Interactive Constraint-Based Assembly Modelling

Mud-Armeen Munlin
Information Technology Department, Sirindhorn International Institute of Technology,
Thammasat University, P.O. Box 22, Thammasat Rangsit Post Office,
Pathumthani 12121, Thailand
Tel : (662) 9869101-8 ext 2003 FAX: (662) 986-9113,
email : amin@siit.tu.ac.th

Abstract

This paper presents a new approach to interactive constraint-based assembly modeling using an intuitive 3D graphical interface. The approach is to integrate 3D interactive graphical techniques such *as automatic constraint recognition, allowable motion,* and *graph-based algorithm* and 3D constraints such as *against*, and *cylindrical fit*. This integration will provide the user with an advanced interactive environment to perform assembly modeling, purely by direct manipulation.

Automatic constraint recognition technique is used to recognize assembly constraints purely from the user's 3D manipulations. A number of additional constraints such as *gear contact, rack-pinion contact, screw fit* and *spherical fit* are supported and can be automatically recognized by this technique. Constraints recognized from the user's 3D manipulations are maintained in a Relationship Graph (RG). The RG maintains the constraint dependency and the allowable motion associated with each constraint to support the constraint propagation to the relevant geometric entities. This frees the user from having to maintain the constraint consistency when the constraints are added or modified and also ensures the interactive response to 3D manipulations of the geometry.

Allowable motion is used to satisfy 3D constraints and simulate the kinematic behavior of an assembly by means of graph-based algorithm. This technique uses the degrees of freedom of objects to derive transformation operations to satisfy a constraint. It is also used to achieve the accurate 3D positioning of a solid model by automatically constraining its 3D manipulations.

Graph-based algorithm reported in this work supports interactive assembly and dragging of under-constrained models. This allows the simulation of the kinematic behavior of an assembly in response to 3D manipulation. This algorithm supports the forward and backward propagation of constraints to simulate the realistic manipulation of assembly models within the virtual environment.

1. Introduction

The recent advent of Virtual Environment (VE) technology has made it feasible to directly interact with objects in 3D space. This offers the potential to develop highly interactive 3D user interfaces for a variety of engineering applications such as virtual manufacturing and assembly modeling [1]. A number of commercial and non-commercial VE toolkits are currently available to support the development of virtual environments for these applications. However, a common weakness of the existing virtual environments is the lack of efficient geometric constraint management facilities. As a result, it is difficult to achieve the accurate 3D positioning of solid models using 3D input devices. Another problem with the assembly modeling application is that most of the current systems employ the menu interaction in order to support the process. This results in a tedious and often non-intuitive process where many menu interactions are required to manipulate a complex assembly. The interactive constraint-based solid modeling approach reported in [2] overcomes this problem by enabling the user to directly interact graphically with the assembly without the use of menus. The purpose of this paper is to extend this approach to realistic engineering assemblies. It will be shown that a combination of high performance graphics systems and management of 3D constraints within the virtual environment provides a new and powerful tool for modeling assemblies.

2. Related Work

This section reviews related work that specifically addresses the degree of freedom analysis for assisting the assembly process and simulating the kinematic behavior of assembly.

The concept of degrees of freedom (DOF) refers to the allowable motions of the rigid body. This concept was originally developed for the analysis and synthesis of mechanism [3] and for planning motion in contact to achieve parts mating. Recently, the degree of freedom analysis has been used in devising constraint satisfaction procedure [4]. In this technique, DOF for a geometric entity are considered as resources that are consumed by moving the geometric entity to satisfy a constraint. Each constraint, upon being satisfied, requires reducing the number of DOFs, further restricting subsequent actions to those that do not violate any previously satisfied constraints.

Morris and Haynes [5] present an assembly the by constraint system that makes use of the degree of freedom between parts and a hierarchical decomposition of the assembly. They identify a set of twenty different constraints in terms of the number of degrees of freedom constrained by one part upon another. Some of these constraints are among the lower kinematic pairs [6], whereas the others are assembly dimensions.

Kim and Lee [3] have developed a system for kinematics analysis by extending the systems reported in [7]. In this system an assembly model is created from the mating conditions between the components in the assembly. The joint information (i.e. degree of freedom) is automatically derived by the system from the mating conditions associated to each link.

Kramer [4] uses the degree of freedom analysis to satisfy the constraint. When the set of constraints has been specified, the system will identify the degree of freedom associated with each rigid body and devise a sequence of transformation operations to solve these constraints.

Mullins and Anderson [8] extend the degree of freedom analysis to solve part-positioning problems for mechanical assemblies in three dimensions. The positioning algorithm consists of a two-stage solution, an initial hierarchical positioning phase and the final simultaneous position solution. In the first stage, the parts are partially positioned one at a time and some degrees of freedom are removed from them. The remaining degrees of freedom will be used in the second stage in which the parts are assigned their final positions in the assembly.

Turner [9] has introduced the *Reduce* algorithm in which the degree of freedom of an object that has a multiple constraint may be reduced. If two rigid bodies are related by multiple constraints, then the *Reduce* algorithm can be applied to determine a single resultant constraint. The resultant constraint is the intersection of the two sets of rigid body motions described by the original constraints. However, the *Reduce* algorithm only works for the simple motion and is not suitable for deriving the motions of mechanisms.

Most of the degree of freedom analysis approach is based on the algebraic and geometric reasoning approach [3,4,8] where the degree of computation is expensive. A set of constraints supported by these systems is also limited [3]. In addition, there has been a limited use of graphical interaction techniques for carrying out assembly operations in the current system [10]. Therefore, it is not possible for the user to directly manipulate the assemblies and interactively analyze the kinematic behavior of an assembly. The result is that defining a complete set of constraints for an assembly model becomes tedious and error-prone [7].

We propose a new approach to integrate the constraints and direct 3D manipulation techniques. This technique enables the user to directly interact graphically with the assembly without having to explicitly specify the constraints and the 3D transformations.

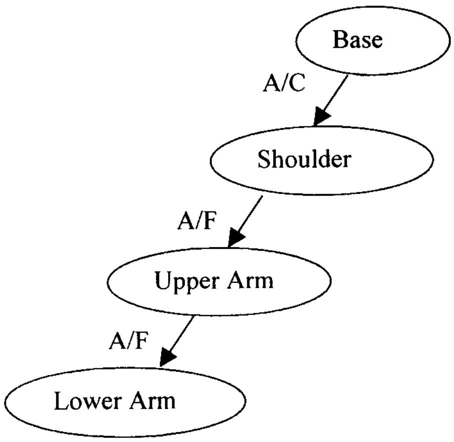
3. Interactive Constraint-Based Assembly Modeling

An Interactive Constraint-Based Assembly Modeling (ICBAM) enables the user to carry out the assembly work purely by direct 3D manipulation. It consists of three 3D-interaction techniques: *automatic constraint recognition, allowable motion,* and *graph-based algorithm.*

3.1 Automatic Constraint Recognition

The *automatic constraint recognition* process is used to identify constraints purely from 3D manipulation of the solid model that is being manipulated (*Tar-SM*). These constraints are represented in the relationship graph (*RG*).

The RG (Fig. 1) is a directed graph where each node corresponds to an assembly component. Each arc in the RG corresponds to a constraint between two components. The direction of the arc denotes the dependency between two components. The component that the arc points to, depends on the component from which the arc departs. This implies that the changes in the 3D location of the latter are always propagated to the former.



A: Against, F: Cylindrical fit, C: Concentric

Figure 1. The relationship graph of a puma robot

During the interactive assembly process, if a new constraint is recognized, a visual feedback is provided to the user by highlighting the relevant geometric elements. If the user continues to move the *Tar-SM*, the newly identified constraint is ignored and the constraint recognition process is continued. On the other hand, if the user does not move the *Tar-SM* within a prescribed time, the constraint is inserted into the RG. Each constraint has its own allowable motion which is used to automatically derive the *motion variable*, *number of degrees of freedom*, and *relative motion* [6] as shown in Table 1, where θ, ϕ, ψ refer to the rotational part whereas s, x, and y refer to the translation part.

Constraint	Symbol	Motion variable	Degree of freedom	Relative Motion
Against	A	Δx, Δy, $\Delta\theta$	3	Planar
Concentric, Cylindrical fit	C, F	$\Delta\theta$ and Δs	2	Cylindrical
Spherical fit	SP	$\Delta\theta$, $\Delta\phi$, $\Delta\psi$	3	Spherical
Screw fit	S	$\Delta\theta$ or Δs	1	C-TR
Gear contact	G	$\Delta\theta$	1	C-RR
Rack-pinion contact	RP	$\Delta\theta$ or Δs	1	C-TR

Table 1. Constraints and their relative motion

In this work, a set of complex constraint such as *screw fit (S)*, *gear contact (G)* and *rack-pinion contact (RP)* has been implemented. A set of simple constraints [3] such as *against (A)*, *concentric (C)*, *cylindrical fit (F)*, and *spherical fit (SP)* is also supported.

3.2 Allowable Motion

The term *allowable motion* refers to the remaining degrees of freedom for an under-constrained solid model or geometric element. An allowable motion of a solid model is used as a means of automatically constraining 3D manipulations of the solid model without invalidating its existing constraints. This results in an intuitive way of achieving the accurate 3D positioning of the solid model simply through 3D manipulation. Allowable motion technique is also used to provide an efficient means of satisfying a constraint by applying a number of transformation operations to the solid model. The following allowable motions are identified and implemented (Fig. 2, 3, 4, and 5).

3.2.1 Translation on Plane (*TR-PL*): The origin of the local coordinate system (LCS) of the component is translated on a plane.

3.2.2 Translation on Line (*TR-LN*): The origin of the LCS is translated along a line.

3.2.3 Translation in 3D Space (*TR-3D*): The origin of the LCS is translated in 3D space.

3.2.4 Non Translation (*NO-TR*): The origin of the LCS is fixed or can not be translated.

3.2.5 Rotation about Point (*RT-PT*): The LCS is rotated about a point.

3.2.6 Rotation about Line (*RT-LN*): The LCS is rotated about a line.

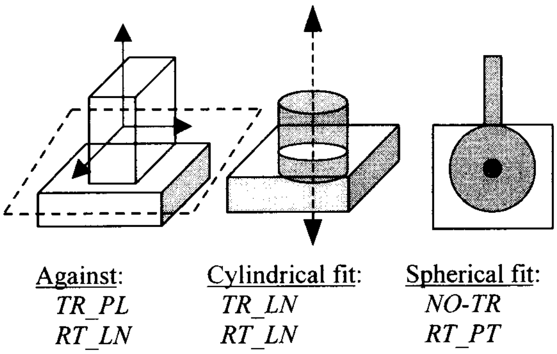


Figure 2. Simple allowable motions

3.2.7 Coupled Translation and Rotation *(C-TR):* This refers to a case where the allowable translation of the *Tar-SM* is coupling with its allowable rotation or with the allowable rotation of its mating component. For example, a screw fit restricts the motion between a screw and a nut to a combined rotation and translation along a common axis. The translation is a function of rotation θ and screw pitch as described below (Fig. 3):

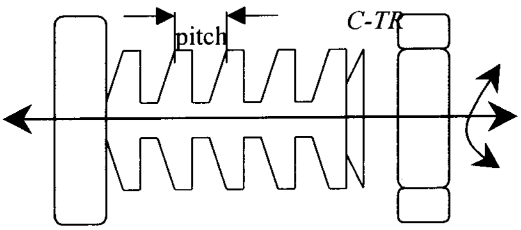
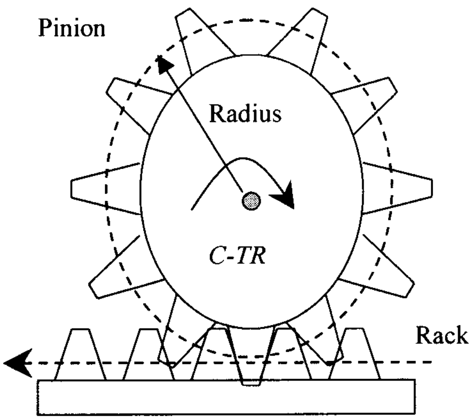


Figure 3. Allowable motion of Screw

$$translation(\Theta) = pitch * \frac{\Theta}{2\pi};$$

where θ is the actual rotation of the screw, and *pitch* is the screw pitch.

Similarly, the allowable motion for the rack-pinion can also be specified as a coupled translation and rotation. In this case, the allowable translation of the *rack (Tar-SM)* is converted into the allowable rotation of its mating component *(pinion)* as shown in Figure 4. The final motion is the translation along a rack's axis and the rotation about a pinion's axis. The rotation part is calculated from:



$$\Theta = \frac{translation}{pinion\_radius};$$

Figure 4. Allowable motion of Rack-Pinion

where θ is the actual rotation of the pinion, *translation* is the actual displacement of the rack, and *pinion_radius* is the radius of the pinion.

3.2.8 Coupled Rotation and Rotation *(C-RR):* This refers to a case where the allowable rotation of the *Tar-SM* is converted into the allowable rotation of its mating component. For example, a mating of gear teeth and a mating cam and follower. In the case of mating gear teeth, the rotary motion of the driver and the follower gears can be written in terms of the speed ratio (ω), where this speed ratio must remain constant in order to prevent any impact and vibration. At any instance the motion of the follower gear may be considered as a rotation about its axis where the direction is opposite to the driver gear's axis while maintaining the tangent constraint (Fig. 5). The actual rotation of the follower gear is calculated from:

$$\omega = \frac{\theta_F}{\theta_D} = \frac{T_D}{T_F};$$

where T is the number of gear teeth, and θ is the actual rotation of the gear. F and D denote the follower and the driver gear respectively.

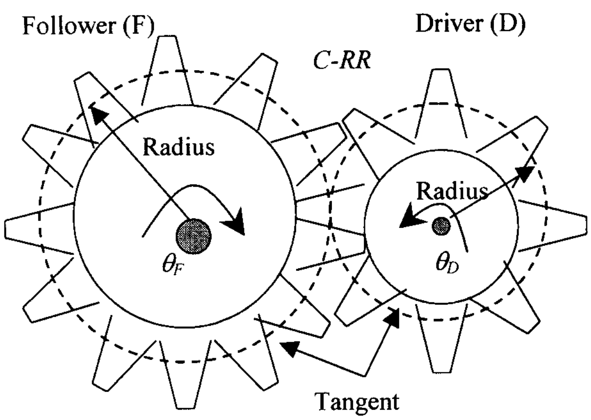


Figure 5. Allowable motion of Gear Contact

When several constraints are imposed on *Tar-SM*, the resulting allowable motion is determined by intersecting the allowable translations and rotations. Table 2 summarizes the results for the intersection of two allowable translations. For example, the intersection of two allowable translations is the translation on the intersection geometry of their motion geometric elements (plane or line). If the intersection geometry is a point, the allowable motion of the *Tar-SM* is non-translation *(NO-TR)* and the intersection point defines its 3D position. For a complete detail of the allowable motion intersection, the reader refers to [12].

TR-Type	TR-3D	TR-PL	TR-LN
TR-3D	*TR-3D*	*TR-PL*	*TR-LN*
TR-PL	*TR-PL*	*TR-LN*	*NO-TR*
TR-LN	*TR-LN*	*NO-TR*	*NO-TR*

Table 2. Intersection of allowable translations

3.3 Graph-Based Algorithms for Kinematics Simulation

This section presents a new graph-based algorithm that can efficiently perform forward and backward propagation in the directed RG from the user's 3D manipulations. The significant feature of this algorithm is its ability to ensure interactive response to 3D manipulations by exploiting the allowable motion techniques and the directed nature of the RG. This algorithm is also used for simulating the kinematic behavior of an assembly in response to user's 3D manipulation. For example, if the user rotates the shoulder of the puma robot (Fig. 6), the lower and the upper arm should be rotated too.

3.3.1 Forward Propagation

This is concerned with propagating the user's 3D manipulations of the *Tar-SM* to its children so as to maintain their associated constraints as valid. To achieve this, a particular propagation sequence is derived that ensures each object to be processed before its dependent children. This sequence also avoids repeatedly updating the same object. It is obtained in two steps:

1) Order the graph nodes through the breadth-first traversal of the RG, starting from the node of the *Tar-SM*;

2) Remove repeated graph nodes in a reverse order.

Consider a puma robot in Figure 6 and its RG in Figure 1, suppose the user rotates the shoulder, the propagation sequence is derived as follows: start with the node *Shoulder*, perform a breadth-first traversal on the RG. This results in the order: *Shoulder – UpperArm - LowerArm*. The changes in the 3D location of the *Shoulder* is then propagated to each dependent object (*UpperArm* and *LowerArm*) simply by transforming it in the same amount as the 3D transformation of the *Shoulder*.

3.3.2 Backward Propagation

Backward propagation is required if the forward propagation can not be initialized from the *Tar-SM*. The propagation sequence is derived by selecting one of its parents in the RG nodes whose allowable motion is available to absorb the 3D manipulation of the *Tar-SM* as follows:

1) Order the graph nodes through the breadth-first traversal of the RG by following the arcs of each node in a *reversed* direction, starting from the node of the *Tar-SM;*

2) Remove repeated graph nodes in a reversed order;

3) Check each node from the beginning of the order until a node (i.e. its parent) is found whose allowable motions are available to absorb the 3D manipulation.

From the previous example, suppose the user translates the *Shoulder*, this translation cannot be absorbed by the *Shoulder* (the allowable motion of the *Shoulder* is *NO-TR* and *RT-LN*). Therefore, the backward propagation is invoked in order to find the object that can absorb this translation. The propagation sequence may be obtained as follows: start with the node *Shoulder*, perform a breadth-first traversal of the RG in a reversed direction. This results in the final order: *Shoulder-Base*. The node *Base* is selected, since it has a free allowable motion. The translation will then be propagated from the node *Base* to all its dependent objects (the entire robot is translated).

4. Implementation and Results

This section evaluates the proposed 3D interactive graphical techniques discussed in section 3. In particular, the automatic constraint recognition, allowable motion, constraint satisfaction and constraint propagation for several constrained models are evaluated. This evaluation has been carried out using assemblies from several industrial case studies such as *a puma robot*, a *gearbox, a ball-socket, a rack-pinion*, and *a bolt-nut assembly*.

The system is implemented on the SGI workstations such as Indigo, Indy, and Indigo2 using C++ language. The IRIS Inventor supports the visualization of the virtual environment. The system enables the user to carry out interactive assembly and disassembly operations simply by 3D manipulations in a realistic manner as in the physical world. Currently, the system supports a rich set of 3D constraints such as *against, concentric, cylindrical fit, spherical fit, screw fit, rack-pinion contact* and *gear contact*.

The interactive assembly for these case studies is conducted based on their mating conditions that are automatically recognized purely from user's 3D manipulation. The allowable motion is used for satisfying these constraints and to simulate the kinematic behavior of an assembly through a graph-based algorithm. Figure 6-10 illustrates the results of the case studies.

Figure 6a shows four components (*Base, Shoulder, UpperArm,* and *LowerArm*) that are to be assembled by 3D manipulations. First, the *Shoulder* is placed on top of the *Base* using *against* and *concentric* constraints. Next, the *UpperArm* is connected with the *Shoulder* using *against* and *cylindrical fit* constraints. Finally, the *LowerArm* is inserted into the *UpperArm* using *against* and *cylindrical fit* constraints. The final assembly is shown in figure 6b in which the user can manipulate the *LowerArm* to reach an arbitrary position as discussed in section 3.3. This case study evaluates constraint recognition and allowable motion for *against, concentric,* and *cylindrical fit* constraints.

Figure 7a shows sub-assemblies of the gearbox (*input shaft, intermediate shaft, output hub,* and *case*) that are to be assembled by 3D manipulations (see [14] for detailed assembly of each of the subassemblies). Figure 7b shows the final assembly of the gearbox. This case study evaluates the *gear contact* constraint. The kinematic behavior of the gearbox can be simulated as follows: first, the user rotates the input shaft. Next, the pinion of the input shaft drives the first reduction gear, which in turn drives the second reduction gear. Finally, the second reduction gear drives the output hub. This kinematic motion is performed simultaneously based on the appropriate gear speed ratio as discussed in section 3.2.

Figure 8a shows the components of the rack-pinion assembly (*rack* and *pinion*), that are to be assembled. When the user translates the pinion until its gear teeth are in contact with the rack, the rack-pinion constraint is recognized. The allowable motion will satisfy this constraint as it makes both gear teeth properly engage as shown in figure 8b. Its kinematic behavior can then be simulated. This case study demonstrates the *rack-pinion contact* constraint.

Figure 9a shows four components of bolt-nut assembly (*plate1, plate2, bolt,* and *nut*) that are to be assembled. First, plate2 is placed on top of plate1 using against constraint. Then, the centerlines of the two holes in both plates are made coincident using concentric constraint. Next, the bolt is inserted into the hole using cylindrical fit constraint. Finally, the nut is inserted into the bolt using the screw fit constraint. The allowable of the screw fit can be demonstrated by turning the nut to lock and unlock the plates as shown in figure 9b. This example demonstrates the *screw fit* constraint.

Figure 10a shows two components of the ball-socket assembly (*ball* and *socket*) that are to be assembled. The ball in inserted into the socket using spherical fit constraint. The allowable motion satisfies this constraint by maintaining the center point of the ball and the socket to be coincident as shown in figure 10b. Therefore, the ball can only be rotated about its center point as defined by the spherical fit constraint (see Table 1 and Figure 2). This example evaluates the *spherical fit* constraint.

5. Conclusions and Future Work

3D interaction and constraint techniques make it possible to support interactive assembly and disassembly operations simply by direct manipulations. Importantly, these techniques allow the kinematic behaviors of under-constrained models to be interactively simulated by 3D interaction. Such techniques are essential for exploiting VE technology in engineering applications such as solid modeling, assembly modeling, and kinematic simulation.

Further research is underway at SIIT in collaboration with the Asian Institute of technology (AIT) to exploit the work described in this paper and the grid generation methods [11] in building a virtual 5-axis milling machine simulation [13] for the Maho600E 5-axis machine. The output from the simulation may be used to estimate an error, handle the tool interference, identify the tool shape, and adapt the tool path before actually testing with the real machine.

6. Acknowledgements

The author would like to thank the National Electronics and Computer Technology Center (NECTEC), National Science and Technology Development Agency (NSTDA) of Thailand for their financial support.

7. References

[1] Sodhi, R., and Turner, J.U., *Towards Modeling of Assemblies for Product Design*, CAD, V26, N2, pp. 85-97, February 1994.

[2] Fernando, T., Fa, M., Dew, P.M., and Munlin, M., *Constraint-based 3D Manipulation Techniques for Virtual Environments*, Virtual Reality Applications, pp. 71-89, 1995.

[3] Kim, S. H., and Lee, K., *An Assembly Modeling System for Dynamic and Kinematic Analysis*, CAD, V21, N1, pp. 2-12, Jan 1989.

[4] Kramer, G. A., *Solving Geometric Constraint System -- A Case Study in Kinematics*, MIT Thesis, 1992

[5] Morris, G. M., and Haynes, L. S., *Robotic Assembly by Constraints*, IEEE International Conference on Robotics and Automation, pp. 1507-1515, Mar 1987.

[6] Joseph Edward Shigley and John Joseph Uicker, Jr., *Theory of Machines and Mechanisms,* McGraw-Hill Book Company, 1980.

[7] David Neville Rocheleau and Kunwoo Lee, *System for Interactive Assembly Modeling*, CAD, V19, N2, pp. 65-72, 1987.

[8] Mullins, S. H., and Anderson, D. C., *A Positioning Algorithm for Mechanical Assemblies with Closed Kinematic Chains in Three Dimensions,* Proc. Symposium Solid Modeling and Applications, ACM Press, New York, pp. 271-282, 1993.

[9] Turner, J. U., and Subramaniam, S., and Gupta, S., Constraint Representation and Reduction in Assembly Modeling and Analysis, IEEE Transaction on Robotics and Automation, V8, N6, pp. 741-750, Dec 1992.

[10] Requicha, A. A. G., and Rossignac, J. R., Solid Modeling and Beyond, IEEE Computer Graphics and Applications, V12, N5, pp. 31-45, Sep 1992.

[11] Makhanov, S. S., *An Application of the Grid Generation Techniques to Optimize a Tool-Path of Industrial Milling Robots*, J. of Computational Mathematics and Mathematical Physics, V39, N9, pp.1589-1600, 1999.

[12] Fa, M., and Fernando, T., and Dew, P. M., *Interactive Constraint-based Solid Modeling using Allowable Motion*, Proc. of ACM/SIGGRAPH Symposium on Solid Modeling and Applications, pp. 243-252, May 1993.

[13] Munlin, M., and Makhanov, S. S., *Constraint-Based Simulation of a 5-Axis Milling Machine*, Proc. of International Conference on Production Research, August 2000 (CD-ROM).

[14] Munlin, M., *Interactive Assembly Modelling within a Virtual Environment*, PhD thesis, School of Computer Studies, the University of Leeds, 1995.

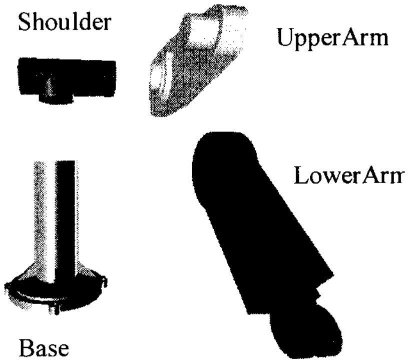


Figure 6a. Puma robot (Initial position)

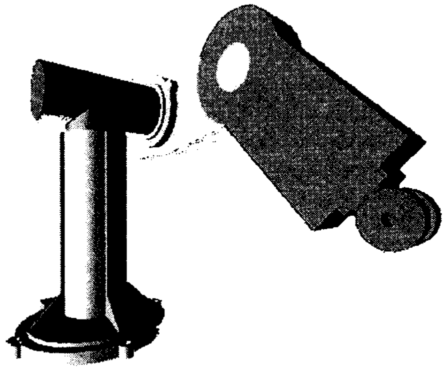


Figure 6b. Puma robot (Final position)

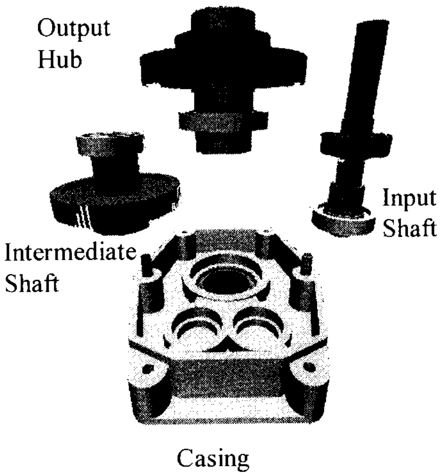


Figure 7a. Gearbox Assembly (Initial position)

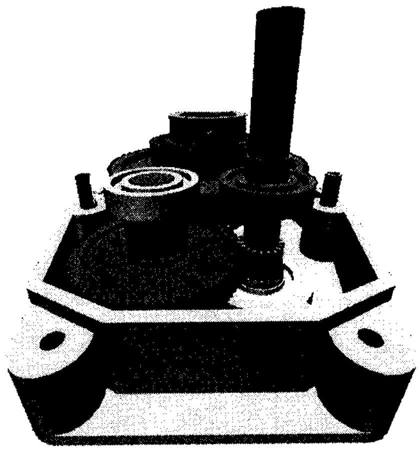


Figure 7b. Gearbox (Final position)

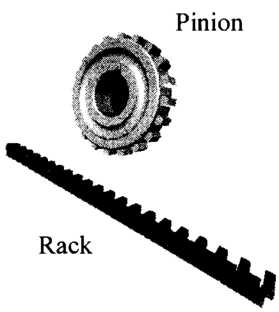


Figure 8a. Rack-Pinion Assembly (Initial position)

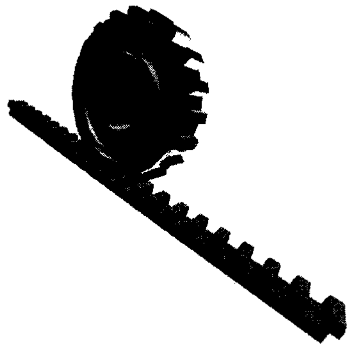


Figure 8b. Rack-Pinion (Final position)

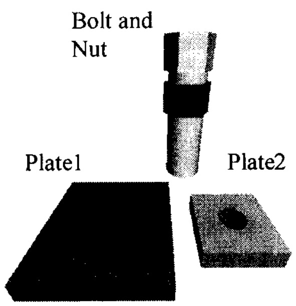


Figure 9a. Bolt-Nut Assembly (Initial position)

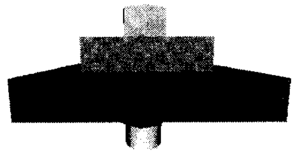


Figure 9b. Bolt-nut (Final position)

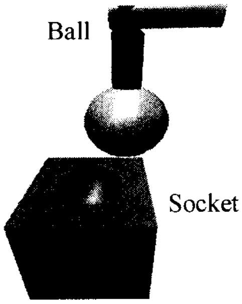


Figure 10a. Ball-Socket Assembly (Initial position)

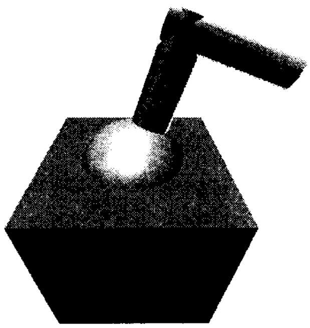


Figure 10b. Ball-Socket (Final position)