

Haptics Aided Kinematic Assembly Modeling and Efficient Determination of Joint Ranges of Motion

Ashvinikumar Patil¹, Dibakar Sen^{2*}

¹ Flight Mechanics and Control Engineering, Aeronautical Development Establishment, Bangalore, India

² Mechanical Engineering, Indian Institute of Science, Bangalore, India

* Corresponding author (email: dibakar@mecheng.iisc.ernet.in)

Abstract

This paper presents the development of a tool for the assembly and kinematic modeling of the tessellated objects. The tool addresses the issue of reducing the effort required to generate the kinematic models. The tool is capable of automatically ascertaining the type of kinematic pairing between two part models represented in STL format by detecting the compatible combination of features and determining the joint range of motion (ROM). The haptic interface is used to realistically simulate the assembly process and to determine the range of motion in 3D. The feature extraction algorithm uses a mix of curvature and surface normal based segmentation of the tessellated surfaces. The computation necessary for the geometry extraction and joint recognition has been decoupled from the complexity of the models by “Region of Interest” identification. A novel method for the determination of the ROM is proposed, wherein the recognized joint type information is used to restrict the computation necessary to the physically allowed degrees of freedom (DOF), unlike in the methods like configuration space, where, the computation needs to be done in 6-DOF space. Also, an efficient haptic rendering scheme for the simulation of the ROM has been proposed, which eliminates the need for the computationally intensive collision detection between the two parts.

Keywords: Kinematic assembly modeling, Tessellated objects, Feature recognition, Range of motion, Haptic assembly.

1 Introduction

The type of constraints that relate the two parts to each other differentiates the geometric and kinematic assemblies. The geometric assembly use the part position constraints like align, mate, insert, etc. while the kinematic assembly specifies the relative motion constraints using the type of joint like cylindrical, spherical, revolute, etc. The usual way for any kinematic modeling of a mechanism essentially involves, importing the CAD geometry into the analysis tool and manually specifying the joints. Gen-

erally, the parts will be assembled in GCAD tool; even if this assembly is imported in to the kinematic analysis tool, the manual specification of the joint is unavoidable as the geometric assembly constraints will not be able to define the correct motion constraints.

The task of creating a kinematic assembly model is also complicated by the lack of universal data interchange format. As various geometric CAD (GCAD) tools used for creating the part models, have their own native data formats and data interchange between the tools is not simple or straightforward.

As the number of components in the mechanism increase, the task of specifying a kinematic assembly becomes increasingly difficult [1].

Commercially various strategies have been used for reducing the effort necessary for the conversion of the GCAD assembly to kinematic assembly. Some tools offer direct geometric and mass property data connectivity between the GCAD and the analysis tool. Once the kinematic model has been created inside the analysis tool by using the parts/assembly generated in a GCAD tool, any changes to the geometry made in the GCAD will automatically be reflected in the analysis tool, thus saving the effort of recreating the kinematic model. Some other tools provide a kinematic analysis kernel inside the GCAD. However, here only data conversion effort is saved and the additional task of assigning the joint types manually still exists. Also, these implementations suffer from the limited capabilities.

This paper reports initial results of implementation of a geometric and kinematic assembly modeling environment; using tessellated part models as the neutral format, which also enables a simple but accurate determination of ROM and the simulation of the joint ROM devoid of any collision detection.

1.1 Background

One of the attempts to implement automatic kinematic model creation is reported in [2]. Here, the part models in native SolidWorks[®] format are used as input and the geometrical data is handled using ACIS[®] geometric kernel. The user interactively selects the location of the intended assembly on both the parts. The kinematic joint, if any, is

output along with the geometry data as a batch file that can be read from ADAMS[®] to create the kinematic assembly. It reported various limitations of the implementation like layering, where, the parts, even though are in kinematic assembly, would not form a valid physical assembly, viz. for a planar 4-bar mechanism, a valid kinematic model is obtained if the joint axis of the two adjoining parts are collinear, but are physically separated along the axis and redundant configuration creation. These limitations appear to be a direct consequence of the way the user interacts with part models and specifies the assembly. The procedure fails to capture the designer's intent about the configuration of the mechanism. When the designer positions the two parts together, he generally has a specific idea about their behavior and of the joints in the mechanism. Capture of design intent is desirable for both meaningful kinematic analysis and validation of the geometric assembly configuration.

1.2 Motivation

The present work is based on the theory presented in [2]. It has been emphasized there that the relative configuration and motion capabilities are determined by the nature of contacts as admitted by the geometric forms of the parts in proximity. Hence geometric constraints as used in assembly modeling and kinematic constraints as used in kinematic modeling are both simultaneously derivable from the understanding of the physically admissible contact configurations and interpretation of the designers' intent.

When parts in contact have relative DOF, their relative motion often can be described as a lower kinematic pair. In that case, the geometric entities (surfaces, edges or points) in contact are necessarily associated with planar, cylindrical or spherical surfaces. These characteristics have been extensively used in [2] for determination of the contact configurations of the parts in proximity and thus eliminating the need for explicit specification of the tedious and error-prone geometric/kinematic constraints. In the present work, dependence of the procedures in [2] on the analytical representation of the faces is relaxed by systematically deriving the analytical information from the tessellated models. Tessellated representation has been growing in popularity for virtual reality, process planning and manufacturing. Hence, it has been used in the present work.

Haptics provides the user with full 3D experience when compared to the devices like 3D mouse as the latter provides access to only the visible portion of the environment. Hence, it's applications to realistic assembly simulation have been growing. One such early attempt is reported in [3]. Issues related to assembly using haptics and its effectiveness is reported in [4] and [5]. A brief discussion on the issue of haptic effectiveness in the design of mechanisms can be found in [6] and the references therein. One of the bottlenecks in the present haptics enabled simulations is the requirement of fast and accurate collision detection. The present work demonstrates that the appreciation of the transitional kinematic behaviour of the parts being assembled can even eliminate the need for collision detection for meaningful and realistic assembly simulation.

2 Environment

The haptic environment is implemented using the Sensable[®] make 6 DOF Phantom Omni device. The part models can be manipulated by the user with this device to intuitively bring them close to the intended assembly positions.

To reduce the computation, only the necessary portions of the two parts are selected using a 'Region of Interest' (ROI) concept. The ROI is depicted on the screen as a semi-transparent sphere. Only the triangles that are enclosed by or intersect the sphere are used for the recognition process. This sphere is automatically generated at the intended assembly position based on the proximity between the parts. It is assumed that, when the parts are positioned for assembly, the surfaces of both the parts that are going to form the joint will be closest to each other. The proximity detection has been implemented using an axis aligned boundary box (AABB) algorithm [7] with an efficient binary-tree data structure.

An axis-aligned boundary box is a rectangular box aligned with the co-ordinate axis, which encloses the entire bounded object. The boundary box hierarchy is the arrangement of boundary boxes in a binary tree where each level consists about the bounding box of that node and address of the two children. The children are obtained by cutting the parent box in to approximate halves in the longest direction (to reduce the depth of the tree). Starting from box of the whole object (Node) the cutting continues until each leaf consists of single triangle. This algorithm pre-selects the probable nearest triangles on the second part for every point on the first. The actual distance between parts is obtained by the nearest distance between vertex on one part and the triangle of the other part.

Apart from reducing the computation, by reducing the number of measurements between the vertices of one part to the triangles of the other, the ROI also achieves another significant result. By restricting the computation to only the selected mating portions of the two parts, *it makes the joint identification process independent of the part complexity* in terms of geometry and number of the triangles forming the parts. Thus, only few surface types need to be identified from a sub set of the total triangles forming the two parts.

3 Kinematic Joint Identification

The process of kinematic modeling using the tessellated objects involves first, the feature extraction (surfaces like plane, cylinder, etc and lines, arc, points representing edges, vertices) from the set of triangles that form the object, characterize the features by determining the size and orientation in space and then using these features to identify feasible combinations that can form valid joints. In this work, the geometrical features are extracted from the tessellated model using a hybrid segmentation method using both curvature and surface normals as the surface descriptors. The various schemes of segmentation like direct segmentation, edge detection and region growing method and their pros-cons are discussed in [8] and [9]. The detailed description and discussion about the vertex

curvature based segmentation can be seen in [10] and [11]. The curvature is the most predominant surface descriptors for segmentation for the scanned models or where the density of the vertices on the surface is dense. The various methods for curvature determination can be found in [12] – [16].

The segmentation and feature extraction of the CAD generated tessellated objects offer a different set of challenges in that the tessellations produced by commercial CAD packages tend to minimize the number of vertices and the surfaces like planes, cylinders often do not have any vertices available except those on their boundary curves. Hence, either the surface need to be populated with vertices in the interior of these surfaces using a subdivision scheme or the surface normal data can be effectively made use of. The segmentation of the CAD generated models has been discussed in [17] and [18]. However, they can not be used here directly for we are interested only in surfaces of specific kind and we do not need to segment the whole object.

3.1 Feature Recognition

In this work, the curvature estimation method in [16] and the ‘watershed’ algorithm for region growing segmentation in [19] have been used for surfaces with well-distributed sample points. For segmentation of the optimally tessellated surfaces, the algorithm uses the variation of the surface normals to classify the surfaces. A planar surface has all its constituting triangles with their normals parallel to each other. The normals on a cylindrical surface are ideally coplanar; hence the normals on the tessellated cylinder would also lie on a plane perpendicular to the axis of the cylinder with a small tolerance zone. Similarly, for a conical surface, the set of surface normals form a cone with a cone angle equal to the complimentary angle of the original cone. Fig. (1) shows the segmentation of some optimally tessellated objects.

In this work, the planar surfaces are obtained and the triangles in each such surface are encoded with a unique number. The remaining triangles are tested for the cylinder and conical surfaces.

When the object has both types of surfaces, i.e. optimal and dense; curvature and surface normal based algorithms need to be used together to segment the object properly. The detection of the two types is carried out through the process of edge detection. The edges of the surface are determined using a threshold on the angles between the two adjacent triangles. For the examples tried, the threshold angle of 30° recognized the object edges effectively! The watershed algorithm scans the model for suitable triangles for segmentation and the remaining triangles are processed by the normal based algorithm. Fig. (2) shows the segmentation of the orb of a sphere to demonstrate this.

Fig. (3) shows the example of the tessellation of a cylinder by Solidworks. Fig. (3a) shows the cylinder with the end faces perpendicular to axis and Fig. (3b) shows the same with one face at an angle to the axis. It can be noted that the second model has a remarkable visibly irregular

tessellation, which makes some of the triangles ‘improper’, in the sense that orientation of their surface normals do not agree with the neighboring triangles in forming a cylindrical surface and lead to creation of multiple cylindrical surfaces separated by planar / conical surfaces. This issue exists in any cylinder or a cone, where the end surfaces are not perpendicular to the axis. Fig. (3c) shows a typical plot of normals for a distorted cylindrical surface.

To a certain extent, the cylinder case can be simply handled by increasing the tolerance band to catch the odd improper triangle. Otherwise, a region growing method implementation is required; where, the identified surface patches are treated as the seed surfaces and intelligently merge the triangles left out during the first phase of segmentation with the appropriate seed surfaces.

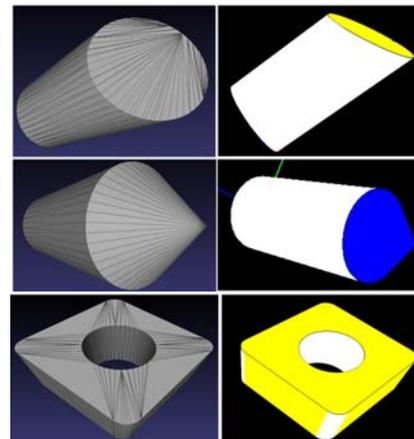


Fig. 1: Segmentation of the shapes. The left panes show the STL models and the right ones the models after segmentation and surface identification.

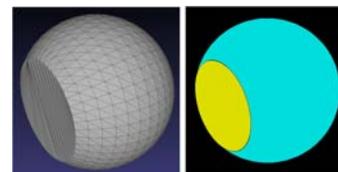
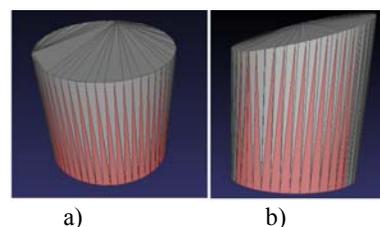


Fig. 2: STL model and the segmented model of the orb of a sphere showing both dense (spherical) and optimal (plane) surfaces.



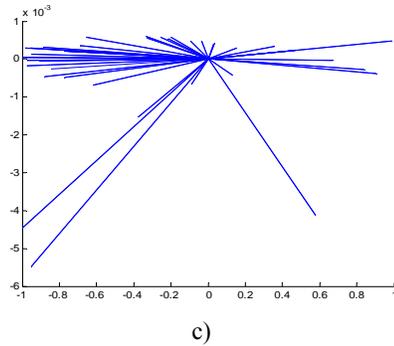


Fig. 3: Tesselation of CAD surfaces.

a) Cylinder with end faces perpendicular to the axis and
b) at an angle to the axis. c) 3-D plot of the surface normals
of the cylinder shown in b).

3.2 Shape Characterization:

The segmented patches of the objects are evaluated to identify the characteristics of the targeted surfaces and are classified. The surfaces are restricted to plane, cylinder, cone and sphere, as they are associated with the five out of six canonical lower kinematic pairs. Fig. (4a) and (4b) shows the necessary surface types identified for classical peg-in-hole assembly. The grayed areas indicate the surfaces that have been ignored in the recognition process. Fig. (4a) shows the parts positioned for assembly and the ROI sphere highlighting the triangles that need to be processed. Similarly, surfaces identified in the assembly of sphere and cylindrical hole can be seen in (4c) and (4d).

These identified surfaces need to be represented in 3D for the detection of the kinematic pairing. The plane needs the surface normal and a point on it; sphere needs the radius and the position of the center. The radius of the sphere is obtained from its curvature value and its concavity or convexity is determined by using any three points on the surface, radius and the surface normal of the triangle formed by the selected triangles. Fig. (5) shows the relation among these variables.

A cylinder needs two points for defining the axis and another number for defining its radius. The line segment defining the axis can be used to represent the effective length of the cylinder as well. For representing a cone, in addition to the axis and base radius, it also needs the cone angle. For the determination of the axis of the cylinder or cone, we can use the Gaussian map of the facet normals. The candidate facets are the ones having all their vertices on the face boundaries. For a cylindrical face, the Gaussian maps of the facet normals lie on a great circle of the unit sphere (Fig. 6a) and for a conical surface the facet normals lie on a circle different from the great circle (Fig. 6b). Let its radius be r . Then the cone angle is given by $\cos^{-1}(r)$. The direction normal to the plane of the circle gives the direction of the axis of the cylinder/cone. The axis length is obtained from the projection of the triangles on the axis. Tesselation of the cylinder or cone produces triangles some of whose edges belong to the generators; for cylinder these generators are parallel to the axis and for cone each of these generators form a plane with the axis and they

intersect the axis. After identifying the generators, their intersection with the plane perpendicular to the axis are determined. The radius of the circle through these points gives the radius and the centre of the circle defines the location of the axis of the cylinder.

3.3 Kinematic Joints

The kinematic joint information is extracted from the set of selected surfaces by identifying compatible combinations of the surfaces. For example, in the peg in a hole example the two cylindrical surfaces form a joint if their diameters are compatible and the joint can be either cylindrical or revolute based on their length.

The AutoKAM tool reported in [2] is used to evaluate the joints from the set of surfaces obtained. Apart from the joints that can be obtained by conformal surfaces as explained in the previous paragraph, the tool is also capable of finding higher pairs like those formed by vertex/surface or edges/surface or edge/edge combinations. Here only extraction of the lower pairs is reported.

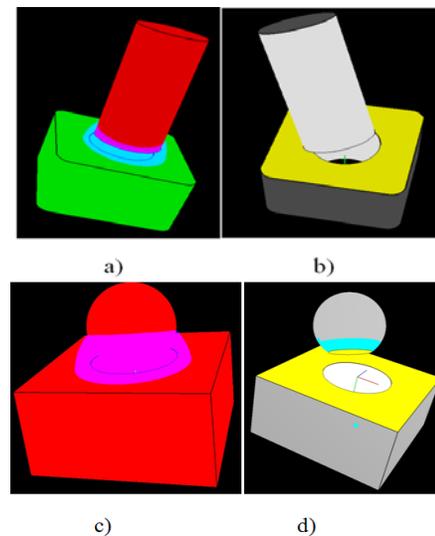


Fig. 4: Segmentation and characterization of the selected surfaces. a) Shows the cylindrical peg and the block with the similar hole in intended assembly position, the region of interest sphere is in blue. b) Shows the identified surfaces. Surfaces in grey are not considered for the segmentation. c) and d) show the sphere and cylindrical hole assembly.

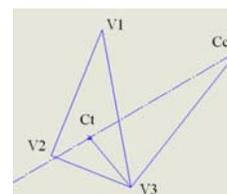


Fig. 5: Triangle with all vertices on the spherical surface used for determining center of sphere.

V1, V2, V3 are the vertices on the spherical surface, Ct is the centroid of the triangle and Cc is the center of the sphere and Ct – Cc is the line perpendicular to the triangle through Ct.

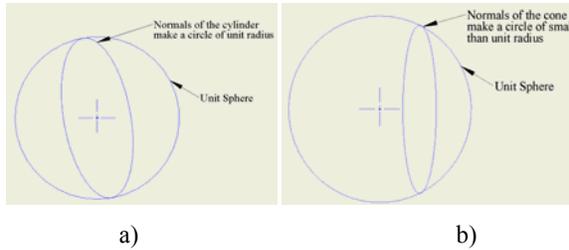


Fig. 6: Circles formed by the normals of the a) cylindrical surface and b) conical surface.

4 Range of Motion

The Range of Motion (ROM) of a joint is defined as the range of physically admissible motion for the joint. The range of motion determination for a pair of tessellated objects is computationally intensive. Any object in 3D space has 6 degrees of freedom. The typical configuration space based approach ([20], [21]) plots the orientation of the two parts with reference to each other over the 6-dimension space to identify likely joints / kinematic pairs. This is achieved by analyzing the relative part positions and the type of contact curves generated.

We present a simple but accurate method to determine ROM. Instead of searching the free space for contact, and then determining the joint type based on the obtained DOF, in our work, the joint type is recognized first based on the compatible surface configurations and then the algorithm searches only in the space defined by the relative mobility of the kinematic joint in the context. This reduces the dimension of search space by $(6-f)$ where, $f = \text{DOF}$ of the joint. It also cuts down the computational effort by eliminating the joint level constraint computation when the mechanism level ROM is being evaluated. In this work, the prismatic, planar, revolute and cylindrical joint types are considered.

The prismatic joint ROM is evaluated by finding the minimum and maximum straight-line distance between the parts, for which the joint is valid. Fig. (7) shows one extreme case for the joint to be valid. The other extreme case is when the shaft is fully inside the hole. In this example, the maximum value of ROM is L1 and minimum is zero. The algorithm based on [22] is implemented for the vertex to triangle distance computation. Fig. (8) shows the assembly of two parts forming a prismatic joint, with the ROM indicated by an arrow. The time for determination of ROM is ~40mS.

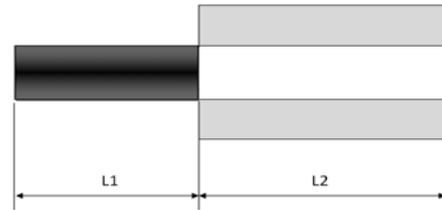


Fig. 7: Prismatic joint range.

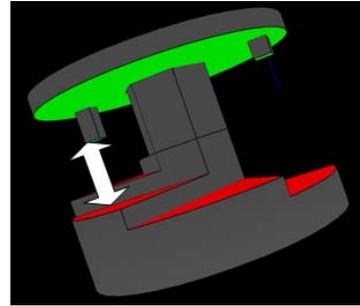


Fig. 8: Prismatic joint; the parts after assembly. Note that the white arrow indicates the ROM and the gray surfaces are not considered for the computation of ROM.

In the case of revolute joints, the ROM is obtained using the R- θ coordinate system. The angular separation of the two parts at every 'R' is evaluated and the minimum of θ in both CW and CCW direction defines the ROM. As the parts in a revolute joint maintain contact throughout 360° the joint is always valid. Fig. (9) shows the formation of a revolute joint, with the ROM indicated by an arrow. The time taken for ROM evaluation is ~60mS.

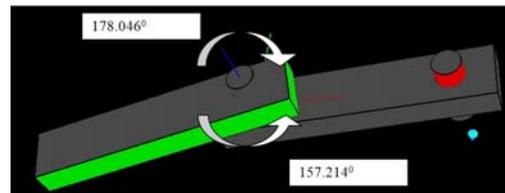


Fig. 9: Revolute joint; the parts after assembly. Note that the white arrows and highlighted numbers indicate the ROM and the gray surfaces are not considered for the computation of ROM.

The range of motion of a cylindrical joint is a 2D space. It is a combination of revolute and prismatic joints. First, with the algorithm for revolute joints, ROM in the direction of rotation is obtained. Then, using the algorithm for prismatic joint the ROM in axial direction is obtained. This sequence of rotation ROM and translation ROM evaluation is repeated at each grid point. Fig. (10) shows the assembly of two parts forming a cylindrical joint, with the ROM indicated by an arrow.

For the planar joint, the range of motion is a 3D space (2 for translations and 1 for rotation). The ROM evaluation uses the algorithms of both the revolute and prismatic joints. Actually, the motion limits need to be evaluated for

every point in the planar space and in every direction. For simplification, only translatory motion in only two perpendicular directions (aligned to the axis of the coordinate frame) and the rotation about discretized rectangular grid points (again, aligned to the axis) on the planar space is considered. Fig. (11) shows the planar joint space and the evaluation of the joint ROM at a point.

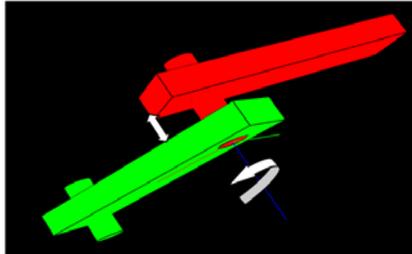


Fig. 10: Assembly and ROM of a cylindrical joint.

5 Haptics

In this work, the haptics has been used to enable the user to interactively assemble two objects and visually validate the calculated range of motion between them. The motion constraint has been added as a high stiffness spring force. For example, in a revolute joint only rotary motion along the axis of rotation is allowed and is restricted to the range of motion as calculated. The other objective of using haptic interaction in this work is to show that physically meaningful haptic rendering can be done without the conventional collision detection.

The issue of insertion of peg in the hole is discussed in [5]. When the geometrical dimensions between the two parts are very close, the haptic rendering using the collision detection method would result in instability in haptic rendering; this may become uncontrollable and make the assembly operation impossible. In [5], it is proposed to use additional data corresponding to the geometric data for better contact determination. In this work, as the joint is identified before the assembly operation, the contact determination is eliminated between the surfaces that form the joint and the assembly rendering is simplified as the assembly constraints are enforced using the geometric entities like line, arc and surface.

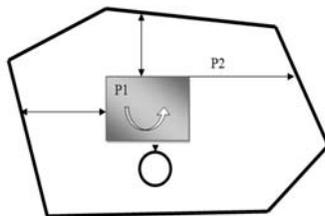


Fig. 11: Planar Joint ROM evaluation.

The prismatic joint ROM simulation is enforced using a line constraint. For the haptic rendering of the joint, the force applied at any arbitrary point on the link forming a prismatic joint simulation can be split into two parts. The first will be along the direction of the joint motion and this

component will not be resisted by the joint. The other component will be in a plane formed by the point of application of force and a plane perpendicular to the axis of the joint. This component of the force constitutes the resistive force. This resolution of the applied forces is shown in the Fig. (12a), where the first component will be along the joint motion axis and the second component will be in a plane comprising of two constraining forces as shown. Similarly, the revolute joint is simulated using an arc (or circle). The force-rendering scheme is as shown in Fig. (12b).

The linear motion or the angle of rotation of the parts is tracked with respect to the body fixed coordinate systems with the Z-axis along the axis of the joint, and X-axis towards the respective parts' CG and the Y-axis being perpendicular to both. If the CG exists on the joint axis, the X-axis is arbitrarily fixed.

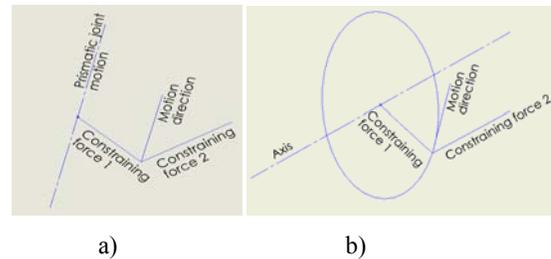


Fig. 12: Force rendering for a) prismatic joint and b) revolute joint simulation.

The planar, cylindrical and the spherical joints need the surface constraints. The surfaces with the continuous boundary as defined by the ROM need to be implemented. The planar joint will have the constraint component of the force perpendicular to the plane of motion and the other two components will lie in the plane of motion. Similarly, the force components for the cylindrical and spherical joints are shown in Fig. (13).

Fig. (14) and (15) show the snapshots of the ROM simulation for the prismatic and revolute joints.

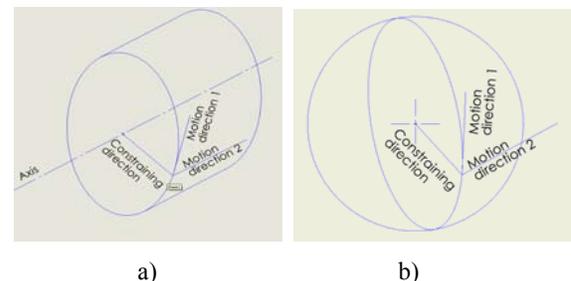


Fig. 13: Force rendering for a) cylindrical joint and b) spherical joint simulation.

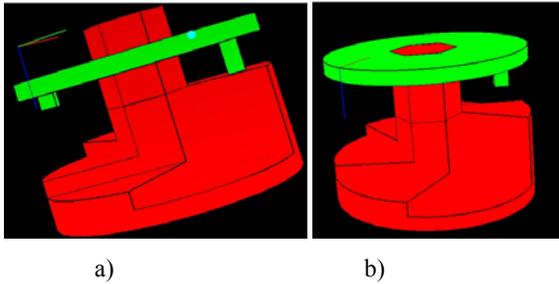


Fig. 14: Range of motion simulation snapshots for prismatic joint; a) parts in contact (minimum ROM value) b) parts at maximum separation (maximum ROM value). The cyan colored cone is the point of grasping.

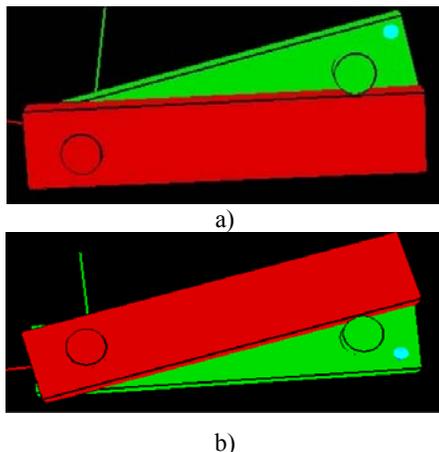


Fig. 15: Range of motion simulation snapshots for revolute joint; a) parts in contact in CW b) parts at contact in CCW direction.

5 Conclusions

The automatic extraction of the kinematic joint using the tessellated objects as neutral format demonstrates the reduction of the user effort by providing an intuitive modeling environment. It also unifies the geometric and kinematic assembly modeling for tessellated models. The environment facilitates capturing the design intent and assembly knowledge.

The adopted approach of using ROI effectively decouples the part complexity and the computation necessary for joint recognition by restricting the segmentation and shape recognition to a small portion of the tessellated part data and avoiding the complete part geometry extraction in case of the scanned data.

It is also shown that the approach of determining the ROM for a kinematic joint by using the joint type is simple and accurate compared to the other methods like configuration space, as the dimension of the search space is same as the DOF of the joint. The ROM is also rendered haptically with simple geometric constraints, which eliminates the need for the computationally expensive collision detection. This feature is expected to be useful for realistic assembly simulation applications.

References

- [1] V. N. Rajan, K. Lyons, R. Sreerangam, "Assembly representations for capturing mating constraints component kinematics", *IEEE international conference on assembly and task planning*, 1997.
- [2] A. J. Dawari, "Automated kinematic assembly modeling", MSc Thesis, Indian Institute of Science, Bangalore, 2007.
- [3] S. Jayaram et al, "A Virtual Assembly Design Environment", *IEEE Computer Graphics and Applications* Vol. 19, No. 6, 1999, pp. 44-50.
- [4] T. Lim et al, "Assessment of a Haptic Virtual Assembly System that uses Physics-based Interactions", *Proc. of the 2007 IEEE International Symposium on Assembly and Manufacturing Ann Arbor, Michigan, USA, July 22-25, 2007*, pp.147 – 153.
- [5] R. Iacob, P. Mitrouchev, J.-C. Léon, "Contact identification for assembly disassembly simulation with a haptic device", *The Visual Computer*, Vol. 24, No. 11, 2008, pp. 973-979.
- [6] S. Payandeh, J. Dill, J. Zhang, Using "Haptic feedback as an aid in the design of passive mechanisms", *Computer-Aided Design* Vol. 39, 2007, pp. 528–538.
- [7] T. Larsson, T. Akenine-Moller, "A dynamic bounding volume hierarchy for generalized collision detection", *Computers & Graphics* Vol. 30, 2006, pp. 451–460.
- [8] M. Vieira, K. Shimada, "Surface mesh segmentation and smooth surface extraction through region growing", *Computer aided Geometric Design*, Vol. 22, No. 8, 2005, pp. 771-792.
- [9] P. Benk"o, R R. Martin, T. Varady, "Algorithms for Reverse Engineering Boundary Representation Models", *Computer Aided Design* Vol. 33, No. 11, 2001, pp. 839-851.
- [10] P. J. Besl and R. C. Jain, "Invariant Surface Characteristics for 3D Object Recognition in Range Images", *Computer Vision, graphics, and image processing*, 1986.
- [11] S. Pulla, Curvature Based Segmentation of 3D Meshes, MS thesis, Arizona State University, 2001.
- [12] R. Sacchi, J. F. Poliakoff, P. D. Thomas, K. H. Häfele, "Curvature Estimation for Segmentation of Triangulated Surfaces", *Proc. of Second International Conference on 3-D Digital Imaging and Modeling*, 536-543, Ottawa, Ont., Canada, 1999.
- [13] G. Taubin, "Estimating the tensor of curvature of a surface from a polyhedral approximation". *Proceedings of the Fifth International Conference on Computer Vision*, 902-907, 1995.

[14] D. S. Meek, D. J. Walton, "On surface normal and Gaussian curvature approximations given data sampled from a smooth surface", *Computer Aided Geometric Design*, Vol. 17, No. 6, 2000, pp. 521-543.

[15] T. Langer, A. Belyaev, H-P Seidel, "Exact and interpolatory quadratures for curvature tensor estimation", *Computer Aided Geometric Design*, Vol. 24, No. 8-9, 2007, pp. 443-463.

[16] M. Meyer, M. Desburn, P. Schroder and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds", In *Visualization and Mathematics III(Proceedings of VisMath 2002)*, Springer Verlag, Berlin (Germany), 35-54.

[17] V. B. Sunil, S. S. Pande, "Automatic recognition of features from freeform surface CAD models", *Computer-Aided Design*, Vol. 40, No. 4, 2008, pp. 502-517.

[18] G. Lavoue', F. Dupont, A. Baskurt, "A new CAD mesh segmentation method based on curvature tensor analysis", *Computer-Aided Design*, Vol. 37, No. 10, 2005, pp. 975-987.

[19] A. P. Mangan and R. T. Whitaker, "Partitioning 3D Surface Meshes Using Watershed Segmentation", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 5, No. 4, 1999, pp. 308-321.

[20] L. Joskowicz, E. Sacks, "Configuration space visualization for mechanical design", *IEEE Visualisation'98*, pp. 483- 486, 1998.

[21] K-J Kim, E. Sacks, L. Joskowicz, "Kinematic analysis of spatial fixed axis higher pairs using configuration spaces", *Computer-Aided Design*, Vol. 35, No. 3, 2003, pp. 279-291.

[22] T. Moller, B. Trumbore, "Fast Minimum storage Ray / Triangle Intersection", *Journal of Graphics Tools*, Vol. 2, No. 1, 1997, pp. 21-28.