

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/>

# A generalized approach to the modelling of modular machines

X T Yan,\* BEng, PhD, K .Case, BSc, PhD, CEng, FEngs, FBCS and R H Weston, BSc, PhD, CEng, FIEE  
Department of Manufacturing Engineering, Loughborough University of Technology, Leicestershire

*This paper describes a method of graphically simulating modular machines within a computer aided design environment. This forms part of a much larger Science and Engineering Research Council (SERC)funded programme aimed at advancing modern practices when designing and building manufacturing machines.*

*A generalized approach to the synthesis of the generic features of various kinematic motion pairs is presented and prismatic and revolute motion primitives generalized in their functional and geometric aspects. A hierarchical ring and tree data structure has been designed and implemented to comprehensively represent these motion pairs and to simulate their performance. More complex modular manufacturing machines can be represented using information from a library of up to three degree of freedom motion modules. Seven two degree of freedom motion primitives and twelve three degree of freedom motion primitives with articulation configurations have been analysed and included in the motion primitive library. The configuration of modular machines comprised of physically separate but logically connected distributed motion primitives are described. Examples of a two-finger industrial robot gripper and a three-finger industrial robot hand are used to demonstrate the general principles.*

## 1 INTRODUCTION

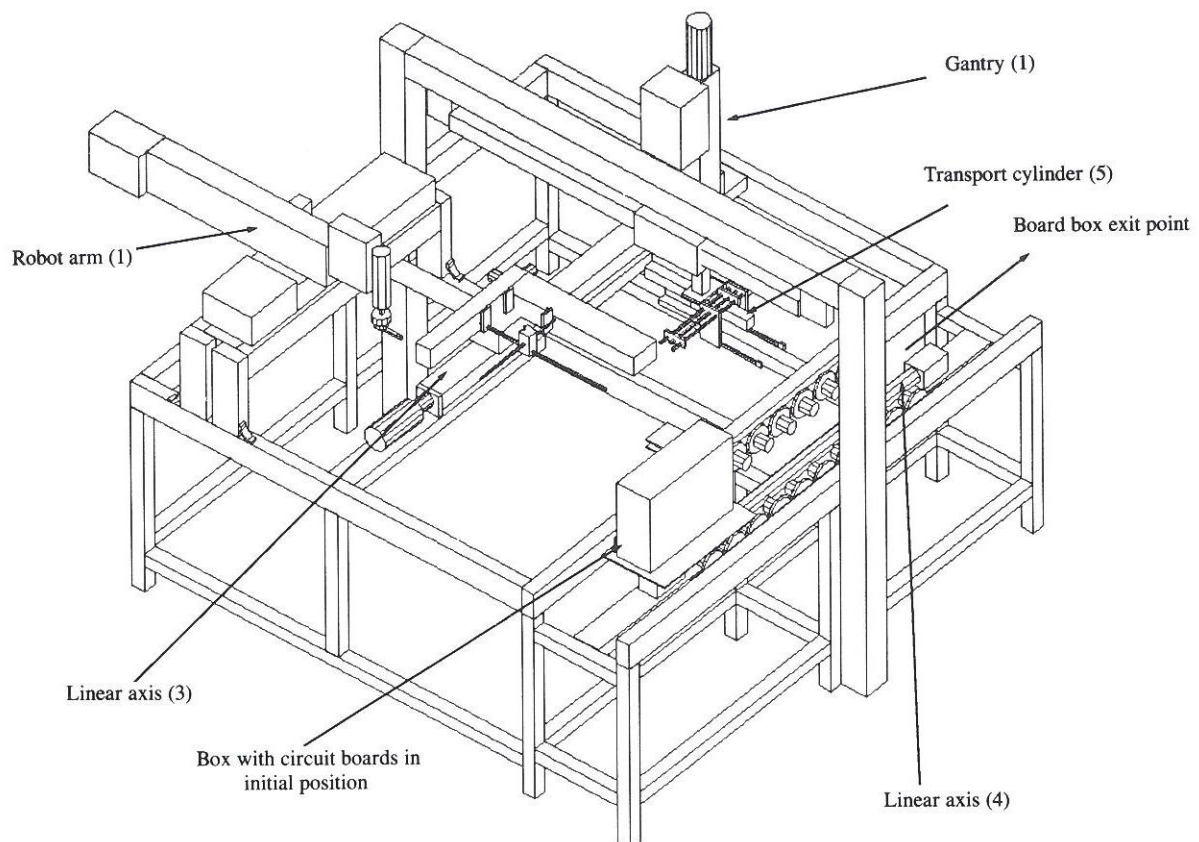
Rapidly changing market conditions are causing automated machines to be used in truly flexible ways within integrated computer controlled manufacturing environments. This introduces a requirement to formulate an integrated approach to the design of both the control software and the physical configuration of machines. Traditional practice in automated machinery design involves customization of a machine and its control system to satisfy specific requirements. More recently, industrial robots have provided increased flexibility, but the fixed configurations adopted often result in far from optimized performance capabilities yet redundant capability, with consequent high costs. The modular approach has been widely adopted by machine builders and as much as possible designers utilize well proven and readily available mechanical and control system components to construct a new machine. However, the ability to select machine building elements, configure them into a wider structure and simulate overall performance of the machine is restricted by a lack of computer tools. This forces designers to consider the design issues at the low level of components rather than at the machine system level. This results in a tendency to constrain solutions to a small number of machine configurations which although well proven do not lead to near-optimal operational performance in terms of cycle times, accuracy and cost. Clearly, therefore, great benefit would be gained from the availability of methods and tools to support a more comprehensive range of well-trying and tested configurations.

The Modular System (MS) Research Group at Loughborough University of Technology has worked with SERC funding towards establishing an integrated methodology for designing and controlling manufacturing machines that can reduce associated life cycle

costs and improve their effective industrial use. The methods are based on the use of graphical and other modelling techniques and the notion of 'enacting' design models so as to facilitate the practical implementation and application of a machine specified and modelled during design processes. The approach brings together interdisciplinary tools to support the design of computational control systems and electromechanical elements of machines. Importantly, the tools provided are based on a consistent set of models of machine primitives (or building blocks) where the models provide means of both simulating and emulating machine performance. As part of the methodology derivatives of the design models are enacted during runtime to drive the manufacturing machine. The tools provided by the MS Group are being evaluated through designing and building a variety of machines for materials handling, packaging and printing.

Thus an important part of the MS Group's approach concerns the derivation of a computer aided simulation tool for modular machine design and machine control program verification, which can lead to model enactment during other life cycle phases of machines. The work reported in this paper describes first-generation solutions which meet that requirement in the form of a computer simulation tool.

Literature surveys in the area of robotic modelling, simulation and off-line programming conducted by Chan (1) and Yan (2) confirm that a wide range of techniques and tools are readily available which can handle workplace modelling and simulation issues for specific robot structures. However, the work to date cannot flexibly model and simulate more general manufacturing machines, including generalized modular machine structures. The underlying techniques do provide a starting point for research which can facilitate the design of more general forms of kinematic structure. Thus further research investigation was required to advance modelling and simulation technologies in a manner that (a) satisfies the requirements of modelling



**Fig.1** A modular machine for the assembly of printed circuit boards

and simulating more general modular manufacturing machines and (b) can lead to the creation of an integrated environment for machine design and control. Figure 1 illustrates just one example target application area for the approach, this being a test rig for the assembly of printed circuit boards which is used by MS Group members for the evaluation and alteration to their machine design and implementation tools. This paper emphasizes key design criteria which underpin aspects of the computer simulation tool produced and describes the parameterization of machine building elements and their computer representation.

## 2 MODULAR MACHINE PRIMITIVES

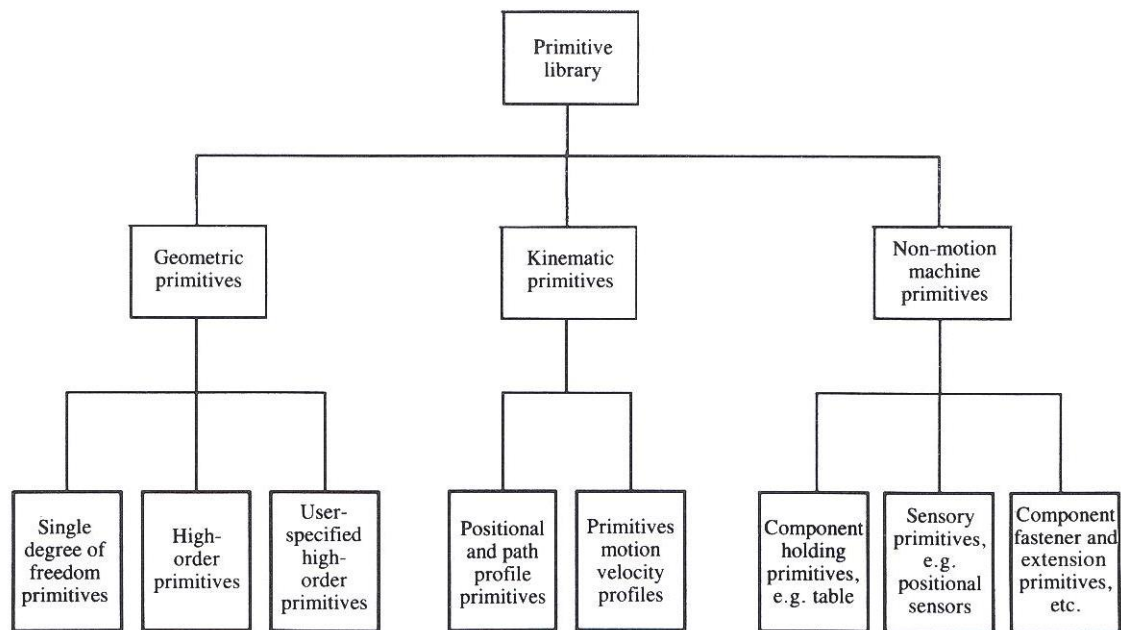
### 2.1 Library primitives

Various industrial machines have been designed to automate (or semi-automate) a spectrum of manufacturing operations. These machines demonstrate a variety of properties but at the same time they exhibit distinct similarities. Hence it is generally accepted that different classes of manufacturing machine can be decomposed into sub-systems which themselves can be further decomposed into machine primitives or modules (3, 4) which individually or collectively can give rise to those properties. For example, a typical machine constituent could be as simple as a single degree of freedom prismatic or revolute joint, or alternatively could implement the functionality of a gearbox or a cam-follower. Conceptually, and as described later, 'modules' providing a single degree of freedom can be considered as low-order primitives. Alternative decompositions might lead

to primitives (or modules) of a higher order which provide motion in more than one degree of freedom. The machine elements or primitives obtained from a physical machine decomposition can be further analysed and generalized according to their features. In this paper, the kinematics of a machine and its primitives are considered to be critical, and machine primitives are classified according to physical shape and kinematic motion type. Non-motion elements which are commonly used in machine building are also considered to facilitate the modelling of modular machines. Parameterization of these various machine primitives enables classes (or families) to be represented in a consistent form. The establishment of such a library of parameterized machine primitives can reduce design effort by standardizing procedures for building machines. Established machine primitives can then be reused by machine builders and new modules added to the library as needs are identified.

The library of machine modules can be viewed as a collection of:

1. Geometric primitives. These are an idealized representation of low- and high-order modular building elements and provide the motion elements that collectively form the moving and base parts of a machine configuration.
2. Motion kinematic primitives. These represent the types of motion provided by a modular machine and its constituents. Spatial path, velocity and acceleration characteristics are defined.
3. Non-motion primitives. These are typically non-powered accessory devices. They are used in conjunc-



**Fig. 2** The hierarchical structure of the primitive library

tion with motion primitives to accomplish tasks such as the storage of components in a magazine.

The library implemented includes examples of these three main types of modular machine building elements within the classification hierarchy shown in Fig. 2.

## 2.2 Synthesis of kinematic motions

Geometric primitives are derived from an analysis of kinematic motion pairs. Kinematic pairs with a surface contact are known as lower pairs and include translational joints, revolute joints and their combinations. Kinematic pairs with point or line contact are known as higher pairs and are typified by gears and cams (5, 6). It has been shown that there are no more than six lower pairs, namely prismatic, revolute, screw, cylindrical, planar and spherical pairs (7). A close study also shows that the constituent motion freedom of the last four motion pairs appears as a combination of the single degree of freedom revolute and prismatic pairs (8). Therefore it will be assumed that the functionality of these four lower pairs can be replaced by an appropriate combination of prismatic and revolute joints.

The benefit of this simplification is that every single degree of freedom building element can be controlled in a modular fashion, and this eases the problems of coordinating combined motions of this type. It might be noted that decomposition of a mechanical mechanism and the associated control system (which itself may be a decomposition of a higher order control system) can lead to a module which is referred to as a mechatronic unit. Since there is less complexity in direct joint control, the substitution of a mechanism with more than one degree of freedom by a combination of directly controlled prismatic and revolute joints can provide more flexibility and higher precision. On the other hand, inter-processor delay may ultimately limit the working of several directly controlled single degree of freedom joints when they need to work in a closely

coordinated manner. High-order pairs such as cam-follower and gearboxes can also be replaced by appropriate prismatic and revolute joints associated with a flexible controller, as discussed in Section 3.2.

With the above brief analysis of motion pairs, it is shown that from the kinematic modelling point of view, prismatic and revolute motion pairs provide essential and sufficient machine building elements for the modelling of computer-controlled modular machines. The modelling of other kinematic requirements and motion types can be addressed by using either single prismatic or revolute joints or their combination. The separation of kinematic modelling and control issues generalizes the modelling of kinematic motion pairs as two types of motion pairs and allows the consistent use of control algorithms throughout the life cycle of machine modelling and control. This is extremely useful in facilitating the integration of computer-controlled machinery within manufacturing systems.

## 3 SINGLE DEGREE OF FREEDOM GEOMETRIC PRIMITIVES

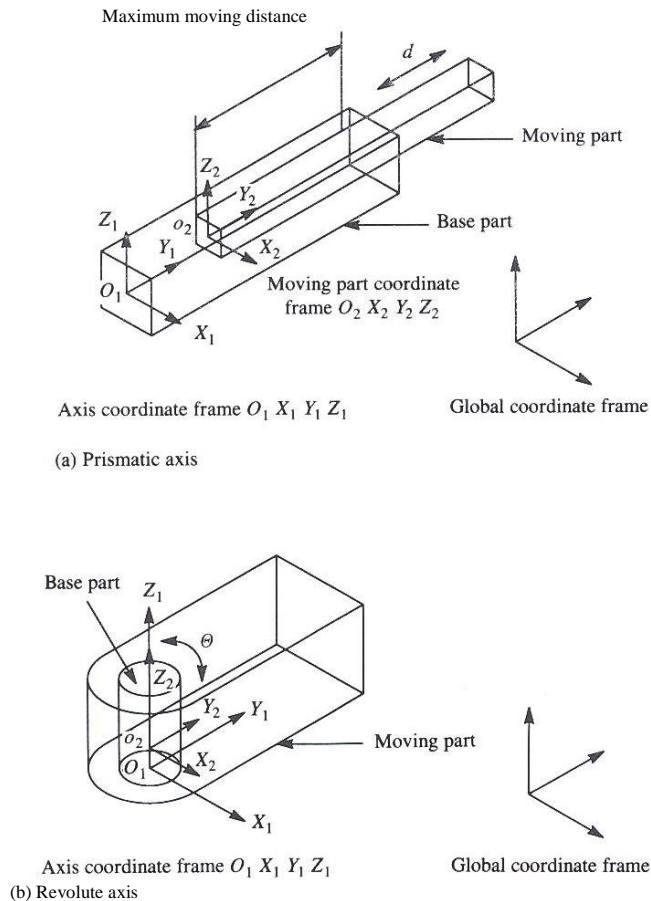
### 3.1 Synthesis of lower motion pair primitives

In computer modelling, a generalized prismatic joint (Fig. 3a) is characterized by (a) its location and orientation in a global coordinate frame, (b) the position of the base and moving parts relative to the global frame, (c) the geometry of the moving and base parts and (d) the kinematic constraints and axis manipulation information.

This generalized approach provides the flexibility in modelling different varieties of the same type of prismatic joint, as the relative position of two local coordinate frames can be arbitrarily defined. This axis coordinate frame also determines the kinematic structure of an axis, and hence the structure will be maintained even if the geometry to represent a part of the axis is missing. This is a very useful simplification feature and allows the hiding of some trivial geometries when the model becomes very complex.

For revolute joints, the same coordinate frame system can be applied. In order to locate the axis origin at an





**Fig.3** Coordinate frame systems of different kinds of axis

appropriate position and orientation, the axis geometries need to be arranged according to the changes of motion type and the origin of the axis geometry. However, the same general axis structure remains (see Fig. 3b). A one degree of freedom rotation around the  $Z_1$  axis in the local coordinate frame  $O_1 X_1 Y_1 Z_1$  is allowed.

### 3.2 Analysis and degeneration of higher pair motion primitives

Although lower pairs have the capability to withstand considerable applied loads due to their surface contact, higher pairs are sometimes indispensable and still find many applications within traditional machine design (8).

A typical example of a higher pair mechanism is a cam-follower, which is traditionally the simplest means of achieving a complicated displacement profile with respect to some variable (commonly time). As an alternative to a cam driver, a computer-controlled actuator can drive a prismatic joint controlling movement to generate a given time and displacement profile. In this way a higher pair (cam-follower) can be replaced by a lower pair (prismatic joint) both in a real machine as well as in the simulation. This replacement of cam drivers has the following advantages:

1. Computer-based controllers can change the 'cam tour' (that is the relationship between time and displacement) flexibly and easily under software

control, and this can significantly reduce the cost and leadtime to produce a new physical 'cam-follower'. Furthermore, the use of programmable transmission elements of this type can lead to less downtime during product changes on manufacturing machines (9).

2. Wear and lubrication problems are reduced by the use of fewer mechanical parts.
3. It is easier to maintain a software cam and reliability is improved as there is no line or point contact (9).
4. Since there is no restriction on the rise and fall profiles, software cams provide a wider range of choices, even for more complicated transformations.

The gearbox is another type of conventional higher pair mechanism, transmitting power to individual drive-shafts at various speeds. The use of gearboxes for transmission is based on the assumption that the size, shape or the handling requirements of the product range to be processed by a particular machine throughout its lifetime are known (8, 9). However, with reducing product life cycles and an increasing pressure to minimize product costs, the use of a gearbox transmission system can become expensive due to its inflexibility in coping with either faster throughput or an entirely new product. The expectations for a new generation of production machines able to flexibly adjust to changing requirements have encouraged researchers to derive a family of 'intelligent' controllers and drivers (10). Software gearboxes can achieve the necessary transformation between a displacement (measured by an encoder, which is a replacement for the input pinion of a gearbox) and the position of an output shaft of a servomotor. Furthermore, the capacity to store different position relationships between the input device and the output shaft can lead to much increased flexibility. Kinematically this simplifies the gearbox into a set of revolute joints rotating in a synchronized and co-ordinated way.

## 4 COMPUTER GEOMETRIC REPRESENTATION OF MOTION PRIMITIVES

Since all constituents of a machine have a physical manifestation, a graphical description of the geometry is a commonly used approach in machine simulation (11). Similarity in shape between the real and modelled geometry of a physical machine can enhance the visualization and identification of static and kinematic behaviour. In some circumstances, such as collision detection, the precise geometric representation of a machine and its working environment may be an essential feature of performance evaluation.

Generally, a joint can be represented by two entities which model the moving and base parts of an axis. Two single pieces of geometry can only statically represent a frame of an axis. A complete axis includes these two pieces of geometry together with a coordinate frame which establishes the relationship between the two items and the axis to form a motion pair. In order to place an axis within its working environment further information is required to relate the local axis frame and the global frame of the machine modelling environment.

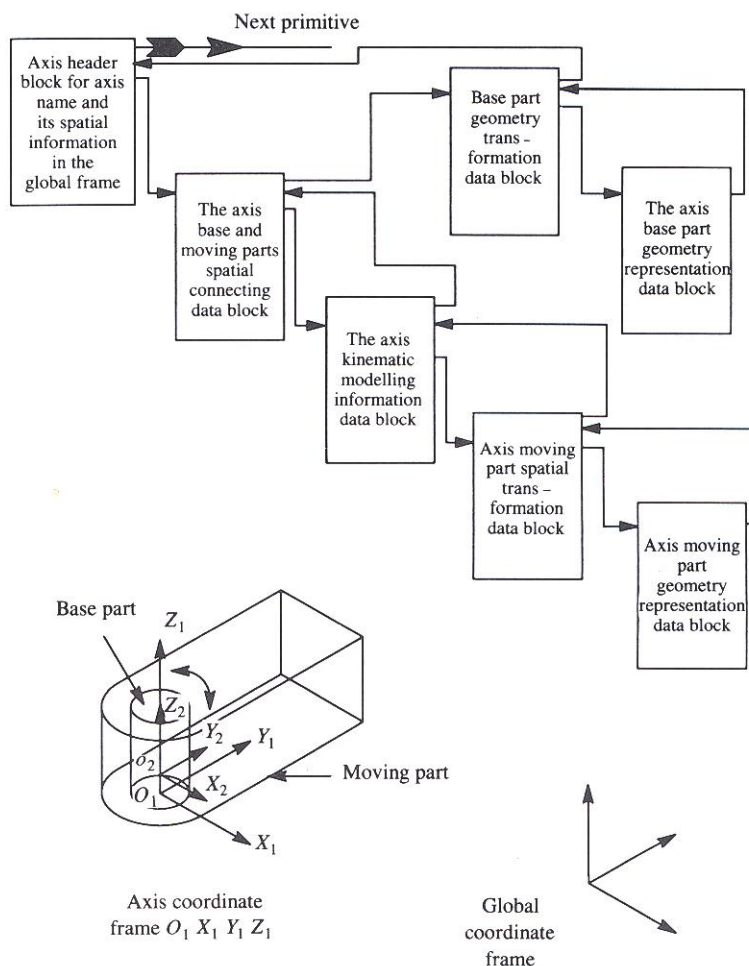
The completeness of the geometric representation of a machine environment will determine the accuracy of the

simulation, but it will conflict with the need for efficient simulation. An axis can be simply represented by two single pieces of geometry and five coordinate frames and be displayed on a screen as two compound geometries and related coordinate frames. The compound geometries are the results of addition and subtraction of several primitive geometries. In terms of geometric representation, the base geometry (which is a part of the axis basic geometry) owns the other geometries which are added to the base geometry, in the same way as the physical joint constituents can be assembled on to a physical joint base. The extending end of a compound geometry is open and any number of different primitive geometries can be owned by the base of the compound geometry, but this will typically result in inferior simulation speed, especially when graphic animation is involved.

In order to clarify a model visually, part of an axis geometry can also be dummy (in the sense that it is not modelled graphically) to facilitate the visualization and understanding of complex configurations at higher speed. By using a simplified representational model of any group of axes it becomes easier to animate a machine; however, this will be at the expense of less accurate modelling. A trade-off between the modelling accuracy and simulation efficiency must be made, although with the availability of increasingly powerful computer systems more effective tools will continue to emerge.

As discussed earlier, most machine kinematic pairs can be replaced by a computer-controlled prismatic or revolute axis. It is therefore of fundamental importance to establish a common, inclusive and flexible data representation to generalize the computerized modelling and simulation of geometric axes. This data structure should be inclusive (in the sense of completeness of joint information) so that it can ensure that the data are informative enough for modelling, evaluation and task programming. A common data structure for an axis should comply with the modular methodology, facilitate the manipulation of axes within a machine modelling environment and simplify the modelling of modular machines. The data structure should also be flexible since the simulation of a machine environment requires large amounts of computation and covers a considerable variety of physical joints. Effective data searching of the data structure of a single axis or a group of axes within a complete machine environment has considerable impact on the flexibility and efficiency of the simulation system. The axis information includes dimensional, spatial and kinematic aspects, and these data are arranged into a ring of blocks for each primitive and into a hierarchical tree structure to represent the logical arrangement of the machine.

Analysis of axis structure and data representation leads to a physical joint being described in the data structure as an axis composed of seven basic data blocks (as shown in Fig. 4).



**Fig. 4** A primitive axis and its generalised axis common data structure

## 5 THE DERIVATION OF A LIBRARY OF GEOMETRIC PRIMITIVES

A family of single degree of freedom mechanical modules has been derived based on the use of the data structure depicted in Fig. 4.

Coaxial prismatic axes are available in the library with default shapes of either two cuboids, two cylinders or a combination of one cuboid and one cylinder (see Fig. 5a). Pointers are provided for users to add any sub-shape on to the default geometry so as to improve the realism of the geometric representation. Offset prismatic axes have been parameterized in the library and can be used as a mechanical slide module or a carriage (see Fig. 5b). Coaxial revolute axes have been provided in the library as shown in Fig. 5c, including a commonly used swing type of axis (see Fig. 5d).

A screw-type axis has also been included in the library as an exceptional case of the single degree of freedom modular unit. Graphically it is represented by a revolute axis (see Fig. 5c and d). However, a special driver is needed should linear or rotational motion be a requirement.

## 6 HIGHER ORDER LIBRARY PRIMITIVES

In the majority of manufacturing application areas there is a requirement for motion in three-dimensional space, and consequently it is unusual to find a single degree of freedom unit used in isolation. However, at the other extreme of multi-degree of freedom mechanisms (such as conventional serially chained robots), redundant motion capabilities are often found. Thus although a multi-degree of freedom mechanism may represent a feasible kinematic solution, it will often not represent the best solution because (a) the machine may

be unnecessarily costly, (b) it may demonstrate relatively poor accuracy (and repeatability) and (c) it may involve relatively long cycle times (due to the limited effective power-weight ratio) (12). One approach to machine design is to specifically design a complex machine tailored to manufacturing certain types of component. However, the alternative approach of designing a distributed machine will gain in popularity with the increased availability of modular building elements. This in turn will lead to cheaper solutions, with improved levels of accuracy and repeatability when compared with conventional industrial robots.

The topological properties of contemporary pedestal-mounted industrial robots are dominated by four types: Cartesian (PPP); cylindrical (PPR); spherical or polar coordinate (PRR); and revolute or articulated (RRR) configurations (6, 13) (here P denotes a prismatic axis and R denotes a revolute one), although SCARA (selective compliance arm for robotic assembly) configured robots have also become widely used, particularly for light assembly applications. Three serially chained low-level machine primitives can have extensive reach within a three-dimensional working envelope. This implies that three degree of freedom modular devices should meet many industrial needs. Also it is practical to limit to three the number of articulated joints in a group provided that it is possible to drive a number of groups (of up to three joints) in a distributed way where the resulting relative motions after alternative means of positioning and orienting workpieces, tools, etc. In such a machine system the practical restrictions, such as computing power of the controller and the complexity of kinematic algorithms which impose a limit on the maximum size of any serial chain (and hence on the range of kinematic solutions), is removed (14). If more joints are required another group of axes can be created

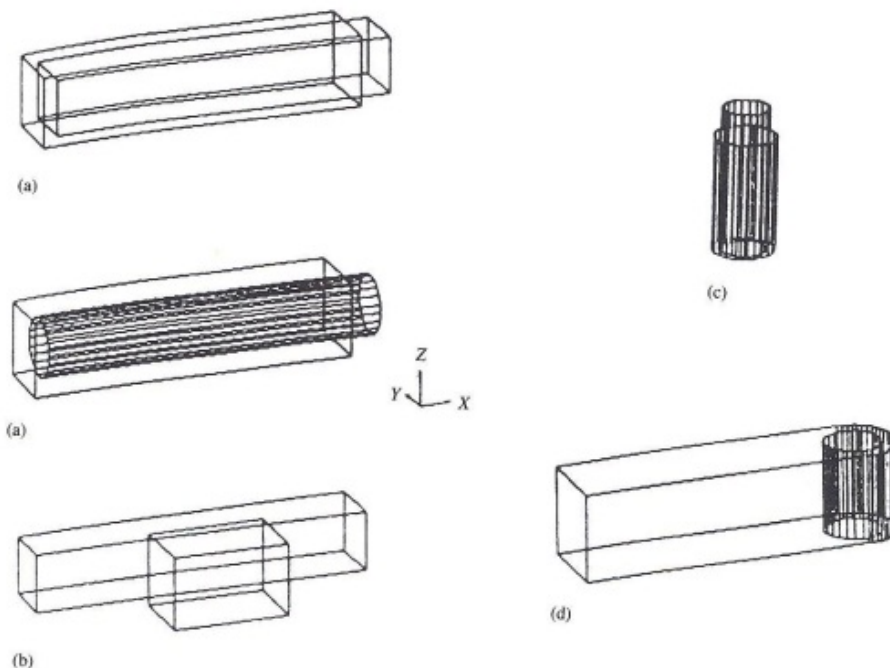


Fig. 5 A family of single degree of freedom modules



via computational means and logically (rather than physically) connected to the previous groups.

In addition, due to the modular approach adopted, tasks requiring motion can be achieved via the concurrent operation of several simple sub-tasks (and associated sub-motions). This results in parallelism which can lead to shorter cycle times (12).

For these reasons it is feasible, and indeed desirable, to limit to three the number of joints in any given group. Mechanical parallelism (that is the use of concurrently operating groups) can then be applied in the configuring of motion mechanisms and in the modelling of a modular machine.

### 6.1 Prismatic and revolute axes combinations

Axis group configurations arise from two approaches: (a) articulated or serially chained configurations of up to three axes and (b) distributed or physically decoupled configurations.

An articulated two-axis configuration is commonly used when building simple manipulators and is often used in industrial robot configurations. The advantage of this type of configuration is that the manipulator has better reach capability (improved dexterity) than other two degree of freedom mechanisms. However, since the base joint has to carry a second (or chained) joint, the moving mass will adversely affect the accuracy (through link deflection, etc.) and the speed of response (the power-weight ratio will be reduced, as will the maximum acceleration of the end effector). All possible combinations of two-axis groups and thus all of their configurations and working envelopes are analysed in Fig. 6.

Milenkovic and Huang (15) have analysed the major combinations of three-joint linkages. Only simple open chains of revolute and prismatic joints were considered with the joint axes either mutually perpendicular or parallel. Among the 36 possible combinations of three joints, there are essentially 12 classes of combination available after discounting redundant configurations and eliminating others through a process of degeneration of degrees of freedom.

However, even for these twelve simple chains only four have found widespread use in industrial robots. In establishing methods of designing machines from serial chains it is useful to study the characteristics of all 12 configurations and to provide corresponding supporting tools to facilitate design processes. Important characteristics include:

- (a) the inherent theoretical accuracy with which specific configurations can be modelled and controlled;
- (b) ease of control which will depend upon the complexity of the kinematic solution for different configurations;
- (c) the working envelope of the configuration;
- (d) speed of a movement of the end point.

Prismatic joints will typically demonstrate better inherent accuracy than their revolute counterparts, particularly when combined with other joints of the same class. In practice, of course, the accuracy achievable will depend on many factors relating to the type of drive used (for example rotational or translational; electric, hydraulic or pneumatic), properties of the power train

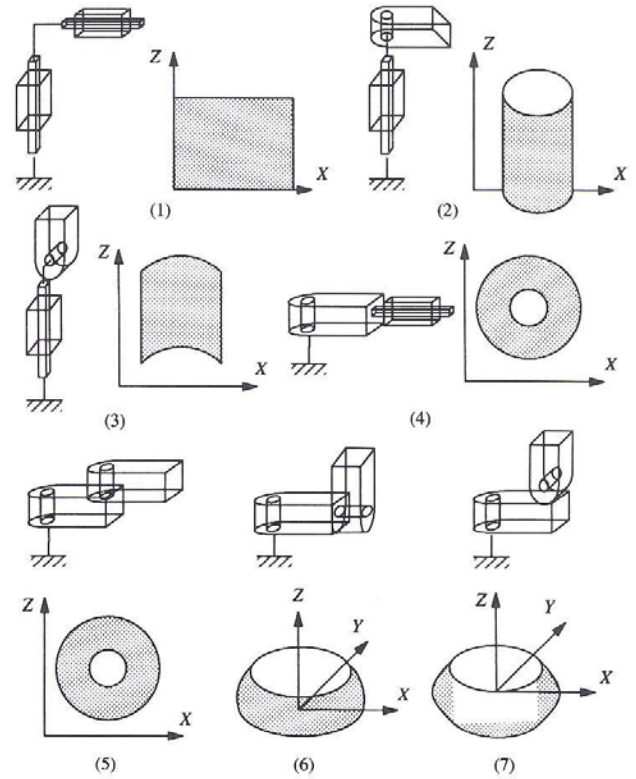


Fig. 6 Analysis of seven possible two-joint configurations and their working envelopes

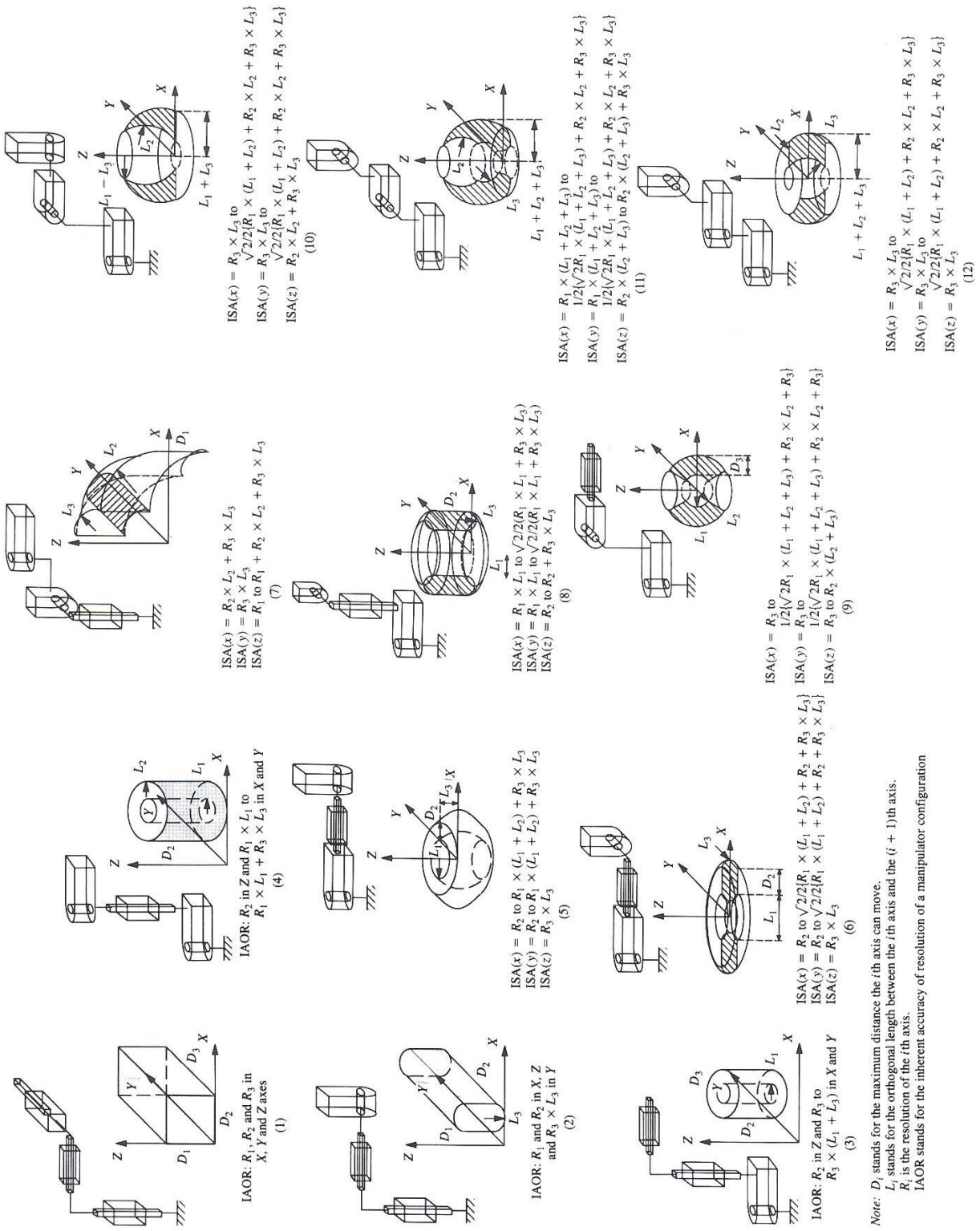
and feedback transmission elements used, properties of the controller and drive amplifier (for example dead-band, nature of control approach and algorithms) and the resolution of the feedback sensors. However, the resolution of a revolute joint will be determined by the product of the joint length and angular resolution and for this reason alone, as the radius arm of the joint is extended, the accuracy with which positioning can be realized will be reduced. This can give rise to practical implementation problems which result in poor inherent accuracy in that degree of freedom.

In addition, since a revolute joint introduces trigonometric functions in the forward kinematic computation, the control of such joints is more complex than for translational joints, particularly when a configuration has a revolute joint as the first in the chain or where there is more than one revolute joint in succession. If a prismatic joint is located in a direction parallel to one axis of a Cartesian coordinate frame, it dramatically simplifies the kinematic computation.

On the other hand, as a revolute joint rotates about its axis, configurations which include rotations will invariably provide greater dexterity than prismatic ones, as they normally rotate within a relatively large envelope. Typically the length of a prismatic joint will be restricted as it will have a linear mating surface requiring the mass of the joint to be restricted. Thus, configurations with revolute joints typically have advantages of large working envelopes and relatively high speed of movement of the arm's tip.

Figure 7 illustrates the 12 valid configurations, their working envelopes and their inherent spatial accuracy





**Fig. 7** An analysis of twelve possible three-joint configurations, their working envelopes and inherent spatial accuracy

in each direction. With respect to spatial properties, on the assumption that factors such as deflection in links can be ignored, the inherent spatial accuracy (ISA) of a multi-axis manipulator system will be some function of the inherent resolution with which links in the manipulator can be controlled. Here that function will depend upon geometric properties of the manipulator system, whereas the inherent resolution will typically be determined by the linear or angular resolution of the position sensing and feedback system (used as part of the controller and drive system for each joint). The resolution for a prismatic joint is expressed as a linear displacement whereas the resolution for a revolute joint is specified as an angular increment, this being converted into an equivalent linear displacement by multiplying the angular resolution by the orthogonal length between this joint and its next adjacent joint (that is the effective link length). Thus the ISA for a high-order (multi-axis group) motion primitive can be determined with reference to spatial functions relating the joints. The ISA for the configurations illustrated in Fig. 7 is exemplified here to demonstrate the principles involved, which were used by the authors to obtain the range of ISA values that occur corresponding to the range of postures that the kinematic structure can adopt.

In Fig. 7, configuration (4),  $R_1$  and  $R_3$  are defined as the angular resolutions (in radians) for the first and third revolute joints respectively, and  $R_2$  is a linear displacement resolution for the second joint (in metres).  $L_1$  is defined as the distance between the rotational axis of the first joint and the sliding axis of the second joint, while  $L_3$  is the distance between the rotating axis of the third joint and its distal point. Since the prismatic joint is associated solely with a linear resolution and movement in the Z direction, the ISA component in that direction will be in direct proportion to the resolution of the second joint (that is  $R_2$ ). However, for each revolute joint, the ISA component introduced by the joint will be proportional to the effective link length multiplied by the angular resolution of that joint, that is  $R_1 \times L_1$  and  $R_3 \times L_3$  respectively. As the three links traverse their working envelope the ISA components of the rotating joints will have varying resolved components in the X and Y directions. For example, suppose that the first joint rotates by  $\alpha$  degrees and the third joint by  $\beta$  degrees; then the ISA components caused by the rotation of the first joint can be expressed as follows:

$$\text{ISA}(x, 1) = R_1 \times L_1 \cos \alpha$$

$$\text{ISA}(y, 1) = R_1 \times L_1 \sin \alpha$$

where  $\text{ISA}(x, 1)$  is the resolved component of the inherent spatial accuracy of the first joint in the X direction, and similarly for Y and Z.

The maximum values of these components are

$$\max[\text{ISA}(x, 1)] = R_1 \times L_1 \quad \text{at } \alpha = 0 \text{ and } 180^\circ$$

$$\max[\text{ISA}(y, 1)] = R_1 \times L_1 \quad \text{at } \alpha = 90 \text{ and } 270^\circ$$

If the third joint rotates by  $\beta$  degrees the components can be expressed as

$$\text{ISA}(x, 3) = R_3 \times L_3 \cos \alpha \cos \beta$$

$$\text{ISA}(y, 3) = R_3 \times L_3 \sin \alpha \sin \beta$$

On considering the rotation of both the first and third joints, the ISAs can be expressed as

$$\text{ISA}(x, 1+3) = R_1 \times (L_1 + L_3) \cos \alpha + R_3 \times L_3 \cos \beta$$

$$\text{ISA}(y, 1+3) = R_1 \times (L_1 + L_3) \cos \alpha + R_3 \times L_3 \sin \beta$$

where  $\text{ISA}(x, 1+3)$  is the combined resolved components of the inherent spatial accuracy of the first and third joints in the X direction.

When  $\alpha = \beta = 0$  or  $180^\circ$  or  $\alpha = \beta = 90$  or  $270^\circ$ , then

$$\text{ISA}(x, 1+3) = \text{ISA}(y, 1+3)$$

$$= R_1 \times (L_1 + L_3) + R_3 \times L_3$$

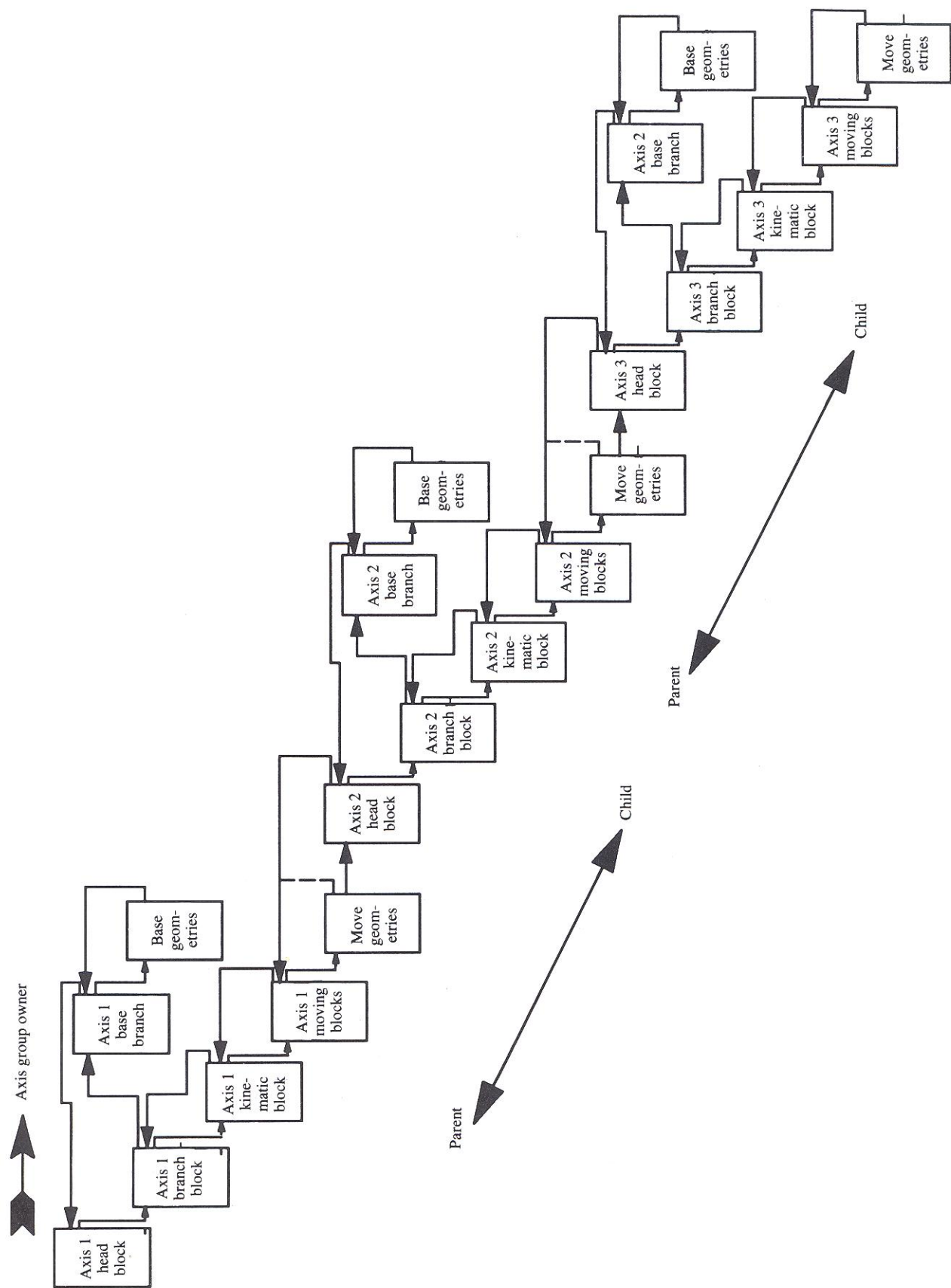
The maximum value and range of the ISA is thus derived. In Fig. 7 only the ISA value and its range for the extreme positions along the X, Y and Z directions are listed. The maximum value of the ISA and its range can be obtained in a similar way for the other configurations.

ISA values can provide very useful information when selecting a high-order motion primitive. Each of these 12 high-order primitives is constructed by successively linking the next constituent joint to the moving part data block of the axis data structure (see Fig. 4) within a ring and tree structure. The axis group is headed by a general data block where all the information concerning the axis group, including ISA values and data for deriving working envelopes, are stored (see Fig. 8).

## 6.2 Distributed configurations

Articulated configurations have been used to automate many types of manufacturing operation, with industrial robots being just one example of the use of the technology. Potentially, however, distributed, but logically coupled, mechanisms can be even more widely applied as they can decompose a complex task into several sub-tasks, possibly accomplishing the whole more simply and quickly. Due to a lack of suitable complementary distributed control system capabilities, this potential has yet to be realized industrially.

On considering the characteristics of possible configurations of distributed machines, each individual device can be a single axis, an axis group or some other form of compound mechanism (for example proprietary devices). Therefore all geometric primitives within the modular machine modelling library can be used as distributed configuration primitives to construct an application-specific configuration. There is no limit to the number of possible distributed machines as far as configuration is concerned, but the performance of a practical distributed machine configuration may suffer from computational limitations related to the software methods and computer hardware. The key issue in adopting a distributed machine configuration then becomes how a designer can be assisted in logically arranging related devices within the machine. Functional decomposition of an ideal machine which satisfies the machine design specification becomes an important criterion in determining the spatial distribution and logical coupling of two or more devices. Three-dimensional motion can very often be decomposed and replaced by two separate relative motions. A robot with an XY- dimensional working envelope can typically be replaced



**Fig. 8** Data structure and axis group relationships

by two single-axis motions which are physically distributed but logically connected to achieve a desired XY spatial path. The methodology for building a distributed manufacturing machine is briefly described here, while Yan (2) provides more detailed information about the control of a distributed machine in a simulation environment.

Co-ordination and synchronization of distributed configurations to achieve some group (or global) functional goal will need to be established by appropriate control of each device, both separately and collectively. A distributed configuration can have its devices arbitrarily placed within its working environment, with electronic or logical coupling between the individual motions. Thus a distributed configuration does not suffer from the same spatial restrictions as serially chained manipulator systems. The logical relationships will determine the global properties of a distributed system.

### 6.3 Aggregation of articulated axis groups

The methods chosen for building an articulated axis group include the two distinct operations of (a) graphical configuration and (b) the establishment of data structure relationships.

Graphical configuration (or aggregation) can be defined as the process of locating each constituent axis at the correct position and orientation. This requires computer assistance or configuration tools which in this project have been built on either graphical manipulation tools or textual/language-based commands. Constituent axes are aggregated to form an axis group. A single axis can easily be created by the selection of a family of primitives from the library and the definition of appropriate parameters. Graphical manipulation at this stage is at the level of the complete axis rather than the separate representations of the base and moving parts, and axes may be added, deleted, scaled, positioned and orientated.

Single axes may then be bound together to form a group, and this results in the establishment of data structure relationships between the individual axes. The data relationships created in this way are of a parent-child nature, and a typical data arrangement of an axis group is illustrated by Fig. 8. Since a child axis is always attached to the moving part of its parent, it is essential to create and maintain the parent-child relationships, and this is achieved by the use of linking pointers.

## 7 BUILDING AN END-EFFECTOR-AN EXAMPLE OF A HIGHER ORDER MECHANISM PRIMITIVE

As an example of building a higher order mechanism by aggregating previously described axis primitives, an analysis of industrial robot end-effectors is outlined and the associated data representations are detailed. Using the methods described here the higher order mechanism can be added to the library of primitive devices to enable re-use as required.

Industrial robot end-effector research has frequently been oriented towards the design of replacements for typical human operator hand functions. Consequently many end-effectors take the shape of a two-fingered

parallel jaw. This type of end-effector possesses the ability to grasp objects in either one or two dimensions of the three translational degrees of freedom in the object's space (16). Another research direction has been towards the creation of industrial robot hands and particular effort has recently been aimed at creating dexterous multi-finger robot hands (17).

Two-fingered gripper configurations are dominated by the rotational and translational types shown in Fig. 9a. Some would consider the ultimate universal gripper to be the human hand, but at present such structures are too complex for industrial use. However, for many applications a three-fingered hand (Fig. 9b) should provide sufficient dexterity (6,16).

The rotational gripper of Fig. 9a consists of two co-ordinated revolute joints, while the translational gripper is composed of two prismatic joints. Both types can be treated as two distributed but logically coupled axis groups for modelling purposes. Two fingers can be modelled graphically by two revolute or two prismatic axis primitives and will typically be owned by the gripper base or wrist. In terms of the data structure, both fingers are children of the base and form a child data ring for the base. A control relationship between the two fingers should be established for manipulation purposes.

A three-fingered hand is much more complex, but by aggregating modules from the library, it is possible to construct such a hand. The three-fingered hand can be composed of three open serial chains and a wrist, where each finger consists of three revolute axes. In terms of

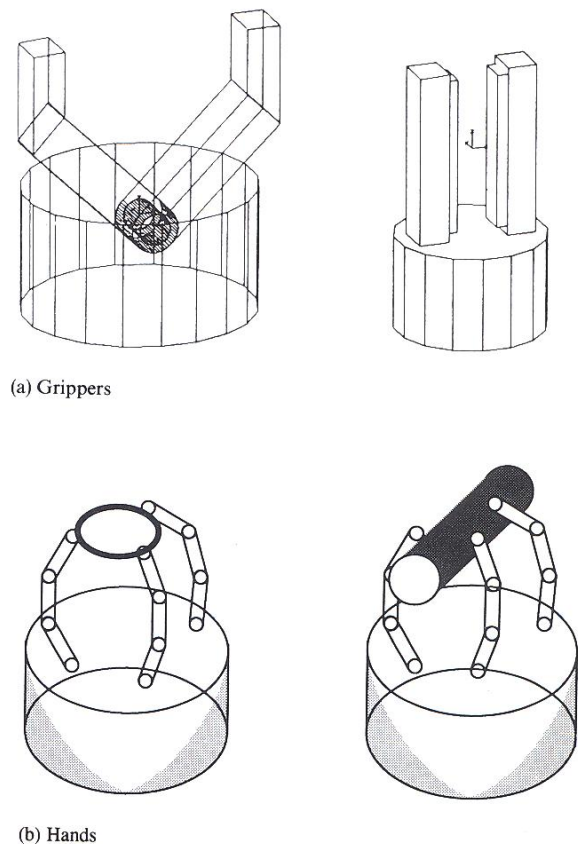


Fig. 9 Modelled industrial robots



the hand configuration, three finger chains are distributed on the wrist. Therefore, such a hand can be modelled by using three three degree of freedom serially chained primitives and configuring them into appropriate positions. In a similar fashion to the two-fingered gripper, the wrist owns three child fingers and a child data ring is formed for the wrist in the internal data structure. However, each finger has a three serial axis chain and each digit or axis can be controlled individually or collectively. The graphical models for a two-fingered gripper and a three-fingered hand are shown as Fig. 9a and b respectively.

## 8 LIBRARY PRIMITIVES AND THEIR MANAGEMENT

The geometric primitives included in the library can be classified into the following types:

- (a) single degree of freedom primitives;
- (b) high-order manipulator primitives;
- (c) user-specified high-order primitives;
- (d) non-motion accessory primitives.

The same types of basic axes with different geometric shapes can be used to distinguish various physical joints. Users are also provided with supporting tools to define their own type of axis should they require some variation from the basic axis representation.

Because of the complexity of the data structures used to describe individual and combinations of library primitives, a library manager is required to ensure their correct and efficient use. The manager automatically creates all primitive constituent data blocks and links them in a specific way within the data structure. Thus the geometric description of a primitive and the assignment of coordinate frames to each geometric entity forms the central issue of primitive creation. The manager also has to ensure that the correct data are assigned to primitives. Since every library primitive has been parameterized, only required meaningful data may be input. It is the manager's task to check the data type and possible value range, provide another chance for input if mistakes are made and finally to assign parameter values when the correct input has been made with appropriate dimensioning applied. As the primitive's data structure is a subordinate of its owner, the manager should call graphical display functions to enable visualization of the primitive on a screen.

## 9 CONCLUSION

This paper has described an approach to modelling and simulating modular manufacturing machines which can be more generally applied for machine modelling and simulation than conventional robot simulation systems. A generalized approach to modelling a modular manufacturing machine has been described based on a synthesis of generic features of various kinematic motion pairs. Prismatic and revolution motion primitives have been generalized and their functional and geometric aspects represented in terms suited to computer modelling. Based on an understanding of the nature of these motion pairs, a hierarchical data structure has been designed and implemented to comprehensively represent them and simulate their performance. This struc-

ture can be used to describe modular machine motion pairs easily and effectively within a modular machine due to its generic and flexible nature. To facilitate the modelling of more complicated modular manufacturing machines, a library of up to three degrees of freedom motion modules has been established. Seven two degree of freedom motion primitives and twelve three degree of freedom motion primitives with articulation configuration have been highlighted and included in the motion primitive library. Distributed motion primitives and configuration were also considered in the paper. Any of these motion primitives in the library can be selected and instantiated by a machine designer to configure a machine easily, and thus the library provides a very flexible facility for modelling modular manufacturing machines. Finally, an example demonstrated how to select some machine building elements from the motion primitive library to construct a two-finger industrial robot gripper and a three-finger industrial robot hand. This example illustrates the power and flexibility gained from an approach that promotes the use of distributed manipulation.

The simulation tool described in this paper provided first-generation solutions to a key area of a wider programme of MS Group study aimed at advancing current practice when producing industrial machines by supporting and integrating lifecycle processes. A second-generation, enhanced version of this modelling and simulation approach has since been developed and is currently being integrated with motion definition tools (which support motion design aspects of software cams, gears and links) as part of a more extensive and comprehensive design environment. In addition the current research focus of the group is on more closely integrating machine design and control processes through model enactment.

## REFERENCES

- 1 **Chan, S. F.** Advancement in robot programming with specific reference to graphical methods. PhD thesis, The Department of Manufacturing Engineering, Loughborough University of Technology, Loughborough, 1989.
- 2 **Yan, X. T.** Graphical modelling of modular machines. PhD Thesis, The Department of Manufacturing Engineering, Loughborough University of Technology, Loughborough, 1992.
- 3 **Benbabib, B., Cohen, R. Upton, M. G. and Dai, M.Q.** Design of a rotary-joint-based modular robot in cams, gears, robot and mechanism design (presented at the 1990 ASME Design Technical Conference), Proceedings of Twenty-first Biennial Mechanisms Conference, Chicago, Illinois, September 1990, pp. 239-243.
- 4 **Tesar, D. and Butler, M. S.** A generalised modular architecture for robot structure. *Trans. ASME, J. Mfg Rev.*, 1989, 2(2), 91-117.
- 5 **Dimarogonas, A. D.** Computer aided machine design, 1988 (Prentice-Hall International (UK)).
- 6 **McCloy, D. and Harris, M.** *Robotics-an introduction*, 1987 (Open University Press, Milton Keynes).
- 7 **Waldron, K. J.** A method of studying joint geometry. *Mechanism Mach. Theory*, 1972, 7, 347-353.
- 8 **Hunt, K. H.** *Kinematic geometry of mechanisms*, 1990 (Oxford University Press).
- 9 **Sinha, P. K.** Transmitting power intelligently. In *Professional Engineering*, April 1990, pp. 52-54 (Mechanical Engineering Publications, London).
- 10 **Quin Systems Limited.** Intelligent motion control for industrial plant automation. An introduction to the principles and application of intelligent control techniques to solve complex velocity, acceleration and position control problems in industrial automation projects. Quin Systems Limited, Wokingham, Berkshire, 1989, pp. 1-11.

- 11 Jayaraman, R. and Levas, A. A** workcell application design environment (WADE). In *CAD based programming for sensory robots* (Ed. R. Ravani), NATO ASI Services, Vol. FSO, 1988, pp. 91-120 (Springer-Verlag, Berlin).
- 12 Weston, R. H., Harrison, R., Booth, A. H. and Moore, P. R.** Universal machine control system primitives for modular distributed manipulator systems. *Int. J. Prod. Res.*, 1989, 27(3), 395-410.
- 13 Tourassis, V. and Ang, Jr, M.** Analysis and design of robotic manipulators with multiple interchangeable wrists. *IEEE Trans. Robotics and Automn*, 1989, 5(2), 223-230.
- 14 Harrison, R.** A description of universal machine control. Internal Report, MS Research Group, The Department of Manufacturing Engineering, University of Technology, Loughborough, 1989.
- 15 Mileokovic, V. and Huang, B.** Kinematics of major robot linkages. Proceedings of Thirteenth International Symposium on *Industrial robots and robots 7*, Vol. 2, Chicago, Illinois, 1983.
- 16 Kato, I.** Mechanical hands illustrated. *Survey Japan*, 1980.
- 17 Li, Z. and Sutry, S.** Task-oriented optimal grasping by multi-fingered robot hands. *IEEE J. of Robotics and Automn*, February 1988, 4(1), 32-44.