing routine expected by the user, or flags can be set or cleared to permit linked parallel processing.

Some of the POP-2 routines provided for controlling the satellite are as follows:

HSET();	Initializes the H316 to an idle loop with interrupts enabled and registers cleared.
HREADY();	Interrupts current program to await further commands. (All commands could be given while program is still running.)
HGOON();	Return to interrupted program.
HLOC(N);	Reads contents of H316 location N, or updates contents, for example, PRINT(HLOC(1024)); or 3→HLOC(1000);

HREG(N);	Reads from or writes to a volatile register (like HLOC) N=0 modifier register 1 main accumulator 2 extension accumulator 5 program counter, etc.
HDEV(N);	Reads from or writes to H316 peripheral with device address N.
HAPPLY(ARG1, . . ., ARGN, SUBR, N);	Causes H316 to jump to specified subroutine with the given arguments. (N specifies how many arguments there are.)
HBRK(LOC);	Sets a break point at the specified location. When a breakpoint is encountered, the location and accumulator contents are printed on the POP-2 console, the executive is entered and awaits instruction until HGOON(); is given.
DTOH(FILENAME);	Loads a machine code file into the satellite.
HTOD(N, LOC, FILENAME);	Dumps N locations of satellite core, starting at LOC on to disc under the given file name.

The link executive programs are written to be extensible. It is easy to provide H316 routines called from the ICL 4130, as extra executive facilities. The executive functions give a solid base upon which to build further. (2K of POP-2, 0.5K of H316 core)

LIB ROBOT MOVE

This is a package for driving the motors of the robot. (It is written as an extension of [LIB LINK EXEC]).

The problem it is designed to overcome is that of ensuring smooth movements from point to point when more than one motor is involved, for example, moving the platform in a straight line. If the new destinations were output to several servo channels simultaneously, all the motors would run at full speed, and get to their new positions at different times, giving a peculiar trajectory. An H316 program drives all the motors involved in small steps at the rate of the slowest so that the trajectory generated is a straight line (except for the rotations). At the end of the movement, an appropriate pause is made to allow any mechanical oscillations to die away.

A general-purpose function is provided for setting some outputs instantaneously, and tracking others smoothly. It is usually more convenient to split up movements into lifting, turning, etc., and a number of functions are provided, as shown below.

A co-ordinate system is embedded in the platform with its origin at the centre, X and Y in its plane and Z vertically upwards. Distances are measured in centimetres, angles are measured in radians. The position of the platform (or rather of the robot over the platform) is specified by the location directly under the centre of the hand.

MOVETO(X, Y);	positions the platform point (X, Y, 0) under the hand.	
MOVEBY(DX, DY);	makes an incremental movement (DX, DY, 0).	
RAISETO(Z);	raises (or lowers) the hand to Z cms. above the platform.	
RAISEBY(DZ);	incremental raise.	
GRASPTO(W);	sets width between palms to W cm.	
GRASPBY(DW);	incremental grasp.	
TURNTO(THETA);	rotates hand about vertical axis.	
TURNBY(DTHETA);	incremental turn.	
TILTTO(PHI);	rotates palms.	to be added shortly.
TILTBY(DPHI);	incremental palm rotation.	

Current positions are kept available in global variables XNOW, YNOW, THETANOW, etc. (0.75K of POP-2, 0.12K of H316 core)

LIB PICTURE PACK TWO

The total number of sampleable points (over 200,000) inside the T.V. picture makes it impractical to read and store them all. It is necessary to be able to read a subset of them. As mentioned earlier, the sampling is performed for picture points one by one; the computer must do any necessary scanning. The satellite is therefore responsible for scanning the appropriate subset of picture points (which it can do more efficiently and faster than the main machine) and storing the samples (packed two to each 16-bit word) in its core. The data are then transmitted, *en bloc*, to the main machine, where they are loaded into an array of 8-bit elements.

The library package consists of two parts, the picture taking routines in H316 machine code, written as an extension to the executive, and their POP-2 counterparts. The following POP-2 functions are available:

PICINT(X, Z);	causes a single picture point, (X, Z) to be sampled, and the brightness is returned as the result.
SETPIC(X, Z);	does not cause a sample to be taken, but merely sets the indicating cross wires on the T.V. monitor to point (X, Z). (Useful for diagnostic purposes, etc.)
LOADPIC	takes as its arguments a specification of a rectangular window in the picture, in terms of upper and lower values of X and Z co-ordinates, together with increments in each. For example, X runs from XLO to XHI in steps of XINC. Normally the increments are chosen to give a square array of sampled picture points (that is, XINC:ZINC=2:1). When LOADPIC exits, the global variable RETINA contains the required array of brightness values. There are several other global variables which are also set by LOADPIC, and contain information such as the position of the T.V. camera when the picture was taken.

DISPLAY	is a general purpose routine for printing pictures. It is given a function to be displayed (for example, an array), the ranges of X and Z over which its values are to be printed and a specification of which characters are to be printed for the various values, overprinting if desired (2.8K of POP-2, 0·2K of H316 core).

LIB PERSPECTIVE

There are four co-ordinate systems associated with the robot (figure 5):

Absolute – a 3-D Cartesian system fixed to the platform.
Relative – a 3-D Cartesian system fixed to the robot.
Picture – a 2-D Cartesian system, specifying a picture point.
Retina – the 2-D system defined by the indices of the array RETINA.

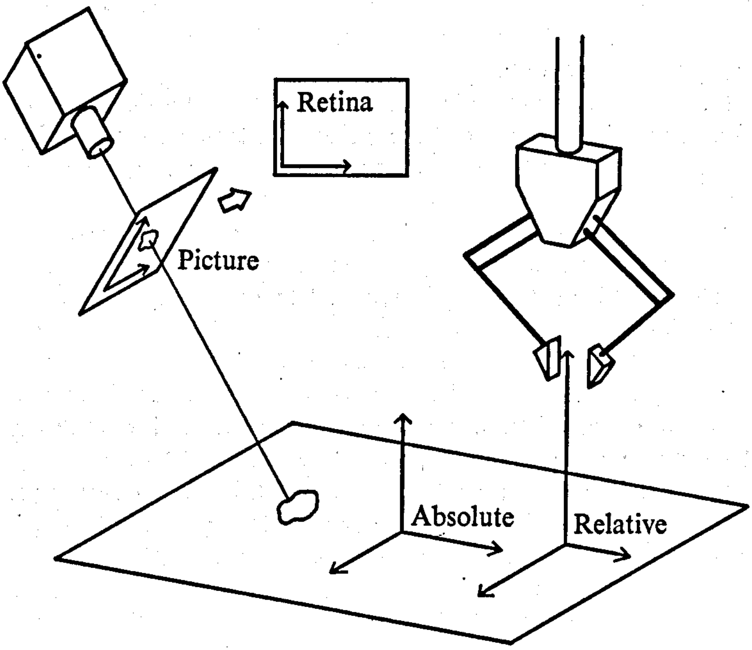

Figure 5

Clearly, we wish to be able to relate a point of the array to points in the absolute system, to determine where objects are from an analysis of the picture. This package provides a set of routines for conversions between systems. Some of the available functions are:

ABSTOREL	RELTOABS
RELTOPIC	PICTOREL
PICTORET	RETTOPIC

One point needs explaining. Absolute and Relative positions are represented by triples, but picture and retina points are duples. RELTOPIC and PICTOREL relate triples to duples and vice versa. These transformations are performed by the well-known method of matrix operations upon homogeneous co-ordinates, the matrices being defined for transformations between the picture plane and the platform plane. RELTOPIC performs a two-stage process of projecting the relative position on to the platform along the line of sight, and then performing the matrix application. It is a many-one mapping. PICTOREL is the only one-many mapping. It produces as its result a point in the platform plane which corresponds to the given picture point. The platform point, together with the camera position, specifies a ray on which the point of interest must lie. (0.5K of POP-2)

LIB VISUAL

Provides a debugging facility. An array in core can be bugged so that whenever it is accessed, the T.V. monitor cross wires are moved to a corresponding point in the picture. It enables the user to see which areas of the picture the computer is considering. (0.12K of POP-2)

LIB REGION FINDER

The region finding process of Barrow and Popplestone (1971) has been found to be of considerable use, and has therefore been incorporated into the program library. As well as a procedure for this process, a simple and fast routine is provided for finding well-defined regions, for example, dark objects against a light background. (4K of POP-2)

Software development

Naturally, as hardware is extended and improved, the software which uses it is also extended and improved. Periodically, however, it may become necessary to re-think and re-write. For example, the addition of a second T.V. camera, not originally anticipated, demands some way of switching between data associated with particular cameras; the perspective transformation depends on the camera in use. At the time of writing, an improved set of library programs has been written, but not yet issued for general use. While possessing similar facilities to their predecessors, the scope of the programs is somewhat wider.

One of our aims is to provide an ever-extending library of facilities, with abilities of higher and higher order becoming routinely available. Clearly, as we proceed, such facilities become more interdependent and their development demands systematization of the knowledge we embodied in them, as well a re-appraisal of what has gone before in the light of experimental experience.

Acknowledgements

A number of people have contributed to the development of the equipment and programs described in this paper. A major contribution was made by Mr S.H.Salter of the Bionics Research Laboratories, University of Edinburgh. Steve Salter performed most of the design and construction of the mechanics and associated servo-systems, and also of the T.V. sampling system. Ken Turner, of the Department of Machine Intelligence, designed and built the prototype buffer units, and helped in the development of some of the H316 programs. We wish to acknowledge general financial support given by the Science Research Council, and also by the GPO Telecommunications Headquarters in the form of a contract in connection with the mechanization of parcel handling.

REFERENCES

Barrow, H.G. (1971) LIB LINK EXEC. *Multi-POP Program Library*. Edinburgh: School of Artificial Intelligence.

Barrow, H.G. (1971) LIB PERSPECTIVE. *Multi-POP Program Library*. Edinburgh: School of Artificial Intelligence.

Barrow, H.G. (1971) LIB PICTURE PACK TWO. *Multi-POP Program Library*. Edinburgh: School of Artificial Intelligence.

Barrow, H.G. (1971) LIB VISUAL. *Multi-POP Program Library*. Edinburgh: School of Artificial Intelligence.

Barrow, H.G. & Popplestone, R.J. (1971) Relational descriptions in picture processing. *Machine Intelligence 6*, pp. 377–96 (eds Meltzer, B. & Michie, D.). Edinburgh: Edinburgh University Press.

Barrow, H.G. & Popplestone, R.J. (1972) LIB REGION FINDER. *Multi-POP Program Library*. Edinburgh: School of Artificial Intelligence.

Barrow, H.G. & Salter, S.H. (1970) Design of low-cost equipment for cognitive robot research. *Machine Intelligence 5*, pp. 555–66 (eds Meltzer, B. & Michie, D.). Edinburgh: Edinburgh University Press.

British Standard 4421 (1969) *Specification for digital input/output interface for data collection systems*. London: British Standards Institute.

Burstall, R.M., Collins, J.S. and Popplestone, R.J. (1971) *Programming in POP-2*. Edinburgh: Edinburgh University Press.

Clarke, A.C. (1968) *2001: A Space Odyssey*. London: Arrow.

Dunn, R.D. (1971) LIB LINK-BASED KEY-IN LOADER. *Multi-POP Program Library*. Edinburgh: School of Artificial Intelligence.

Dunn, R.D. (1971) LIB ROBOT MOVE. *Multi-POP Program Library*. Edinburgh: School of Artificial Intelligence.

McCarthy, J., Earnest, L.D., Reddy, D.R. & Vicens, P.J. (1968) A computer with hands, eyes, and ears. *Proc. 1968 Fall Joint Comput. Conf.*, **33**, pp. 329–38. Washington DC: Thompson Book Company.

Nilsson, N.J. (1969) A Mobile Automaton: an application of artificial intelligence techniques. *Proc. of Int. Joint Conf. on Art. Int.*, pp. 509–20. Washington DC.

Popplestone, R.J. (1971) *How Freddy could pack parcels.* (Privately circulated.)

Salter, S.H. (1972) *Arms and the robot*. (Privately circulated.)

Tenenbaum, J.M. (1970) Accommodation in computer vision. *Stanford Artificial Intelligence Project Memo, AIM*–134.