Robotica (1990) volume 8, pp 97–103

A transputer-based system for locating parts and controlling an industrial robot

D.T. Pham,* Huosheng Hu† and J. Pote‡

(Received in Final Form: May 6, 1989)

SUMMARY

A parallel-processing system for locating parts and controlling an industrial robot is proposed. The system employs Transputers and *Occam* to achieve parallelism. In conjunction with a novel vibratory sensor, the system enables a robot to determine the exact location of parts which have been picked up from a semi-ordered work place. A new algorithm for obtaining the coordinates of the parts using the sensed vibration and deflection signals is described. The algorithm dispenses with the lengthy and complex equation-solving procedures previously required. Instead, it only involves looking up a data table and performing simple two-dimensional interpolation calculations. The design of the algorithm to ensure efficient parallel operation is described. Experimental results showing the successful implementation of the algorithm on the proposed system are presented.

Keywords: Parallel processing; Transputers; Occam; Vibratory sensors; Robots; Parts feeding; Bin picking; Stack picking.

1. INTRODUCTION

This paper describes part of a project aimed at developing a robotic cell to demonstrate the concept of stack or tray picking. As explained in previous publications,[1–5] stack or tray picking is a flexible technique of parts feeding whereby a robot is allowed to pick up the parts to be handled directly from a stack or tray rather than from special-purpose feeding equipment. The initial location of the parts is not known accurately. A sensor is employed to find the exact coordinates of the parts once they have been acquired by the robot.

A key element in a stack/tray picking system is the sensor used to refine information on the parts location. A family of low-cost sensing devices have been designed which are all based on vibrating the parts and measuring the resulting natural frequencies of vibration. A problem with the devices has been the time required to process the vibration signals which can range between 10 and 30 seconds for each part to be located. This has been worsened by the slow speed of movement of the robot used in the demonstration cell.

This paper presents a parallel-processing system developed to expedite the operation of both the sensor and the robot. The system comprises an array of Transputers programmed in the *Occam-2* language. Transputers have been utilized instead of conventional microprocessors as in a number of other parallel-processing projects[6–14] to achieve higher communication bandwidths. This is possible due to the inherent structure of the Transputer which facilitates communication through direct one-to-one high speed links and avoids the need for common buses.

This paper contains three main sections. The first describes the hardware configuration of the original and the proposed systems. The second details the organization of the new system software, in particular the design of the part location algorithm. The third presents the experimental results obtained.

2. SYSTEM REALIZATION

2.1 Original system

As shown in Figure 1, the system consists of three computers, a LOCOMAN assembly robot,[15] a vibratory sensor[5] and associated electronic interface. Two of the computers, i.e. the CBM 4032 and Intel 8748 single chip microcomputer are assembled into the robot controller. Figure 2 shows the block diagram of the sensor and control system. It can be divided into two subsystems: the location subsystem (the left half of Figure 2) and the control subsystem (the right half of Figure 2).

A Motorola EXORSET computer was used to obtain the frequencies of free vibration and static deflections measured by strain gauges which are the basic sensing elements in the vibratory sensor. An algorithm residing in the EXORSET computed the coordinates of a part located on the sensor platform. This position and orientation feedback information was then sent to the CBM 4032 which drove the robot to pick up the part from the sensor platform.

The CBM 4032 computer was operated as a central processor which translated the sensory feedback information into motion commands for the 8748 microcomputer. The latter in turn converted these commands into trains of pulses to trigger the driving amplifiers which controlled the robot actuators. The motion of the three axes of the robot occurred sequentially.

---

* Intelligent Systems Research Group, School of Electrical, Electronic and Systems Engineering, University of Wales (Cardiff), Cardiff, CF1 3YH (U.K.).
† Robotics Research Group, Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ (U.K.).
‡ Department of Electronic and Electrical Engineering, University of Birmingham, Birmingham, B15 2TT (U.K.).

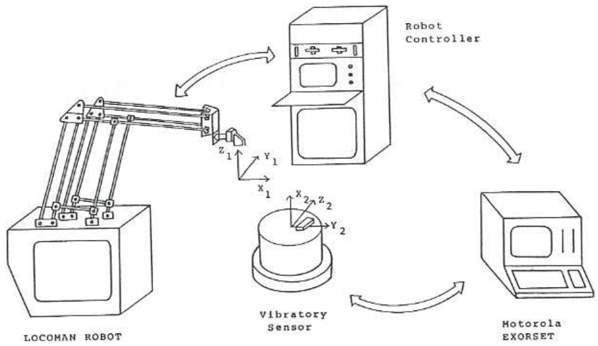


Fig. 1. LOCOMAN Robot and Vibratory Sensor.

There were two main problems to be solved in this system. One was to reduce the calculation time of its location subsystem from more than ten seconds to below one second. The second was to enable the robot's three actuators to be operated concurrently, thus tripling its speed.

2.2 New system structure

Figure 3 shows a block diagram of the new system. There are four parts: an IBM PC computer, a network of three Inmos T414B Transputers, a sensing subsystem and a control subsystem. The IBM PC computer acts as the host computer and is used for software development and control of the complete system. Link adaptor L0 serves as an interface between a serial link of Transputer T0 and the IBM PC bus. The other link adaptors have been configured to provide ten independent byte-wide interfaces with full handshaking,[16] i.e. five input and five output bytes. These interfaces are used to connect the Transputer network to the location and control subsystems.

The three Transputers communicate with one another by means of point-to-point communication links for which the standard bit rate of 10 Mbits/sec has been selected.[17] Transputer T0 is the root Transputer, i.e. the master processor. Transputers T1 and T2 are slave processors. They are directed by T0 in performing their measurement and control functions.

The location and control algorithms are respectively divided into several modules and separately programmed as *Occam* processes. These processes are mapped onto the three Transputers for parallel processing.

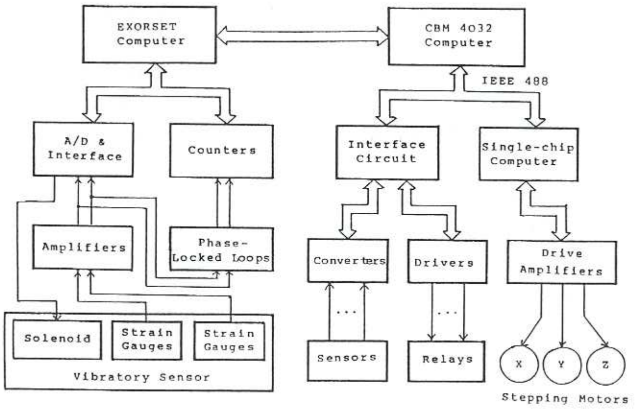


Fig. 2. Block Diagram of Original System.

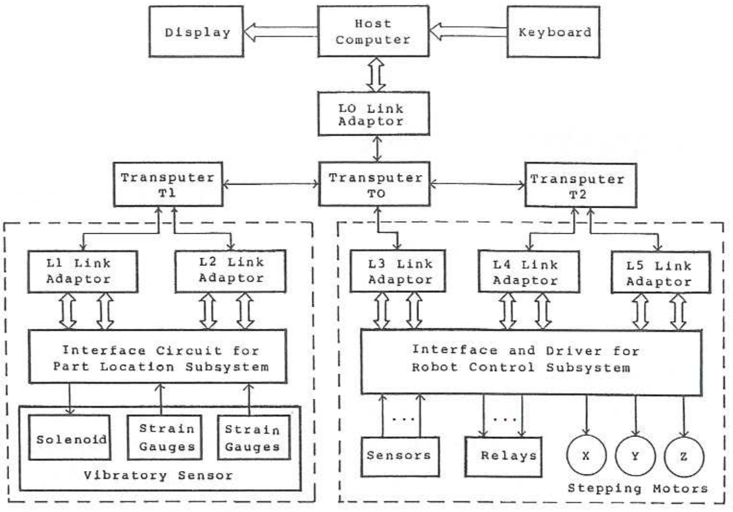


Fig. 3. Block Diagram of Proposed System.

2.3 Location subsystem

The location subsystem is controlled by Transputer T1. The latter takes readings of the static deflections and frequencies of vibration of the sensor platform and uses these to compute the position and orientation of the part. Figure 4 shows the details of the interface between the sensor and the Transputer system.

The interface is arranged as two channels: one for the input of analogue signals (the static deflection signals) and the other for the input of digital signals (the vibration frequency signals). The static deflection signals originate from strain gauge amplifiers. They are fed to the Analogue-to-Digital (A/D) converters. Two vibratory signal frequencies are obtained by two Phase-Locked-Loop circuits (PLL). Here, each PLL is used in such a way that when a signal of frequency within a given range is present at its input, it generates a square wave with the same frequency. The center frequency of each PLL has been separately adjusted to fall in the middle of the respective input signal frequency range.

Two A/D converters and a latch are controlled by Transputer T1 (see Figure 4). The serial stream of control signals from T1 is converted into parallel control words by link adaptor L1 to operate the A/D converters and obtain the *y* and *z* coordinates of the part. Similarly, two counters and a latch, connected via link adaptor L2, are also controlled by Transputer T1. Through L2, T1 activates a solenoid to give an impulse to the sensor to start its vibration. T1 then resets the two counters to begin concurrently counting the numbers of square waves output by the PLL circuits. Simultaneously, a high priority TIMER with a 1.0 microsecond cycle[18] in T1 is used to time the period of the square waves.

3. PROGRAM DEVELOPMENT

3.1 Location algorithm

The vibratory sensor described in refs. 4–5 has been used with the new system. The position of a part located on

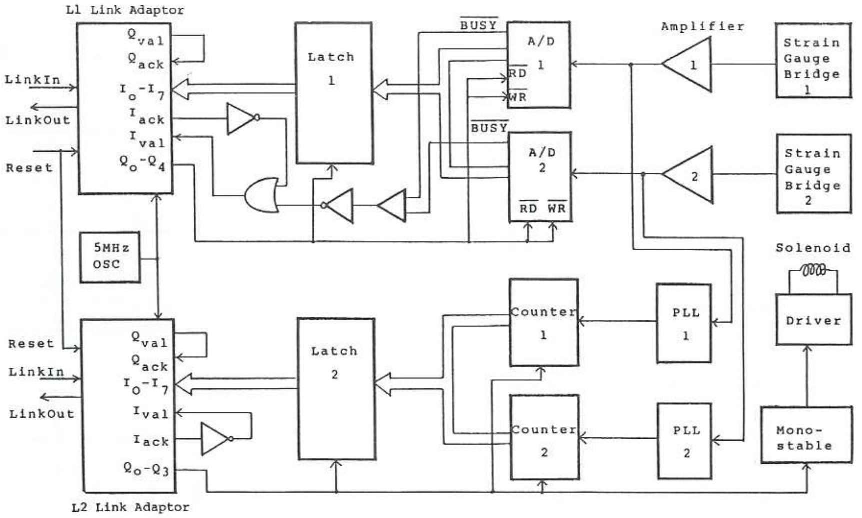


Fig. 4. Functional Block Diagram of Part Location Subsystem.

this sensor can be directly obtained from measurements of its static deflections. However, determining its orientation is non-trivial. In the original system, the orientation algorithm needed to calculate a 6th order determinant up to 180 times. This involved more than 50000 floating-point arithmetic operations.

The table look-up method of obtaining orientation described here is based upon empirical relationships between the periods (or frequencies) of vibration and the orientations of a part placed on the sensor platform. The advantages of the new location algorithm are its simplicity and high speed.

It has been found that if a part placed on the sensor platform is translated along the Y direction, as shown in Figure 1, the period of vibration varies approximately linearly with the Y-coordinate of the part's center of mass (see Figure 5). The same applies when a part is translated in the Z direction (see Figure 6). Because of these almost linear relationships, interpolation may be used here to reduce the size of the look-up table while maintaining a high accuracy level.

The table actually constructed comprises thirty-five pages in each of which the Z coordinate is a constant. The difference between the Z coordinates of adjacent pages is 2 mm. Within each page, the division of Y coordinates and orientation angles is as illustrated in Figure 7 which shows a page of the table. The data in the table are the different vibratory periods. The selection of thirty six orientations and thirty five Y coordinates in each page is a compromise between the size of the table and the acceptable location error.

Figure 8 shows how interpolation is performed to obtain a set of periods of vibration of a part located at point A on the sensor platform. Suppose that A is the center of mass of the part and its Y and Z coordinates have been found. Then four reference points, i.e. B1, B2, B3 and B4, are selected so that they form a square enclosing A. B1 and B2 are on the same page and B3 and B4 on the next page. The sets of periods at points A1 and A2 shown in Figure 8 can then be calculated using the following interpolation equations:

$$P_{a1i} = \frac{y - y_1}{y_2 - y_1} P_{2i} + \left(1 - \frac{y - y_1}{y_2 - y_1}\right) P_{1i} \tag{1}$$

$$P_{a2i} = \frac{y - y_1}{y_2 - y_1} P_{4i} + \left(1 - \frac{y - y_1}{y_2 - y_1}\right) P_{3i} \tag{2}$$

where $i = 0, 1, \ldots, 35$. P_{1i}, P_{2i}, P_{3i}, P_{4i} are the four sets of periods corresponding to B1, B2, B3 and B4. P_{1i}'s and P_{2i}'s form two adjacent columns in the first of the pages, and P_{3i}'s and P_{4i}'s, two adjacent columns in the second page. P_{a1i} and P_{a2i} are two sets of periods at A1 and A2.

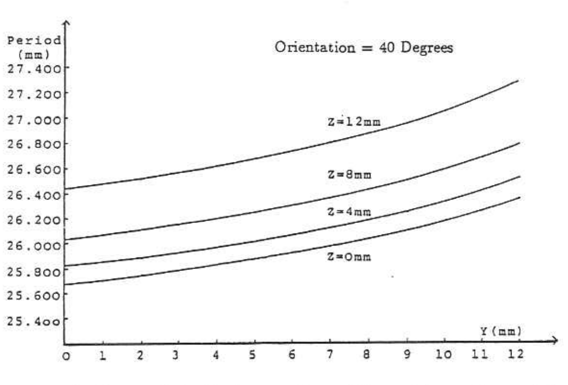


Fig. 5. Variation of Period as a Part Translates along Y-axis.

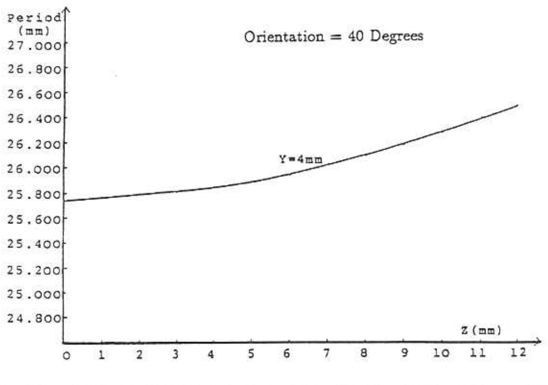
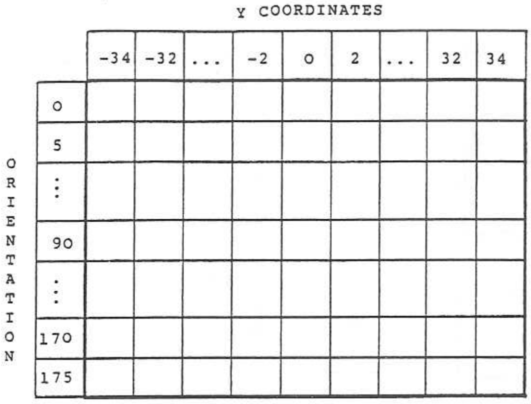


Fig. 6. Variation of Period as a Part Translates along Z-axis.



	-34	-32	...	-2	0	2	...	32	34
0									
5									
⋮									
90									
⋮									
170									
175									

Fig. 7. A Page of the Look-up Table.

Similarly the set of periods for point A may be obtained from those for points A1 and A2 by a second process of interpolation. Hence from (1) and (2), the overall interpolation equation for the set of periods at point A is:

$$P_{ai} = (1-k_1)(1-k_2)P_{1i} + (1-k_2)k_1P_{2i} + k_2(1-k_1)P_{3i} + k_1k_2P_{4i} \quad (3)$$

where $i = 0, 1, 2, \ldots, 35$, $k_1 = (y-y_1)/(y_2-y_1)$, $k_2 = (z-z_1)/(z_2-z_1)$.

Figure 9 shows the flow chart of this location algorithm which is divided into four execution steps. First, four reference points need to be located according to the measured position of the part's center. Second, the four sets of periods corresponding to these points are obtained from the look-up table. Third, interpolation equation (3) is used for computing the set of periods at point A. These periods relate to thirty six orientations of the part at 5 degree intervals (0, 5, 10, . . . , 175 degrees). Finally, according to the measured period, the orientation of the part is calculated by applying the following interpolation equation:

$$\gamma = \gamma_{ai} + k\frac{P - P_{ai}}{P_{ai+1} - P_{ai}} \quad (4)$$

where $i = 0, 1, \ldots 0, 35$. $k = \gamma_{ai+1} - \gamma_{ai} = 5$. γ lies between γ_{ai} and γ_{ai+1}.

P is the measured period. P lies between P_{ai} and P_{ai+1}. γ_{ai} and γ_{ai+1} correspond to periods P_{ai} and P_{ai+1} respectively.

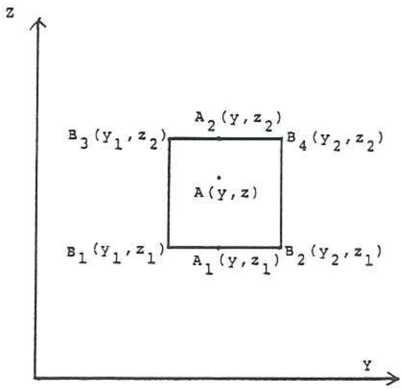


Fig. 8. Points Used During Interpolation.

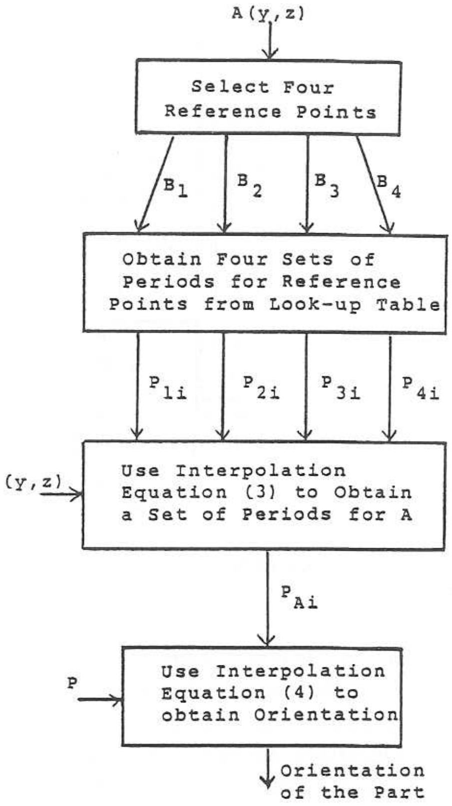


Fig. 9. Flow Chart of Part Location Algorithm.

It should be noted that a look-up table is particular to a certain part. Therefore, if the part is changed in size and/or mass, then the contents of the look-up table, i.e. all the period values, also need to be modified.

3.2 Top level Occam process

In the design of software for an array of Transputers, an important early decision is the manner of sharing the processing load between the different Transputers. In this application this depends upon how the location and control tasks may be subdivided so that maximum use of the parallel capability (the highest processing speed) can be achieved. Figure 10 is an overall functional block diagram of the system. The monitor handler, display handler and keyboard handler, as shown in Figure 10, may be written directly as three *Occam* processes that are allocated to the master processor, Transputer T0.

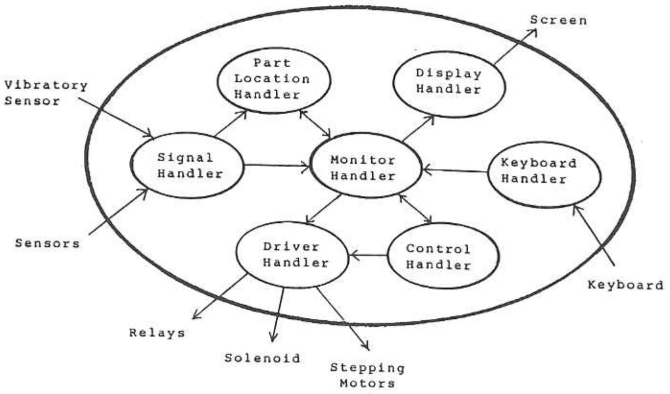


Fig. 10. Overall Functional Block Diagram of the Software.

```
{{{F Network
... SC T0.process
... SC T1.process
... SC T2.process

CHAN OF ANY Keyboard, screen, appl.in, appl.out:
CHAN OF ANY app2.in, app2.out, ad.in, ad.out:
CHAN OF ANY count.in, count.out, sensor.in:
CHAN OF ANY control.outl, control.out2, relay.out:

PLACED PAR
  PROCESSOR 0 T4
    ... Channel allocation
    to.process (keyboard, screen, appl.in, appl.out,
                        app2.in, app2.out, control.outl)
  PROCESSOR 1 T4
    ... Channel allocation
    t1.process (appl.in, appl.out, ad.in, ad.out,
                                   count.in, count.out)
  PROCESSOR 2 T4
    ... Channel allocation
    t2.process (app2.in, app2.out, sensor.in,
                                relay.out, control.out2)
}}}
```

Fig. 11. Details of Top Level *OCCAM* Process.

However, the signal handling and driver handling tasks need to be divided into several subtasks. These subtasks are again written as separate *Occam* processes and respectively mapped onto the three Transputers for concurrent execution.

In practical application, after the part location handler is executed, i.e. the position and orientation of a part are obtained, the control handler is activated. If they are executed by two Transputers, a situation could arise where only one Transputer is busy and the other two are idle. Therefore, it is necessary to divide the part location and control handler into three subtasks and program them into corresponding *Occam* processes. These processes are then allocated to the three Transputers for concurrent execution. The details of the division will be discussed in the next section.

Based upon the above considerations, the top-level *Occam* process has been designed as shown in Figure 11. Three processes, t0.process, t1.process and t2.process, are respectively mapped onto Transputers T0, T1 and T2. In the t0.process, there are six subtasks such as display, keyboard, monitor, part location, control and driver processes, some of which are executed in parallel. Similarly, t1.process is composed of signal, part location, driver and control processes, and t2.process includes control, signal, driver and part location processes.

3.3 Parallel design consideration

In a high-concurrency system, every processor is always busy. During the design of the program, efforts were made to divide the whole task of the system into several subtasks to achieve high concurrency. As mentioned above, these subtasks are allocated to appropriate processors in the system. In this section, the details of the division of the part location and control tasks are discussed.

The control task includes the control of three stepping motors, eight relays and a solenoid, see Figure 3. The control of the relays is very simple and can be programmed as a process allocated to Transputer T2. Similarly, the solenoid control process is mapped onto Transputer T1. However, the control of the three motors requires three control algorithms. These algorithms could be programmed as three processes in *Occam* and then mapped onto a single Transputer, says T2. But this means Transputers T0 and T1 may be inactive during the execution of the control algorithms. Hence, it was decided to map the three processes onto the three Transputers with each Transputer acting as a digital controller to control one stepping motor. The calculation results, i.e. control values, are concurrently output to the motor's interface circuit by three serial links in T0 and T2.

Similarly, the part location algorithm has also been divided into three subtasks and allocated to the three Transputers. The method of division is different from above. It may be deduced from Figure 9 that the execution of the location algorithm is not entirely repetitive. As mentioned before, four sets of periods for four reference points need to be obtained from the look-up table according to the position of the part's center of mass. An interpolation equation is then used to compute a set of periods corresponding to different orientations at that position. This involves thirty-six calculations yielding thirty six periods which relate to 36 orientations (0, 5, 10, . . . , 175 degrees). These computations have been divided into three separate processes and then also allocated to the three Transputers to realize parallel processing. Figure 12 shows the flow chart of the parallel part location algorithm. There are five execution steps in this

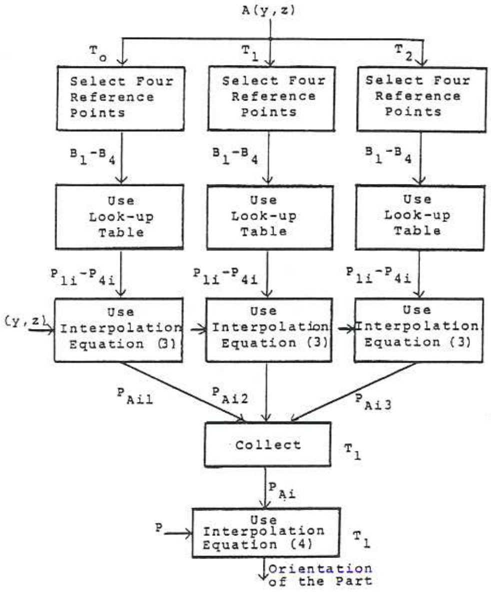


Fig. 12. Flow Chart of Proposed Algorithm.

algorithm. The first three steps are concurrently executed by the three Transputers. In the second step, the look-up table is split up into three parts, each covering 12 orientations. This means that each Transputer need only do one third of the interpolation calculations in the third step. In the 4th step Transputer T1 collects the results from Transputers T0 and T2. All of the results, i.e. P_{ai1}, P_{ai2} and P_{ai3}, form the set of periods (P_{ai}) of vibration at the measured point. According to the calculated 36 periods and the measured period, interpolation equation (4) is executed on Transputer T1 to obtain the actual orientation of the part. This is the fifth step.

3.4 Construction of t1.process

Based upon the parallel design of the location and control algorithms, t1.process consists of part location, control, position and period measurement processes. Transputer T1 is controlled by Transputer T0 to begin the measurements of the position and periods. The measurement results, Y and Z coordinates as well as the periods, are sent to Transputer T0 and T2. This ensures that the part location and control algorithms are executed in parallel on the three Transputers. T0 also sends these data for display on the host computer screen. During the execution of the control process, T1 transfers control values to T0 for output to the motor interface. The details of t1.process are shown in Figure 13.

```
{{{F t1 process
PROC t1.process (CHAN OF ANY appl.in, appl.out, ad.in,
                                   ad.out, count.in, count.out)
  ... initialization
  ... position measurement
  ... period measurement
  ... part location process
  ... control process
  SEQ
    ... assignment
    ... preprocessing
    WHILE TRUE
      SEQ
        appl.in ? up.order
        IF
          up.order = measurement
            SEQ
              count.out ! acti.solen
              PAR
                period.measuring ( )
                position.measuring ( )
              ... results transfer
          up.order = location
            SEQ
              location.process ( )
              ... transfer results
          TRUE
            SKIP
:
}}}
```

Fig. 13. Details of t1.process.

4. EXPERIMENTAL RESULTS

4.1 Execution speed of the system

During the development of the system an investigation of the execution speed was carried out. Also the relative activity of each Transputer was assessed. This is an important parameter as the advantage of using several processors in parallel (faster execution speed) is reduced unless all the processors are usefully employed most of the time. The processes of the part location algorithm were mapped onto different numbers of Transputers to obtain the execution speeds and the activities of the Transputers for different configurations. This was easily realized with small changes to the software. The measurements reported below were made by means of the high priority TIMER provided in the Transputers which counts every microsecond as already mentioned. The results are given in Table I.

These results show a significant improvement in the execution time of the location algorithm when the number of Transputers is changed from one to three. At the same time, a deterioration in the Transputer average activity can be observed. This is because Transputers T0 and T2 are not used in steps 4 and 5, as can be seen in Figure 12. In fact, the activities of Transputers T0 and T2 in the whole system are larger than the above values indicate. The reason is that T0 and T2 also execute other processes, such as the monitor and sensor processes, after completing the part location process.

The total time for locating a part was measured at 364 ms of which 351.45 ms is the sampling time. The speed of the new system is over thirty times faster than the original system.

4.2 Location accuracy

A test part with dimensions $10 \times 38 \times 77\ \text{mm}^3$ and a weight of 238.3 grams was used to evaluate the location accuracy. With the part placed at different orientations, it has been found that the new system can achieve location errors of less than 0.4 mm in position and ±2 degrees in orientation.

4.3 Locating parts using a T800 transputer

A T800 Transputer, which has an integral 64-bit floating-point arithmetic unit, was also used to implement the location algorithm. All processes of the algorithm were mapped onto the T800 with little change. The speed of execution of the part location algorithm was then measured as 5.4 ms. This is about six times faster than for one T414.

Table I Execution times of the location algorithm

Transputer	Execution time	Activity	Average activity
T0	32.408 ms	100% (T0)	100%
T0, T1	17.512 ms	100% (T0), 85% (T1)	92.5%
T0, T1, T2	12.55 ms	100% (T1), 79% (T0), 78% (T2)	85.6%

5. CONCLUSIONS

In this paper an investigation into the use of a Transputer-based parallel-processing system for robot control and sensing has been presented. In the investigation, a new part location algorithm for a vibratory sensor has been programmed as *Occam* processes and mapped onto a network of three Transputers to execute concurrently. A location time of 364 ms has been obtained, which is over thirty times smaller than in the original system.

The proposed system is a better alternative than parallel processing systems using traditional sequential microprocessors from the point of view of execution speed, ease of programming, extendability and flexibility. However, it has also been found that considerable care is needed when dividing up the algorithm for parallel execution to ensure that all processors in the system are kept busy most of the time.

6. ACKNOWLEDGEMENTS

The authors would like to thank Dr M.R. Jane and Mrs F.M. Childs, SERC Transputer Initiative, and Professor D.J. Creasey, University of Birmingham, for the loan of the Transputer equipment used in this work.

References

1. D.T. Pham and M.W.M.G. Dissanayake, "Feasibility Study of a Vibratory Sensor for Locating 3-D Objects", *Proc. 25th Int. Machine Tool Design and Research Conference, Birmingham, U.K.*, (1985) pp. 201–211.
2. D.T. Pham and M.W.M.G. Dissanayake, "A Three Degree-of-Freedom Inertial Sensor for Locating Parts" *Proc. 15th Int. Symp. On Ind. Robots, Tokyo, Japan* (1985) pp. 613–629.
3. D.T. Pham and M.W.M.G. Dissanayake, "Inertia-Based Method for Locating 3-D Objects" *Proc. 5th Int. Conf. on Robot Vision and Sensory Control, Amsterdam, The Netherlands* (1985) pp. 223–237.
4. D.T. Pham and J. Menendez, "A Vibratory Device for Locating Objects: Theory and Experimental Results" *Proc. Int. Conf. on Computer-Aided Design, Manufacture and Operation in the Automotive Industries, Switzerland* (1987) pp. 121–137.
5. D.T. Pham and J. Menendez, "Development of a Six-Degree-of-Freedom Vibratory Device for Locating Objects", *Int. J. Machine Tools and Manufacture,* Special Issue on Robotics and *AI,* (D.T. Pham, ed.) **28,** No 3, 197–205 (1988).
6. J.Y.Z. Luh, "Scheduling of Distributed Computer Control Systems for Industrial Robots" *Proc. 3rd IFAC Workshop DCCS,* New York, Pergamon (1981) pp. 85–102.
7. M. Kametani, T. Watanabe and K. Kawada, "Speeding up of Robot Control Calculation by Parallel Processing using Multimicroprocessor" *Proc. 1st Conv. Robotics Society of Japan* (Dec. 1983) pp. 167–168.
8. K. Plessmann, "A Multi-microcomputer-based Robot Control System" *Proc. IFAC Real Time Digital Control Applications Conf.* (1983) pp. 265–282.
9. S. Elgazzar, D. Green and D. O'Hara, "A Vision-based Robot System using a Multi-processor Controller" *Nat. Res. Council, Canada Ref. 23485* (June, 1984).
10. R. Gaglianello and H. Katseff, "A Distributed Computing Environment for Robotics" *Proc. 1986 IEEE Int. Conf. Robotics and Automation* **3,** 1890–1896 (April, 1986).
11. C. Isik and A.M. Meystel, "Pilot Level of a Hierarchical Controller for an Unmanned Mobile Robot" *IEEE J. Robotics and Automation* **4,** No. 3, 241–255 (June, 1988).
12. R.C. Luo, M.N. Lin and R.S. Scherp, "Dynamic Multi-Sensor Data Fusion System for Intelligent Robots" *IEEE J. Robotics and Automation* **4,** No. 4, 386–396 (Aug., 1988).
13. H. Kasahara and S. Narita, "Parallel Processing Scheme for Robot Control Computation using Multiprocessor Scheduling Algorithm" *J. Robotics Japan* **2,** 387–401 (Oct., 1984).
14. H. Kasahara and S. Narita, "Parallel Processing of Robot-Arm Control Computation on a Multimicroprocessor System" *IEEE J. Robotics and Automation* **RA-1,** No. 2, 104–113 (June, 1985).
15. D.T. Pham and P.T.N. Pham, "The Locoman Assembly Robot" *Proc. 4th Int. Conf. on Assembly Automation Tokyo, Japan* (Sept., 1983) pp. 24–38.
16. Inmos, "IMS C011 Link Adaptor" *Engineering Data* (Inmos Ltd., Bristol, U.K., Aug., 1987).
17. Inmos, "IMS T414 Transputer" *Engineering Data* (Inmos Ltd., Bristol, U.K., Aug., 1987).
18. C.A.R. Hoare, *OCCAM 2 Reference Manual* (Prentice Hall, U.K., 1988).
19. P.J. Fleming, D.F. Garcia Nocetti and H.A. Thompson, "Implementation of a Transputer-based Flight Controller" *Proc. IEE Int. Conf. on Control 88,* Oxford, England (April, 1988) pp. 719–724.
20. D.I. Jones and P.M. Entwistle, "Parallel Computation of An Algorithm in Robotic Control" *Proc. IEE Int. Conf. on Control 88, Oxford, England* (April, 1988) pp. 438–443.

KYBERNETES

International Journal of Cybernetics and Systems
(a quarterly)

EDITOR: B. H. Rudall

ASSISTANT EDITORS:

Professors
W. H. von Alven (USA), F. H. George (UK), T. C. Helvey (USA),
E. Nicolau (Romania) and Wu Xuemou (China)

ADVISORY BOARD:
Professors
A. Bensoussan (France), Chen Hanfu (China), S. Dillon Ripley (USA)
and M. Manescu (Romania)

Founder Editor: Professor J. Rose (UK)

There are also 15 section editors from 8 countries and 21 members of the Editorial Board from 11 countries

This journal is concerned with the interdisciplinary study of cybernetics and systems in the widest sense. Among the extensive range of topics covered are the following: *Artificial Intelligence, Automation, Cybernetic Modelling, Computer Simulation, Biocybernetics, Economic and Social Systems, Nature and Validation of General Systems, Ecosystems, Adaptive Systems, Philosophy of Cybernetics, the Interrelation between Cybernetics and other Sciences.*

Kybernetes, now in its eighteenth year of publication, aims to endow cybernetics and general systems with an authoritative voice of its own and to establish a competent forum for the exchange of knowledge and information in relevant fields. One of the primary objectives of this publication is to weed out pseudo-cybernetics claims and to base cybernetics on a sound foundation; hence, highly speculative papers that lie on the fringe of cybernetics and systems sciences will not be published. The journal contains only *original papers*; it is the official publication of the World Organisation of General Systems and Cybernetics.

Enquiries and orders to:

MCB UNIVERSITY PRESS
62 Toller Lane
Bradford
BD8 9BY
England
Tel. 0274 499821
Telex. 51317 MCBUNI G
Fax. 547143

For Subscription Rates, Advertising Rates and Back Issues enquire at MCB University Press, as above.

A PRESTIGE PUBLICATION, INDISPENSABLE TO
YOUR ORGANISATION AND TO YOU
NOW IN ITS EIGHTEENTH YEAR