SOFTWARE INTERFACE AND VISION SYSTEM FOR A SCARA ROBOT

Sean G. McSweeney and William M. D. Wright

Department of Electrical and Electronic Engineering, University College Cork, College Road, Cork, Ireland

Email: s.g.mcsweeney@mars.ucc.ie; bill.wright@ucc.ie

ABSTRACT

This work describes the development of a novel software interface for an AML-based IBM 7545 SCARA robotic arm to a host computer, and its subsequent integration into a vision system utilising cross-correlation and other object recognition techniques. The purpose of the work was the development of an 'intelligent' robot-centred manufacturing cell capable of recognition, classification and sorting of various randomly orientated components fed into the manufacturing cell. A pressure-sensing feedback controller for the SCARA gripper was also developed, with the gripping pressure controlled through a NI 6259 DAC incorporated into the overall software architecture. The gripper controller could not be implemented via AML, so various software languages were used for the development of the overall software interface including AML, Visual C++, MATLAB and LabVIEW. This upgrade of the original IBM 7545 AML DOS-based system to interface to Windows-based applications required a large degree of manipulation of the data packets on the host in order to make commands intelligible to the controller RS232 uplink. Future work includes the completion of a GUI for the system, the use of neural networks for online 'pick and place' decisions, and the optimisation of vision system recognition techniques through the minimisation of the processing required for Fourier Mellin methods using K-means clustering.

KEYWORDS: Robotics; Machine Vision; Control

1. INTRODUCTION

Robots, more specifically robotic arms are the workhorse of a number of the major industries of the world, two of the most well known being the automotive and electronics industries. A robotic arm is most often tailored to the task that it is required to perform, be that pick and place operations or more complex assembly operations. In spite of this level of customisation of robotic arms there are a number of standard configurations [1, 2] including gantry, Cartesian, cylindrical, SCARA and articulated. The link diagram of the SCARA configuration is illustrated in Figure 1.

Significant advances have been made in recent years in the area of understanding the kinematics [3-5] and jacobians [6-8] of complex and simple linked structures and this work makes use of many of these advances.

Analysis of an isolated robot arm on its own is insufficient to bring it into the context of a manufacturing environment. For the arm to be of any meaningful use it is essential that the supporting architecture for the arm be in place, namely the surrounding workcell.

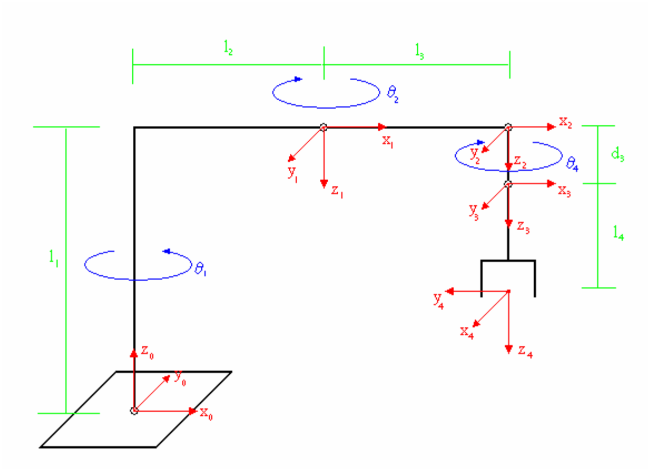


Figure 1: Link diagram of SCARA robot.

There is a huge number of potential configurations of workcells. The standard workcell consists of a number of sensors and mechanisms to service the central machine, quite often a robotic arm, however in many circumstances the manipulator is not even the central machine in a workcell, but rather a loading mechanism for another machine. The fundamental architecture of a workcell involves a "master-slave" system, in which the intelligent host computer controls the simple sensing devices (through PLC's (Programmable Logic Controller) if there are PLC's in the cell or directly if there are not). The hierarchical communication architecture is shown in Figure 2 with a number of typical devices in a manufacturing cell:

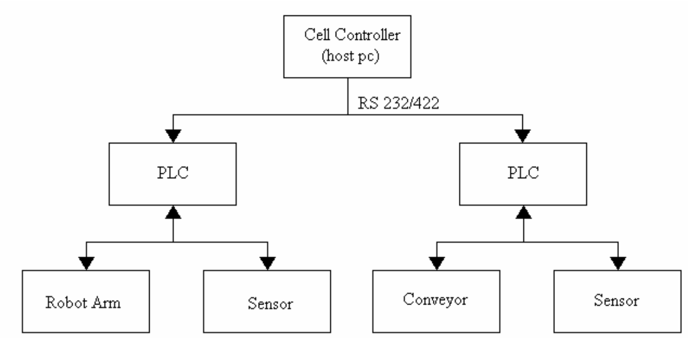


Figure 2: Overview of cell architecture.

In all manufacturing cells communication standards are extremely important to the continued transfer of data and some automated cell structures could not operate without the sharing of large amounts of data [9]. Control Systems also form an essential part of any work cell. The typical control methodologies employed are simple on/off control applied by the use of a plc (programmable logic controller), however for the purposes of manipulators such as the SCARA tool tip force sensing capabilities are also commonly found [10, 11]. The use of control and control systems is an essential part of the project and thus a section is dedicated to giving relevant background theory on both modelling and implementation.

Shape recognition is an integral part of any computerised imaging system used in a manufacturing cell and is the thrust of much on-going research [12]. Shape recognition in 2-D space, although less complicated than using 3-D, is far from an easy problem and is a far greater challenge than the previously mentioned edge detection. One of the simplest methods of object recognition is through the use of a template mask matrix of an expected object and convoluting this matrix with the image greyscale matrix to determine the location of the mask within the image matrix. However this method does not account for orientation and scaling of the template matrix, i.e. rotation of the expected object and image size of the expected object due to distance from the camera and homogenous manipulation of the image plane required for robust recognition.

2. BACKGROUND THEORY

The method of template matching used in this work was simple cross correlation. This method is typically utilised where a background is in motion but the object dimensions remain the same and is ideally suited to optimising speed of recognition. The 2D cross correlation discrete equation is given by:

$$C(i,j) = \sum_{m=0}^{Ma-1} \sum_{n=0}^{Na-1} A(m,n).conj(B(m+i,n+j)) \qquad where \begin{cases} 0 \le i < Ma + Mb - 1 \\ 0 \le j < Na + Nb - 1 \end{cases} \tag{1}$$

which is derived from the estimate of the true cross correlation formula given by the discrete sequence:

$$\hat{R}_{xy}(m) = \begin{cases} \sum_{n=0}^{N-m-1} x_n +_m y_n^* & m \ge 0 \\ \hat{R}^*_{yx}(-m) & m < 0 \end{cases} \tag{2}$$

With equation (1), object tracking within an acceptable time frame is possible.

A conceptual closed loop system can be given by the loop configuration diagram representing data flow shown in Figure 3:

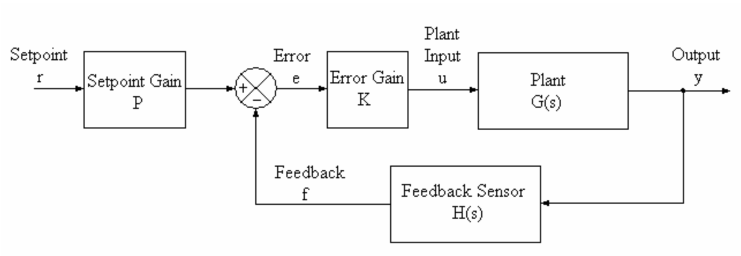


Figure 3: Closed loop system

Utilising the Laplace representation of the transfer functions feedback control can be easily translated from the geometric representation above to mathematics, for instance the transfer function to encapsulate the above diagram (Figure 3) of a generic system would be:

$$\frac{Y}{R} = P\frac{KG(s)}{1+KG(s)}H(s) \tag{3}$$

where *Y* and *R* are the Laplace transformations of the time varying *y* and *r* respectively shown in Figure 3.

3. EXPERIMENTATION AND SIMULATION

It was established that the serial communications protocol was software handshaking utilising the 'Xon' and 'Xoff' protocol i.e. ASCII 17&19. Additionally, it was found that the protocol of the IBM 7545 controller was that an 'EG' ASCII string be sent by the transmitter to indicate to the receiver that the data packet was complete. Controller end communications which were limited to involve low level commands (i.e. simple move commands) were handled in AML while Host end communications which would involve packaging of data and other complex operations were handled by Matlab. Matlab and not C++ was chosen for this task as implementation speed of the packaging would not be critical as the controller end process would be the limiting factor in execution speed. Figure 4 illustrates this communication method.

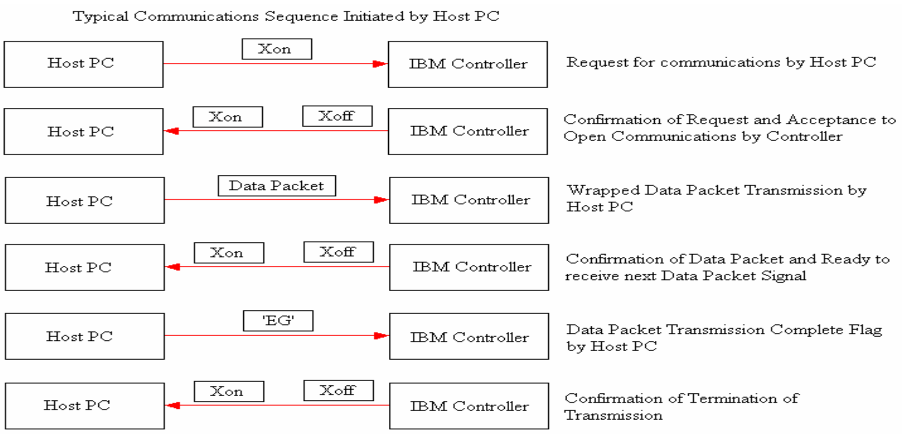


Figure 4: Communications sequence of IBM 7545 controller

With the communications sequence established, some basic communications with the controller could be initiated. As a progression from basic communications, a cyclic program was written on controller end of communication that waited for confirmation of initiation of communications from the host PC for coordinates. On the host end of communications, this meant that if the cyclic program was being accessed for coordinates for the first time in a sitting, then initiation of the program before sending coordinate data would have to be done. This was implemented automatically using an additional input into functions governing communications on the host end. The communications data block minus the external packaging was required to be encoded into the hexadecimal of mantissa representation of itself and then this representation converted into a character string. Thus the required software architecture was developed, which handled communications on both the controller and host end.

As mentioned above in the first approach taken to tackling the problem of controlling the gripper was to use a NI (National Instruments) DAQ (Data AcQuisition device) 6259 with downloaded drivers developed by an MIT group to act as plug-ins from LabView© to MATLAB©. After much testing of this approach using 741 op amps to buffer and boost the signal the bandwidth of the system was found to be too narrow (1 second lag). This sluggish response of the system, it was believed, would lead to instability in a closed loop system, as the time delay in response of the system to a setpoint is unacceptably long. A better solution to the problem of establishing open loop control of the pneumatic gripper was developed and thus a replacement to the DAC with superior response for the task. The developed architecture is shown in Figure 5.

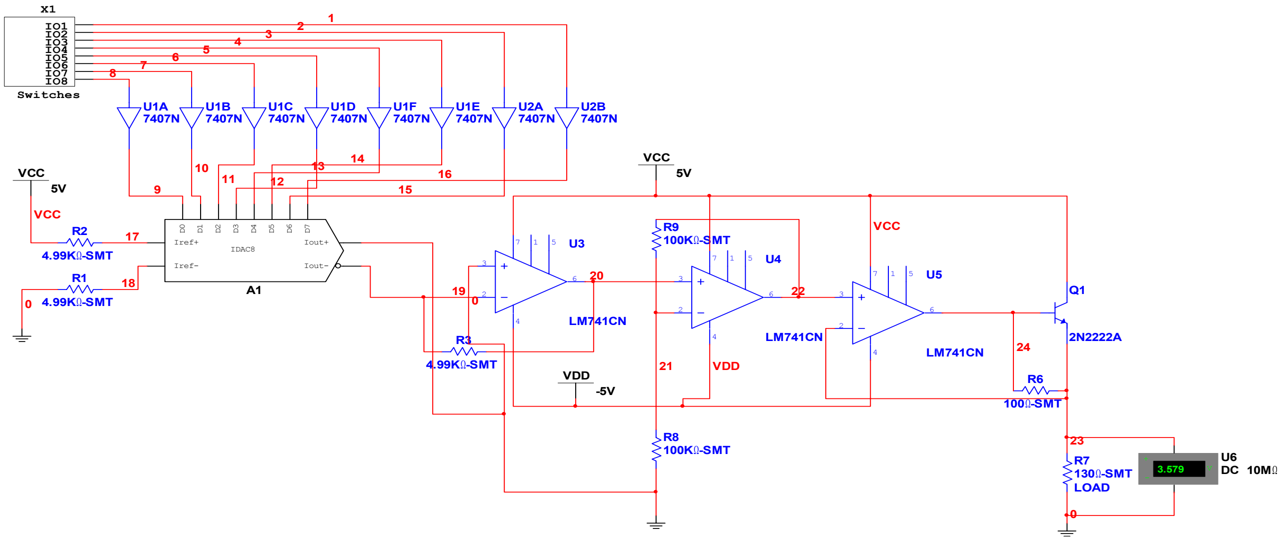


Figure 5: Gripper open loop circuit.

An algorithm was built to perform template matching cross correlation in ‘semi-real time’ (5 frames/sec) as this would mean that a much more reliable reading for the location of the target object could be obtained in a reasonably complex background. The method devised was initially to rotate the image of the target object, perform a 2D cross correlation, locate the maximum and determine this to be the object. This method was however found to result in errors due to the cross correlation attempting to match the black regions in the image once it had been rotated. As a result of these errors it was decided to rotate the camera image itself, more computationally intensive but more robust. The algorithm was designed to use drill down levels which scan across the range in steps of 1/12 of the range i.e. the first level would be steps of 30^o, the next in steps of 5^o to $\pm 30^o$ and so on to whatever level of detail is required. The algorithm then performs an analysis of the cross correlation, determining the maximum and highlights this as the target object. In addition to this operation the algorithm also outputs the rotation of the object, as it is anticipated that this will be required by its caller function in order to integrate the various software elements. The algorithm repeatedly analyses the input stream from the camera and locates a template image within the camera image and highlights its location. The template image is selected from an earlier snapshot from the camera and must be smaller (i.e. have less pixels) than the image that it is being searched for in. In order to facilitate rapid testing of the code and make inputting new objects trivial code was written to allow the selection of a template image by simply dragging a selection box over an area within the initial snapshot. A template that has been selected is shown in Figure 6, this template is then scanned across the rgb camera image.

Figure 6: Template image.

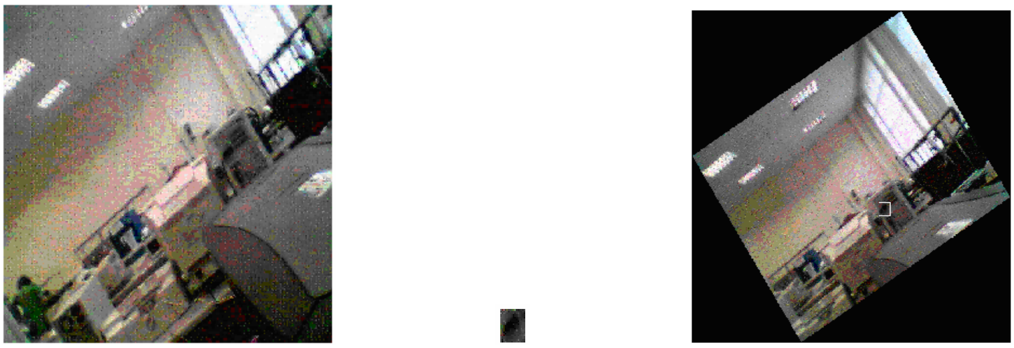
The next step in the process was to code in a peak detect in the cross correlation and then show a maximum at the point of best fit (i.e. the point in the image that the template most closely resembles). Accommodation for the variation in template sizes selected was made through the use of arbitrary assignment of the box dimensions and offsets depending on the template. In Figure 7 the matching of the template to itself in the scan image can be seen.

Figure 7: Simple cross-correlation target acquisition.

These two systems were tied together under a master control algorithm, which calls on the vision system protocols and move commands as needed. The development of this algorithm was trivial as it merely uses plug-ins from the aforementioned vision system and motion algorithms.

4. RESULTS AND DISCUSSION

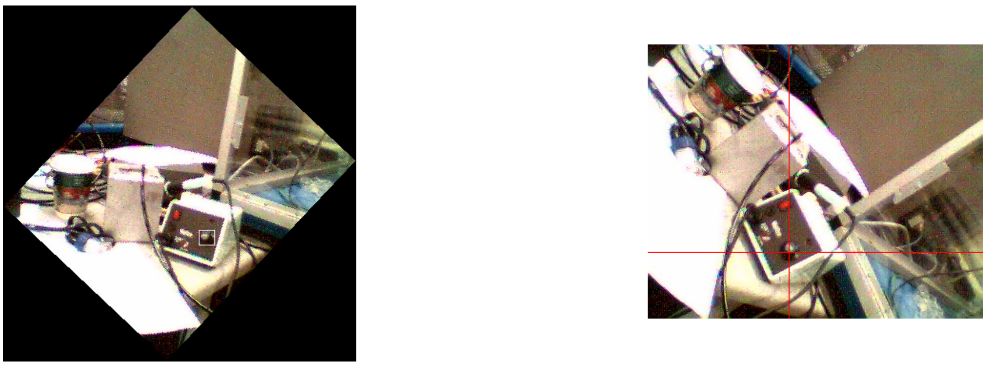
The tests conducted on the vision system that were to determine how far the method could be taken (i.e. how small and obscure an image could be located) firstly and then the accuracy of the homogenous transformation algorithm and thus the vision systems ability to locate the template in the original image was tested. To perform the first test the template image, Figure 8(b) was extracted from Figure 8(a), which is the power cable to the desktop seen on the right of the image. As can be seen this object size is almost undistinguishable to the human eye and thus could be assumed to be the smallest possible element that would be used in a manufacturing cell. Figure 8(b) was located within Figure 8(c), which is the image taken from the camera at an angle of 32.5° to the original template rotated back into line with the original image.



(a) (b) (c)

Figure 8: (a) Original, (b) template and (c) location images.

From this test of the system it was clear that the method could handle almost any possible 2D rotation of an object so long as the background of the original image that the template object is determined from and that in which the template is searched for remain reasonably similar. In Figure 9(a) the location of the target object (the soldering iron fuse head) in the rotated image is shown while Figure 9(b) displays its successful determination in the original camera image thus displaying the correct functioning of the transformation algorithm, which then performs another transformation into the robotic arm plane thus allowing a move command to be issued for pick and place operations:



(a) (b)

Figure 9: (a) Target acquisition and (b) absolute determination.

5. CONCLUSIONS AND FUTURE WORK

In conclusion this report outlined how control of the SCARA was fully achieved, error free, through serial port communications with the IBM controller by a 'host' PC. It has been described how communication protocols for the IBM7545 controller were built up on the 'host' PC end from the bit level upwards using MATLAB© including testing and debugging. The outline was given for a cyclic program that was developed on the controller end of communications using the programming language AML to deal with the requirement for continuous move commands being sent to the controller. A description of the developed vision system has also been provided and outlines of a number of algorithms developed. It has been shown how using MATLAB© algorithms that perform edge detection and algorithms that recognise objects using template matching have been developed. In addition the integration of the two separate systems has been outlined briefly.

In the area of imaging for future work it is planned to implement the Fourier Mellin Transform, initially in MATLAB© and finally to migrate the algorithm to C++, where the increase in execution speed, from observing demos in the OpenCV library should be significant.

REFERENCES

[1] R. H. Bishop, *The Mechatronics Handbook*, CRC Press Inc, 2002.

[2] Mikell P. Groover*, Fundamentals of Modern Manufacturing:Materials, Processes and Systems*, 2nd edition, John Wiley and Sons Inc., 2001.

[3] B. Sprenger, L. Kucera and S. Mourad*,* Balancing of an Inverted Pendulum with a SCARA Robot, *IEEE/ASME Trans. Mechatronics*, Vol. 3, No. 2, (1998), 91-97.

[4] S. P. Chan and H. P. Chen, An Efficient Algorithm for Identification of SCARA Robot Parameters Including Drive Characteristics, *Proc. 25th Annual Conference of the IEEE Industrial Electronics Society (IECON '99)*, San Jose, California, 1999, 1014-1019.

[5] J. Angeles, A. Morozov and O. Navarro, A Novel Manipulator Architecture for the Production of SCARA Motions, *Proc. 2000 IEEE International Conference on Robotics and Automation (ICRA'00)*, San Francisco, California, 2000, 2370-2375.

[6] Gyula Mester, Adaptive Force and Position Control of Rigid Link Flexible Joint SCARA Robots, *Proc 20th Annual Conference of the IEEE Industrial Electronics Society (IECON '94),* Volume 3, 1639 - 1644

[7] Wolfgang Risse and Manfred Hiller, Dextrous Motion Control of a Redundant SCARA Robot, *Proc 24th Annual Conference of the IEEE Industrial Electronics Society (IECON '98),* Aachen, Germany, 1998, 2446-2451 vol.4

[8] Masatoshi Nakamura, S. Rohan Munasinghe, Satoru Goto and Nobuhiro Kyura, Enhanced Contour Control of SCARA Robot Under Torque Saturation Constraint, *IEEE/ASME Transactions on Mechatronics*, Vol. 5, No. 4, Dec. 2000, 437-440

[9] Ching-Long Shih, J. Peter Sadler and William A. Gruver, Collision Avoidance for Two SCARA Robots, *Proc. 1991 IEEE International Conference on Robotics and Automation*, (674-679) vol. 1

[10] Yutaka Yoshida, Akihiko Yabuki, Yasuyuki Nakata and Katsushi Nishimoto, High Speed Force Controller for SCARA Robots, *Proc. 15th Annual Conference of the IEEE Industrial Electronics Society (IECON '89),* 629-633 vol.3

[11] Naoki Uchiyama and Shoji Takagi, Robust Control System Design for SCARA Robots Using Adaptive Pole Placement, *IEEE Transactions on Industrial Electronics*, Vol. 52, Issue 3, 915-921

[12] Yiwei Wang, John F. Doherty and Robert E. Von Dyck, Moving Object Tracking in Video, *Proceedings. IEEE Computer Society 2000 29th Applied Image Pattern Recognition Workshop (AIPR 2000),* 16-18 October 2000, Washington, DC, USA,