

Title:**Kinetic Model Generation for Virtual Commissioning**Authors:

Ji Myoung Park, jmpark89@ajou.ac.kr, Ajou University

Min S. Ko, killay@ajou.ac.kr, Ajou University

Sang C. Park, scpark@ajou.ac.kr, Ajou University

Keywords:

Virtual commissioning, Fixture modeling, Kinetic model, Geometric model, Gauss map

DOI: 10.14733/cadconfP.2019.278-282

Introduction:

The modern manufacturing environment can be characterized by significant cost constraints, global competition, shifting customer values, shortening of product life-cycles and rapid time-to-market. To be successful in the fast-changing manufacturing scenario, manufacturers must strive to improve not only the quality of their products but also the efficiency of their production systems. This demand has resulted in the concept of a virtual factory which is a model executing virtual manufacturing processes within a computer simulation [1, 5, 8]. For the implementation of a virtual factory, it is necessary to construct digital models for all the physical and logical elements (entities and activities) of a real manufacturing system [2, 4, 6, 7]. Once a virtual factory is constructed, it is possible to evaluate the physical validity and efficiency of co-working machines, as well as the production capability of the system. The key idea of virtual commissioning is to include real control devices (PLCs, PCs) of a production system in the simulation of a virtual factory. One of the major benefits of virtual commissioning is the early detection and correction of errors generated during the design stage of a production system. As a result, we can save a lot of time and efforts for debugging and correction expended during real commissioning. An experienced study in virtual commissioning [3] shows the positive effect of virtual commissioning on the error rate during real commissioning. The results showed a reduction of real commissioning time by 75%, resulting from enhanced quality of the manufacturing system at the start of real commissioning.

Although virtual commissioning may provide various benefits, it is not easy to enjoy the benefits of the technology in reality. One of the major obstacles of the virtual commissioning is the excessive time and efforts expended during the construction of virtual device models which require high fidelity. A virtual factory consists of various virtual devices such as robots, conveyors, fixtures, machining and assembly tools. Many of these devices need to contain both of a geometric model and a kinetic model.

To enjoy the benefits of virtual commissioning technology, it is essential to develop a more efficient methodology to construct virtual device models. As depicted earlier, a virtual device model consists of two sub-models, a geometric model and a kinetic model. While the geometric modeling of a virtual device has been given a great deal of attention, the kinetic modeling of a virtual device has rarely been brought into focus. Currently, the kinetic modeling of a virtual device is performed manually, and it takes much time and effort.

To cope with the problem, this paper proposes a procedure for the efficient construction of the kinetic model of a virtual device. The proposed procedure only focuses on general industrial devices consisting of revolute and prismatic joints without any complicated joints which can be found in special purpose machines.

Approach to Construct a Kinetic Model:

Our objective is to develop an efficient procedure to construct a kinetic model for a virtual device. The input of the procedure is the geometric model of a virtual device which consists of multiple components (links). Each link of a virtual device is represented by a solid model (a triangular mesh). Two adjacent links are connected through a joint. Most of virtual devices can be represented by using two types of joints; 1) a revolute joint describing single-axis rotational movements between two links, and 2) a prismatic joint providing a linear sliding movement between two links. Formal definitions of the two joints are as follows:

- A revolute joint requires a line (axis of the joint) in the moving body (link) to remain co-linear with a line in the fixed body, and a plane “perpendicular” to this line in the moving body maintain contact with a similar “perpendicular” plane in the fixed body.
- A prismatic joint requires that a line (axis of the joint) in the moving body (link) remain co-linear with a line in the fixed body, and a plane “parallel” to this line in the moving body maintain contact with a similar “parallel” plane in the fixed body.

Both of joints impose five constraints on the relative movement of the links, which therefore has one degree of freedom. To define a joint, we need to find the axis as well as the type (prismatic or revolute). As shown in Figure 1, the proposed procedure extracts the kinetic model (set of joints) from the given geometric model of a virtual device.

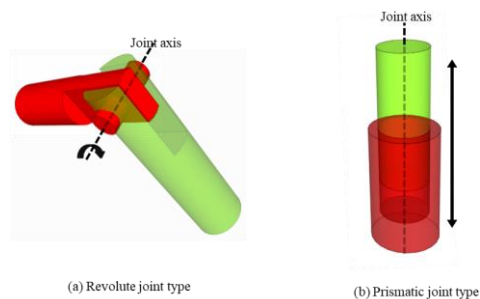


Fig. 1: A revolute joint and a prismatic joint.

When two links touch, a certain portion of their surface areas will be in contact each other. Since a joint may exist between two connected links, it is necessary to find the ‘contact surface areas (CA)’ between them to identify the joint. Friction is the force that resists motion when the surface of one object comes into contact with the surface of another. Since, friction reduces the mechanical advantage or the ratio of output to input, it is desirable to minimize the force due to friction. Without loss of generality, we can assume that the contact surface areas between two links include cylindrical shapes which minimize the force due to friction. As shown in Figure 2, we can extract the cylindrical areas from contact surface areas, and call them ‘sliding surface areas (SA)’. In the case of non-cylindrical surface areas of contact surface areas are referred to as ‘limiting surface areas (LA)’. In other words, contact surface areas consists of sliding surface areas and limiting surface areas ($CA = SA + LA$).

Without the limiting surface areas of the contact areas, it is not possible to differentiate the revolute joint from the prismatic joint. While the limiting surface areas of a prismatic joint allow the linear movement along the joint axis (Figure 3), those of a revolute joint do not allow any linear movements along the joint axis (Figure 2). Once the sliding surface areas and limiting surface areas are identified, the joint type can be determined by checking the possibility of linear movements along the joint axis. To do so, it is necessary to have an efficient algorithm to find the limiting surface areas and sliding surface areas between two links. The algorithm will be explained in the next section.

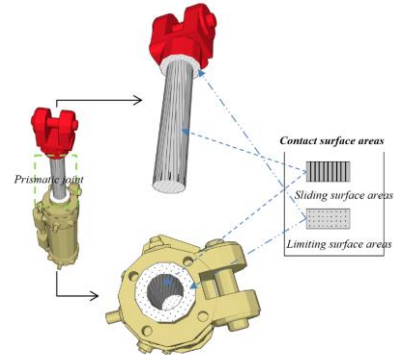
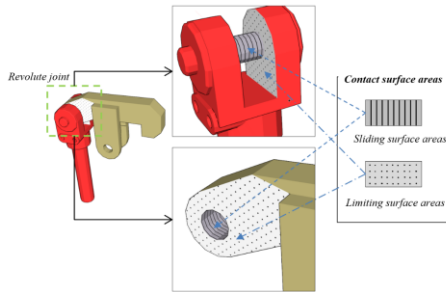


Fig. 2: Contact surface areas of a revolute joint. Fig. 3: Contact surface areas of a prismatic joint.

Algorithm to Find Sliding Part & Limiting Part:

The proposed approach consists of four steps; 1) find all cylindrical shapes from two links, 2) identify sliding surface areas from the set of cylindrical shapes, 3) identify the limiting surface areas by finding the adjacent triangles of the sliding surface areas, and 4) identify a joint between two given links. Among the four steps, the first step, finding cylindrical shapes from a solid model, is not a trivial problem when the design history information is not available. To solve the problem, we devise the concept of a T-Gauss map. By using the T-Gauss map, we can easily find cylindrical shapes by finding great circles on a T-Gauss map. The problem of identifying the joint between two links can be described as follows:

- Joint_Identification_between_Two_Links
 - Input: Two links (solid models) L_1, L_2
 - Output: *Joint_axis*, *Joint_type* (revolute or prismatic)
 - Variables:
- S_1, S_2, S_3 : set of cylinders, $Gmap1, Gmap2$: T-Gauss maps
- 1) $Gmap1$ = Construct a T-Gauss map with L_1 ;
 - 2) $Gmap2$ = Construct a T-Gauss map with L_2 ;
 - 3) $S1$ = find all cylinders by identifying the great circles of $Gmap1$;
 - 4) $S2$ = find all cylinders by identifying the great circles of $Gmap2$;
 - 5) For (every cylinder c_i of S_1) {
 - $a1$ = axis of c_i ;
 - For (every cylinder c_j of S_2) {
 - If ($a1 ==$ axis of c_j) {
 - $Joint_axis = a1$;
 - Go to Step 7;
 - 6) Return *joint_axis* & *joint_type* (NULL); // no joint
 - 7) $S3$ = extract cylinders having the same axis with the *joint_axis* from S_1 & S_2 ;
 - 8) *Limiting_part* = Find all adjacent faces of S_3 ;
 - 9) If (Limiting_part constrains a linear movement along with the *joint_axis*)
 joint_type = revolute ; Else *joint_type* = prismatic ;
 - 10) Return *joint_axis* & *joint_type* ;

As mentioned earlier, the objective of this paper is to extract a kinetic model from the geometric model of a virtual device. A kinetic model is a set of joints, and a joint has an axis and its type (prismatic or revolute). By using the joint identification algorithm, we can describe the full procedure as follows.

- Kinetic model extraction algorithm from a geometric model
 - Input: A virtual device consisting of multiple links, L_1, L_2, \dots, L_k
 - Output: KM (a kinetic model), a set of joints
 - Variables:
 - J : a joint, L_a & L_b : links
- ```

1) KM = {};
2) For (int i= 1; i < k ; i++) {
 For (int j = i+1 ; j <= k ; j++) {
 La = Li ;
 Lb = Lj ;
 Joint_Identification_between_Two_Links (La, Lb, axis, type) ;
 If (type != NULL) {
 J = new joint (axis, type);
 KM = KM + J;
 }
 }
}
Return KM ;

```

Figure 4 shows a virtual device model consisting of four links. As mentioned earlier, we need to go through four steps for each pair of links. As shown in Figure 4, it is possible to identify the sliding surface areas and the limiting surface areas by making use of the found cylindrical shapes. Once the sliding surface areas and the limiting surface areas are identified, we can determine the joint axis & type, as shown in Figure 5.

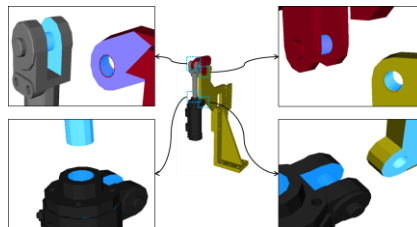


Fig. 4: Identification of contact surface areas for a given virtual device consisting of four links.

### Conclusions:

The virtual commissioning includes real controllers in the simulation of a virtual factory, a digital model imitating the physical and logical aspects of a real manufacturing system. A virtual factory consists of multiple virtual devices, and a virtual device model consists of two sub-models, a geometric model and a kinetic model. This paper proposes a methodology to extract a kinetic model from the geometric model of a virtual device.

The input of the proposed procedure is the geometric model of a virtual device which consists of multiple components (links). Each link of a virtual device is represented by a solid model (a triangular mesh). Two adjacent links are connected through a joint. Most of virtual devices can be represented by using two types of joints; 1) a revolute joint describing single-axis rotational movements between two links, and 2) a prismatic joint providing a liner sliding movement between two links.

The proposed approach, extracting a kinetic model from the geometric model of a virtual device, consists of four steps; 1) find all cylindrical shapes from two links, 2) identify sliding surface areas from the set of cylindrical shapes, 3) identify the limiting surface areas by finding the adjacent triangles of the sliding surface areas, and 4) identify a joint between two given links. Among the four steps, the first step, finding cylindrical shapes from a solid model, is not a trivial problem when the design history information is not available. To solve the problem, we devise the concept of a T-Gauss map. By using the T-Gauss map, we can easily find cylindrical shapes by finding great circles on a T-Gauss map. Although the main application of the proposed approach is an automated production system, it can be applied to any other industries requiring the virtual commissioning.

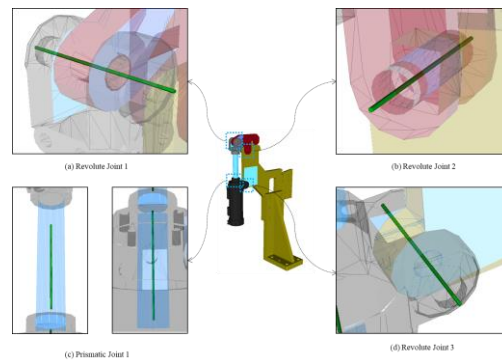


Fig. 5: Motion simulation & the virtual commissioning of a robotic cell.

#### Acknowledgements:

This work was supported by the technology innovation program (20002772, Development of the smart manufacturing collaboration system for the innovation of pipe and steel outfit and block logistics in the shipbuilding and marine) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea). Also, the research was supported by the Defense Acquisition Program Administration and the Agency for Defense Development (UD180018AD).

#### References:

- [1] Anglani, A.; Grieco, A.; Pacella, M.; Tolio, T.: Object-oriented modeling and simulation of flexible manufacturing system: a rule-based procedure, *Simulation Modeling Practice and Theory*, 10, 2002, 209-234. [https://doi.org/10.1016/S1569-190X\(02\)00100-4](https://doi.org/10.1016/S1569-190X(02)00100-4)
- [2] Choi, B.K.; Kim, B.H.: New trends in CIM: Virtual manufacturing systems for next generation manufacturing, *Current Advances in Mechanical Design and Production Seventh Cairo University International MDP Conference*, Cairo, February 15-17, 2000, 425-436.
- [3] Hoffman, P.; Maksoud, T.M.A.; Schuman, R.; Premier, G.C.: Virtual commissioning of manufacturing systems a review and new approaches for simplification, *Proceedings 24th European Conference on Modeling and Simulation*, 2010, 407-412. <https://doi.org/10.7148/2010-0175-0181>
- [4] Iwata, K.; Onosato, M.; Teramoto, K.; Osaki, S.: A modeling and simulation architecture for virtual manufacturing systems, *CIRP Annals*, 44(1), 1995, 399-402. [https://doi.org/10.1016/S0007-8506\(07\)62350-6](https://doi.org/10.1016/S0007-8506(07)62350-6)
- [5] Klingstam, P.; Gullander, P.: Overview of simulation tools for computer-aided production engineering, *Computers in Industry*, 38, 1999, 173-186. [https://doi.org/10.1016/S0166-3615\(98\)00117-1](https://doi.org/10.1016/S0166-3615(98)00117-1)
- [6] Onosato, M.; Iwata, K.: Development of a virtual manufacturing system by integrating product models and factory models, *CIRP Annals*, 42(1), 1993, 475-478. [https://doi.org/10.1016/S0007-8506\(07\)62489-5](https://doi.org/10.1016/S0007-8506(07)62489-5)
- [7] Park, S.C.; Chang, M.H.: An algorithm to extract machining volumes, *The International Journal of Advanced Manufacturing Technology*, 36(9-10), 2008, 942-949.
- [8] Ye, L.; Lin, F.: Virtual system simulation - A step beyond the conventional simulation, *Proceeding of the 22nd International Conference on Computer and Industrial Engineering*, 1997, 304-306.