



This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

CAD SOLID MODELLER EMULATION OF MODULAR MANUFACTURING MACHINES

K. CASE*, C. D. WRIGHT* AND X. T. YAN**

*Loughborough University of Technology, ** Lancaster University, UK

SUMMARY

A method for geometrically modelling and emulating modular machines based on the Universal Machine Control reference architecture is described. Geometric modelling is achieved through extensions to a proprietary robot simulation system. A library of modules consisting of 1, 2 or 3 degree of freedom manipulators is used to construct models of multi-degree of freedom distributed machines with appropriate kinematic characteristics. Logging of data from the real or emulated control handlers is then used to drive the geometric model. A case study of a printed circuit board assembly machine is used to demonstrate how this approach allows the investigation of machine performance before and during the building of the machine with real hardware elements.

INTRODUCTION

Modular manufacturing machines are devices where some overall functionality is achieved by the combination of a number hardware and software elements. The importance of the approach is that the systems created can be readily re-configured or extended to accommodate new products and processes, and the component parts (software as well as hardware) can be re-used. Additional benefits arise from the improved system maintenance and upgrade capability inherent in the modularity.

The modular approach has much to offer, but its successful implementation requires an appropriate control architecture together with effective design and evaluation tools. In the work described in this paper, the control architecture is a central aspect of the Universal Machine Control (UMC) methodology that has been devised within the Manufacturing Systems Integration (MSI) Institute at Loughborough University of Technology. A UMC control system is comprised of a set of concurrent software processes which conform to a reference architecture and provides a means for integrating proprietary mechatronics products from diverse sources. Off-line software tools are provided for the configuration and management of the runtime system. The control architecture uses software handlers which implement communication to and from real machine elements. These can be replaced by emulation handlers which mimic the actions of the real devices and remove the need for them to be present. In the early stages of design the machine may be entirely represented by emulation handlers and this provides an ideal opportunity to develop the logical behaviour without the encumbrance of

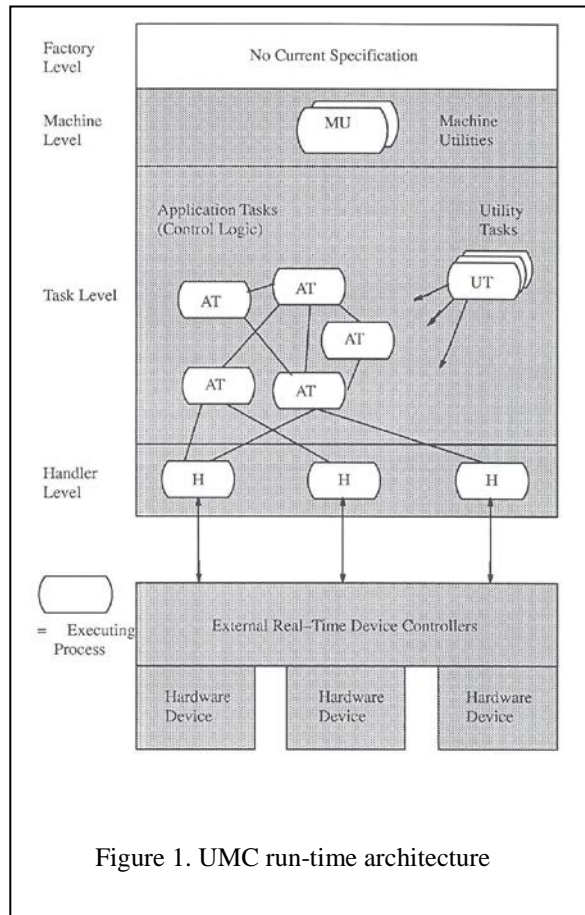
having to build real machines for evaluation purposes. As the design proceeds towards full system evaluation the emulation handlers can be progressively replaced with real handlers (and hence real devices).

Creation of anything other than an extremely simple modular machine requires some form of visualisation that assists in understanding the spatial, temporal and logical aspects of the machine. This has been achieved by the development of a Computer Aided Design (CAD) solid model machine emulator which forms the main focus of this paper. Information from the executing control system can be directly piped to a CAD solid model representation of the physical device, thus providing the required visualisation of the spatial aspects alongside an accurate representation of the temporal and logical constraints imposed by the operating machine.

UNIVERSAL MACHINE CONTROL

The Universal Machine Control (UMC) system consists of a set of concurrent software processes which conform to a reference architecture, together with off-line software tools for configuration and management of the run-time system [1] [2]. Proprietary mechatronic products are known as 'external devices' as they will be controlled by non-UMC means. However their external behaviour can be expressed in UMC terms to provide a unified view of the total machine to the run-time software processes. This is an inherent capability of the control software which communicates with all devices (external and internal) through device handlers. The operational machine includes external devices, internal devices under the direct control of UMC, concurrent system processes and protocols for interprocess co-ordination and management. An

important aspect of this is that distributed machines and associated control systems are naturally and easily handled. (ie. mechanical linkages designed to produce a particular pattern of motion may be replaced by a number of mechanically independent degrees of freedom that are coordinated by the control



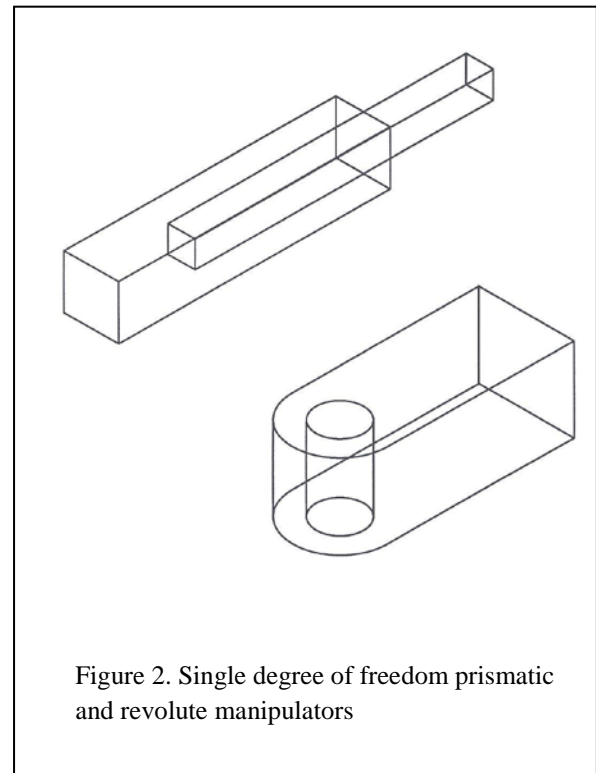
mechanism). Figure 1 illustrates the hierarchical nature of UMC.

The UMC control structure has been established for several years, but this work required extensions to the basic framework to (a) allow execution of a control system without the existence of external mechatronic devices, and (b) the collection of data from an executing control system. The first extension allows control logic development without the hindrance of hardware related issues, thus leading to a safer, less costly method of prototyping machine design. The second extension allows the investigation of many machine performance parameters and forms the basis of the machine emulation.

MANIPULATOR MODELLER

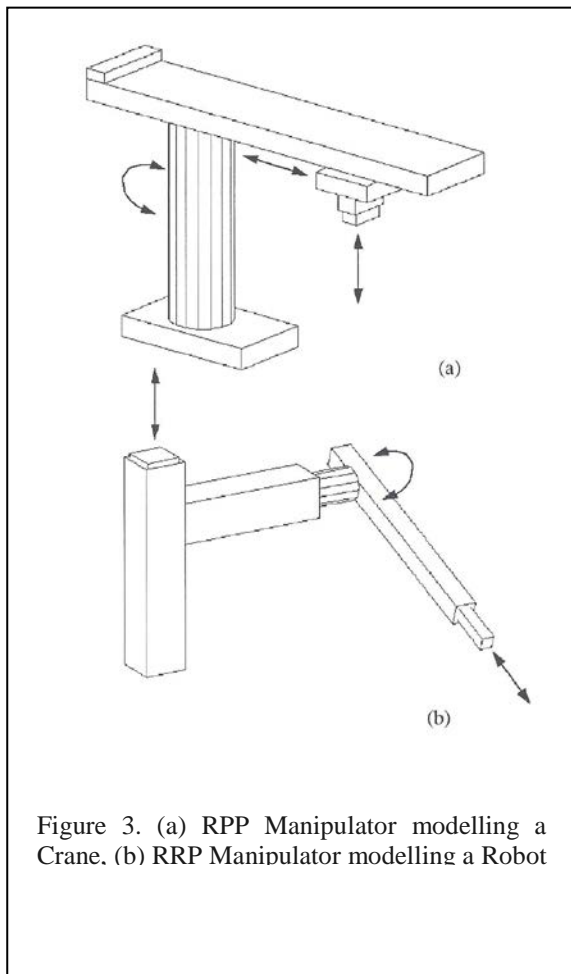
General purpose mechanism design packages are available and are seen to be extremely useful for the mechanical aspects of initial design, but have more limited use in mechatronic products. The modelling

requirements of specific machine structures such as those found in robotics are well provided for (see surveys in [3] and [4]). These robot simulation systems exhibit many of the kinematic and some of the control characteristics required for modular machine design, but are related specifically to the limited range of kinematic structures found in industrial robots.



The approach adopted in this work has been to enhance and extend a proprietary robot modeller (GRASP [5]) to accommodate variably configured machine elements and to provide an emulation using real data collected from an operating machine. A library of mechatronic elements is provided (for detail see [6] and [7]) and facilities are available to assist in configuring these into groups to represent the functionality of modular machines. Each library element represents a 1, 2 or 3 degree of freedom serially linked manipulator. Each degree of freedom can be either revolute (R) or prismatic (P) and is geometrically and graphically represented as a base part and a moving part. Figure 2 shows simple single degree of freedom prismatic and revolute modules. Arbitrarily complex geometry (within the domain of the underlying solid modeller) can be used to replace the symbolic representations shown for situations which require more precise modelling.

Each single degree of freedom is assigned a configurable set of kinematic characteristics including maximum and minimum positions and maximum velocity and acceleration. Individual modules can be chained together to form serial chains of up to three degrees of freedom and by suitable parameter substitution can be made to represent commercially available mechanisms. (Some examples of three-degree of freedom modules are shown as figure 3). If multi-module manipulators are to be simulated (rather than emulated by data collected from an actual machine) then predictive algorithms are required which derive joint extensions by inverse transformation techniques [8].



All combinations of two degree of freedom configurations for manipulators (with joint axes either mutually parallel or perpendicular) have been analysed [6]. Figure 4 illustrates the seven distinct combinations of two degrees of freedom formed from prismatic and revolute joints together with their working envelopes.

Three degree of freedom configurations have been previously studied [8] in the field of robotics. Although there are 36 possible combinations of

revolute and prismatic joints in an open chain link mechanism (with joint axes either mutually parallel or perpendicular), only 12 of these are distinct. The remaining configurations are either redundant or degenerate to one of the twelve basic classes. Of the remaining twelve only four are commonly used in robot structures - these being the ones that have the required characteristics of accuracy, working envelope, etc.

These predefined configurations of 1, 2 and 3 degree of freedom open chain link systems form an important aspect of the library facilities for the building of modular machines. Substitution of suitable kinematic and dimensional parameters enables most commonly encountered situations to be modelled.

The effective control of models of multi-degree of freedom mechanisms requires a suitable data structure within the CAD system. Open chain link systems of the type described are inherently hierarchical and thus it is natural to consider a hierarchical data structure for the representation. This was a primary reason for selecting the GRASP robot modelling system as a starting point as it has a tree structure implemented in a hierarchical ring based data structure. Figure 5 shows a fragment of a data structure developed for this purpose where each module has a data structure that allows the separation of spatial, geometric and kinematic parameters. This is particularly suited to the parameter substitution that forms a central part of any modular design system. Linking such sub-structures within a tree establishes the ownership relationships between modules and facilitates the evaluation of forward transformations to determine a manipulator configuration from a set of joint extensions and of inverse transformation to predict the joint extensions required to attain a specified point within the working envelope.

MACHINE EMULATION

In the context of this work, *emulation* refers to a situation where the abstraction of graphical and solid modelling is enhanced by the use of machine operation data captured from an executing control system (which may be controlling real or emulated hardware handlers). This is in contrast to a *simulation* system where all information would be synthesized from algorithms, heuristics, etc. Hence emulation can be an important tool for studying the detailed performance of machines and their associated control systems.

The capture of control system information is described more fully elsewhere [7], but is achieved through the use of information modules which are common data areas used for communication with

handler processes. Action requests from the application processes can be intercepted at the information module and appended to a serial datastore for subsequent loading into the Manipulator Modeller.

CASE STUDY

The Institute's modular testbed has been used to demonstrate the UMC control system in a printed circuit board component insertion application (see CAD model, figure 6). The hardware consists of two multi-degree of freedom manipulators and three single degree of freedom devices giving a total of 10 distributed degrees of freedom. Circuit boards arrive at the assembly station by a conveyor (not shown) and the operation is to insert a component into each board and to periodically eject a box of completed boards from the back of the cell. The assembly process can be briefly described as (i) the box of incoming boards is picked up by the linear axis (4) and moved towards the gantry (1); (ii) the gantry gripper moves onto a board; (iii) the transport cylinder (5) extracts a board from the box and places it between the gripper jaws; (iv) The gantry moves the board to the linear axis with the table (3) and locates it on the table; (v) The linear

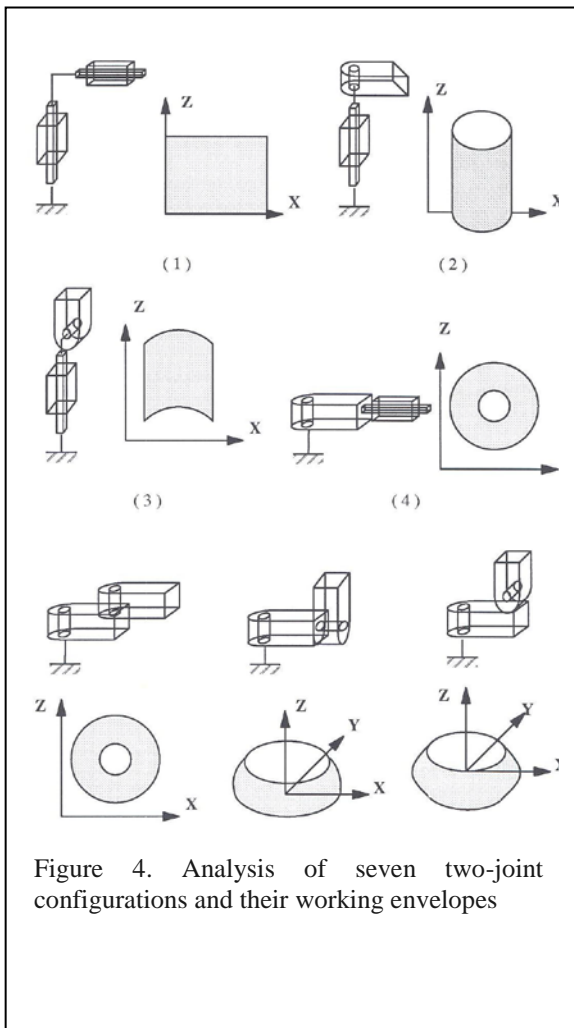


Figure 4. Analysis of seven two-joint configurations and their working envelopes

axis (3) positions the board under the robot arm (2) which inserts a component; (vi) the board is returned to the box by reversing the previous actions; (vii) the process is repeated for all boards.

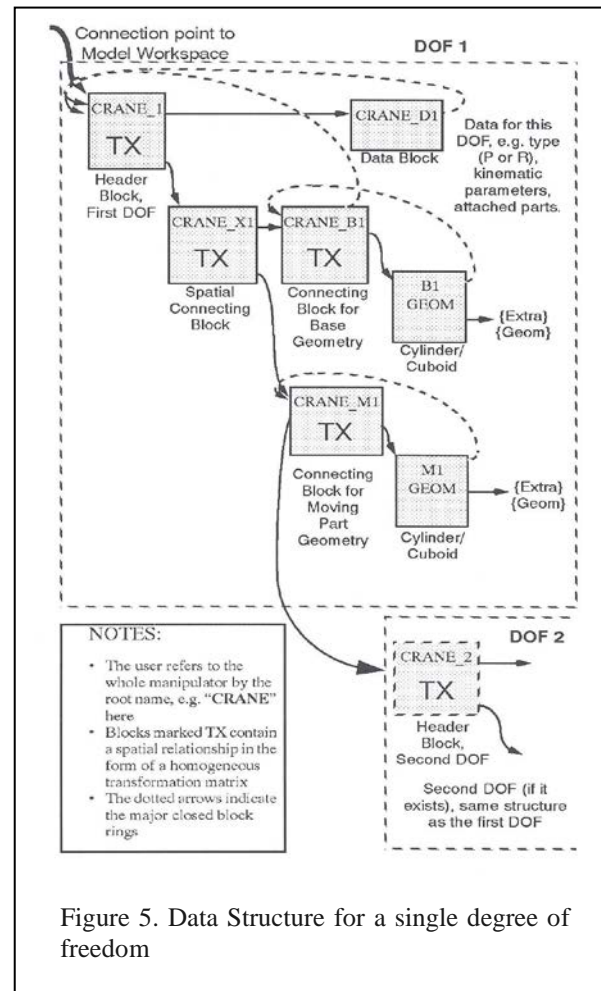


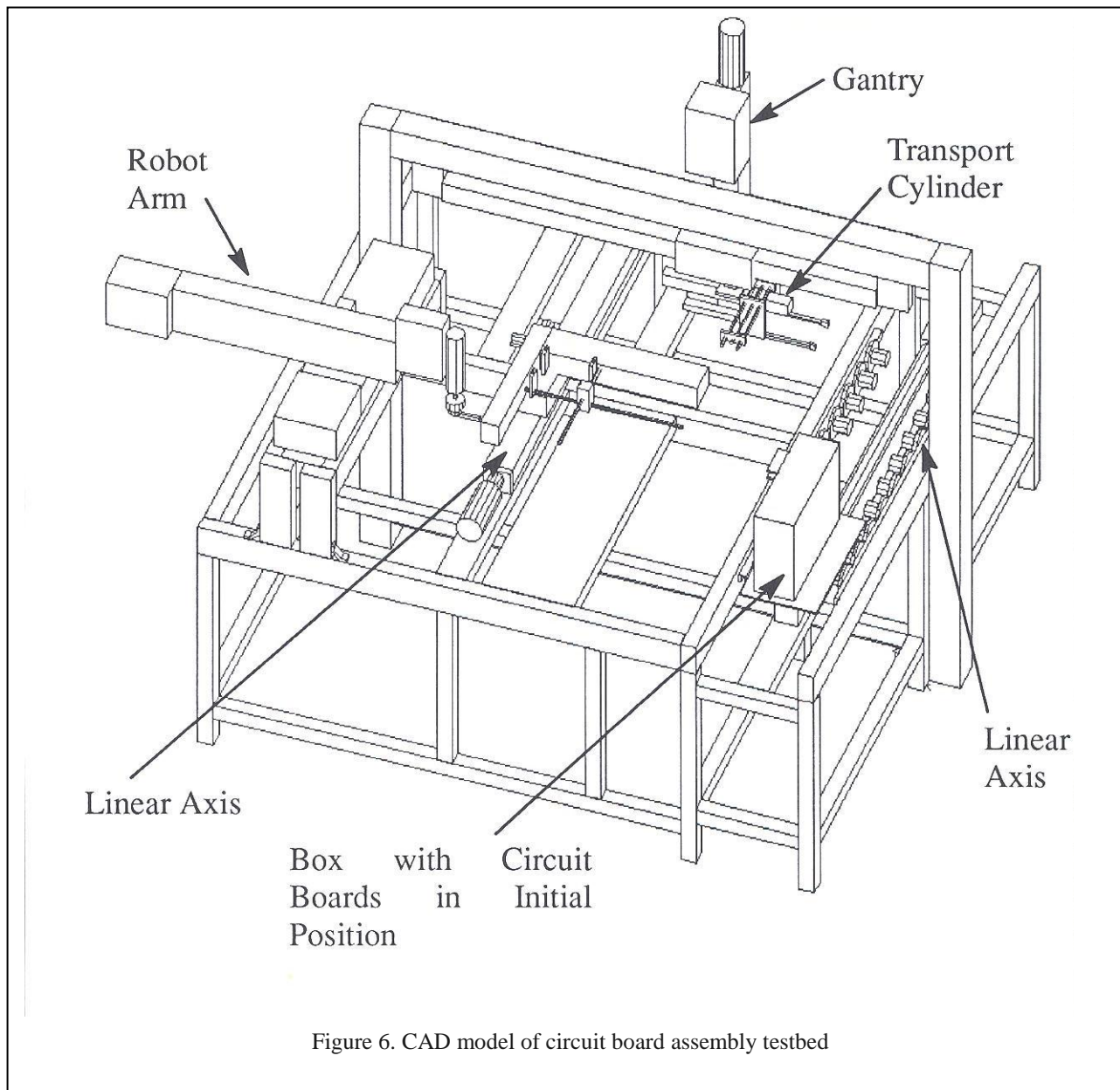
Figure 5. Data Structure for a single degree of freedom

The CAD model (figure 6) is first created either by using library items or by generating new models to represent specific hardware. The application tasks representing the required logical operations as outlined above are then created (sec [9] for details).

These application processes are then used with a set of emulation handlers to execute the machine and to generate data logs which are essentially graphs of displacement, velocity and acceleration against time for each degree of freedom. The data logs are then loaded into the Manipulator Model and assigned to

the appropriate degrees of freedom. The model can then be run in continuous or single step mode at a chosen time frame interval. If errors in the application logic are found then the tasks can be changed, recompiled and re-executed to generate new data logs. Once a suitable emulation has been achieved the

emulation handlers can be progressively replaced with real (data-logging) handlers and the control system executed using some or all of the real hardware. Fine-tuning can now take place to account for any delays introduced by the hardware.



IMPLEMENTATION

UMC currently runs on the Motorola 68xxx family of processors under the OS-9 operating system. The CAD modeler is based on an 'open' version of the GRASP robot modeler which has been extended to accommodate frequent changes in product or process, and the re-

configurable nature of the system described makes this readily achievable without the use of expensive hardware prototypes. Ongoing research is investigating the driving of the (CAD) model directly from the executing control system (rather than off-line as at present).

ACKNOWLEDGEMENTS

The research described in this paper has been funded by the Applications of Computers in Manufacturing Engineering (ACME) directorate of the Science and Engineering Research Council (now the Engineering and Physical Sciences Research Council (EPSRC), and by the SERC/DTI High Speed Machinery LINK initiative. The authors wish to acknowledge the important contributions to the work on UMC concepts and implementation from the following members of the MSI Research Institute at Loughborough University: Prof. R.H.Weston, Prof. P.R.Moore, Dr R.Hanison, A.H.Booth, A.J.Carrott and N.A.Armstrong.

REFERENCES

1. R.H. WESTON, R. HARRISON, A.H. BOOTH and P.R. MOORE (1989) A new concept in machine control. *J. Computer Integrated Manufacturing Systems* 2(2), p.115-122
2. R. HARRISON (1991) A generalised approach to machine control. Ph.D. Thesis, Loughborough University of Technology.
3. X.T. YAN (1992) Graphical modelling of modular machines. Ph.D. Thesis, Loughborough University of Technology.
4. S.F. CHAN (1989) Advancement in robot programming with specific reference to graphical methods. Ph.D. Thesis, Loughborough University of Technology.
5. Y.F. YONG, R.J. MARSHALL, and M.C. BONNEY, (1988) Using CAD to Plan and Evaluate a Robotic Cell, *in Control and Programming in Advanced Manufacturing*. Edited by K. RATHMILL, Springer-Verlag, p. 235-247.
6. X.T. YAN, K.CASE and R.H. WESTON, (1994) A generalized approach to the modelling of modular machines, *Proceedings of the Institution of Mechanical Engineers*, 208, p. 191-203.
7. C.D. WRIGHT and K. CASE, (1994) Emulation of modular manufacturing machines using CAD modelling, *Mechatronics*, 4 (7), p. 713-735.
8. V. MILENKOVIC and B. HUANG, (1983) Kinematics of major robot linkages. *Proceedings of Thirteenth International Symposium on Industrial robots*, 7 (2), Chicago, USA.
9. A.H. BOOTH and A.J. CARROTT, (1993) UMC Version 3.0 reference manual, MSI Research Institute, Loughborough University of Technology.