Proceedings in MANUFACTURING SYSTEMS

Proceedings in Manufacturing Systems, Volume 8, Issue 1, 2013 ISSN 2067-9238

DEVELOPMENT OF ASSEMBLY/DISASSEMBLY PROCESS SIMULATION PLATFORM

Robert IACOB [1], Diana POPESCU [2,*], Nicoleta CARUTASU [3]

[1, 2, 3)] PhD, Lecturer, Machines and Production Systems Department, University “Politehnica” of Bucharest, Romania

Abstract: *Nowadays, the design of a product cannot be made without considering the Assembly/Disassembly (A/D) process simulation required for reducing the time and cost of the different assembly stages: mounting operations for manufacturing, mounting and dismounting operations for replacing/repairing components/subassemblies, automated disassembly for product recycling at the end of its life. In this context, the current paper proposes a software platform for the complete analysis and simulation of the A/D process named VIPAD – Virtual Platform for A/D Simulation. The solution developed is based on a set of two new concepts: connection interface and mobility operator, which are used to detect the interfaces between components, to identify the fasteners from an assembly, respectively to calculate the components mobility, hence generating the valid disassembly trajectories of a component relative to its neighboring ones. The mobility of components is computed through a kinematical model able to represent all the valid movements of an element from an assembly: translation, rotation and helical movements. Thus, the proposed software solution will help designers to simplify their work in generating valid A/D plans by providing complete simulations starting from the 3D CAD model of the assembly product. A further consequence of the present work is the ability to generate models with semantic data attached. This supplementary information will improve the integration of the assembly models in immersive environments by taking into account the haptic and visualization data needed.*

Key words: *assembly/disassembly, design, simulation, connection interface, mobility operator.*

1. INTRODUCTION

The main challenges in the design and development of new products are related to the necessity to meet the needs of users, ensuring compliance with the environmental legislation and, in the same time, profitability for the producer. It is important to note that all the three aspects should be accomplished in the same time.

Today, an important part of the initial phase of product design is occupied by the evaluation of the impact of different design decisions on the assembly and disassembly, these processes influencing other operations such as maintenance, repair and component/material re-using or recycling. Related concepts such as Design for Assembly (DfA), Design for Disassembly (DfD), Design for Sustainability (DfS), Product Green Design or Life-Cycle Design were coined in the last couple of years and different approaches for finding optimal assembly/disassembly (A/D) sequences for complex products (mechanical products, electrical equipment, electronic instruments, etc.) were developed or are currently under investigation [1–5]. Therefore, an automated generation of valid A/D sequences starting from the 3D CAD model of the product should simplify the engineers work and help them to take the best design decisions when modeling a product, therefore reducing the costs and time associated to assembly, repair, disassembly or recycling operations.

A previous assessment of the analysis software, simulation platforms and CAD programs showed that existing solutions do not offer the necessary information and versatility required for a complete A/D process modeling and simulation [6]. In this context, the main objective of the present paper is to describe the development steps of a Virtual Platform for A/D Simulation (ViPAD) which allows the virtual analysis and simulation of the A/D sequences. It provides a set of new functionalities – currently inexistent, through the use of a set of two innovative concepts: connection interface [7] and mobility operator [8], and it will have the following structure: import module, interface module, mobility module, types module, sequence module, immersive module and export module. The proposed platform will provide a virtual environment for interactive and real-time simulations, managing data both for static and dynamic configurations, thus assisting the engineer in establishing valid A/D plans based on the 3D representation of the product. The platform will be completely integrated in the Product Development Process (PDP).

The rest of the paper is organized as follows: section 2 reviews the most important literature references in the field, while the concepts and models used for developing the platform are presented in section 3. Section 4 de-

[*] Corresponding author: Splaiul Independenţei 313, Sector 6, 060042, Bucharest, Romania;
Tel.: 0040214029759;
Fax: 0040214029759.
E-mail addresses: *robert.iacob@gmail.com* (R. Iacob), *diana@mix.mmi.pub.ro* (D. Popescu), *nicoletacarutasu@yahoo.com* (N. Carutasu).

scribes the framework of the platform, the modules, the type of information and the specific data flow, as well as the algorithm (pseudo-code) associated to the Interface module of ViPAD. Finally, section 5 presents the conclusions and future work.

2. LITERATURE REVIEW

Modeling the A/D operations requires a significant amount of geometrical, kinematical and technological data, as well as their synthesis in order to reduce the algorithmic complexity of these processes. In the last decades, an important number of studies were made using and defining different methodologies for assemblies or mechanisms structures. However, the current methods focus primarily on the operating conditions, not offering models for the extraction or insertion of a component (i.e. a constituent of the assembly which cannot be further disassembled) from/into a mechanism as needed for A/D simulations. So, the first conclusion is that an efficient simulation application, able to simulate all the possible relative movements between components at each stage of the A/D process, was not developed so far, our ongoing research having this purpose.

One of the most comprehensive literature survey on the assembly sequencing – which is considered as the most important part of the assembly planning – is presented in [9]. Explicit (directed graph, AND/OR graph) and implicit representations (i.e. based on constructing precedence relationships between components) for assembly sequencing are described, as well as optimization algorithms. Reducing the number of possible A/D solutions and finding the optimal ones represent a mandatory problem to deal with, knowing that an increase of the number of components increase, in an exponential manner, the number of solutions. In these cases [10–12], literature presents approaches based on neural networks, genetic algorithms, simulated ant colony optimization, etc.

Also, literature presents different surveys on disassembly sequencing, simulations of components path removal, computer-aided design or other software for automatic disassembly [13–15], which are focused on describing the main approaches used by different researchers: artificial intelligence algorithms, special AND/OR graphs, heuristic algorithms, wave propagation methods, etc.

An exhaustive survey of the literature in the field of A/D process modeling presented in [6] made a clear distinction between researches focused on immersive platforms developed for assembly simulation, assembly analysis software and existing CAD software. This analysis was at the base of defining the platform development pipeline and it was used to establish the main structure of ViPAD. Thus, the main requirements for platform are:

- to detect the connection interfaces, components positions, contacts common zone;
- to generate all the valid A/D trajectories and the mobility of the components;
- to identify the functional role of components;
- to calculate the optimal or the near-optimal mounting and dismounting sequences;
- to offer a realistic simulation environment combined with haptic interaction;
- to be able to collaborate with other software – assembly models export with semantic information attached for further use.

3. NEW CONCEPTS USED IN THE RESEARCH

A realistic and complete definition of the A/D operations and valid movements can be used to determine, from the design phase of a product, the architecture of a mechanism, machine, robot or assembly tooling adapted to perform the operations in question. This is important both in the process of interactive simulations, as in the context of immersive simulations (real-time). If some types of movements are omitted, the simulations may lose some configurations and therefore they are no longer significant. Consequently, in order to have a complete simulation of the A/D process, the proposed platform should be able to analyze the 3D model and to extract all the information related to the components and assembly in a single consistent manner. These objectives are accomplished through the use of the innovative elements: connection interface and mobility operator. The developed operator is based on a kinematical model able to represent all the valid relative movements of a reference component with respect to its surrounding ones, which form a family of trajectories.

3.1. Connection interface

The connection interface is defined as the boundary across which two independent components meet, act on or communicate with each other, including information about geometrical constraints, contact surfaces relative position, common area and neighboring components. This type of information can be used to define the kinematic pairs (mechanical joints) based on the product virtual model. Some pairs can be defined by one interface (e.g. planar joint – Fig. 1), but others could require two, or more, interfaces in order to have a complete description (e.g. revolute joint – Fig. 2).

The notion of *interface* between components expresses different type of information:

- the role of the contact surfaces;
- a functional relationship;
- the relative mobility.

The detection of the interfaces between components represents an essential step of the assembly model analysis. The algorithm developed for implementation in ViPAD depends on the 3D representation of the components (assemblies) and it is important to note that the interfaces between components can be structured according to the following three categories:

- contact – the contact surfaces of the two components in contact are overlapping (see Fig. 1. – interface I_1);
- interference – the intersection volume between the two components in contact is non-vide (see Fig. 1. – interface I_2);
- gap – there is no intersection between the two components in contact, despite the fact that the components have a functional link defined at the product level.

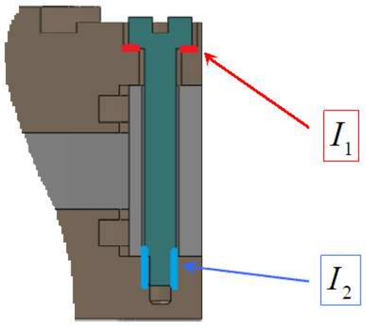


Fig. 1. Mechanical joints: I_1 = planar joint; I_2 = helical joint.

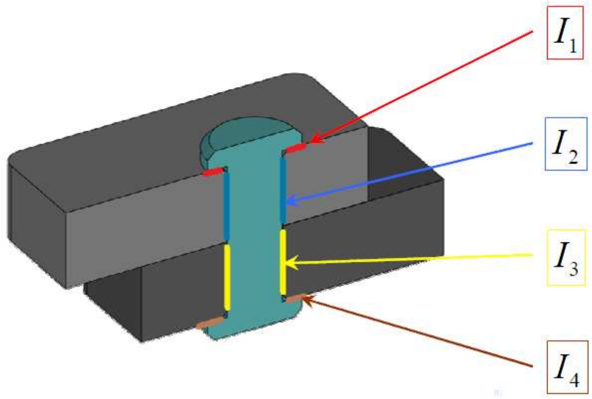


Fig. 2. Revolute joint – defined by a set of interfaces.

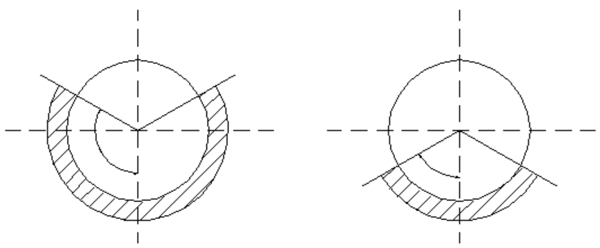


Fig. 3. Concept of partial interface:
a – complete interface (α > 90°); *b* – partial interface (α ≤ 90º).

Using a different classification criterion, the interfaces formed by curved surfaces can be divided as:

- complete interfaces – interfaces defined by closed surfaces (the sphere is the only closed surface);
- partial interfaces – interfaces defined by open surfaces (surfaces limited by at least one closed curve that defines a finite area: plane, cylinder or cone) (Fig. 3).

It is important to note that partial interfaces may produce supplementary translation movements and, therefore, they are distinguished from the standard configurations. Thus, these types of interfaces are considered separately using a distinct method for common zone detection.

3.2. Kinematical model

In kinematics, the contact type and the geometric nature of its corresponding surfaces can help characterizing the nature and the kinematic parameters between two components in a mechanical assembly. This may be helpful for VR simulations, for maintainability, when haptic devices are of interest, in order to avoid side effects due to configurations where surfaces of two distinct components are close to each other and may generate some collisions.

In order to define the set of relations between elementary components of a product, it is necessary to determine the relative mobilities of the assembly components, i.e. the authorized displacements between components. These displacements fall into two complementary categories: infinitesimal and finite displacements. Infinitesimal ones correspond mainly to translations and are modeled on the basis of kinematic relations between components. Indeed, an infinitesimal translation is enough to remove/establish the contact between two components. At the opposite, contact areas along an arc of circle are not able to remove/add a contact.

Regarding the determination of the families of trajectories for a component, the screws and the algebra of quaternions are among the methods most often used [16, 17]. However, screws are not homogeneous in writing combinations of rotations and translations and hence, they are not suited to describe general helical movements and their combinations when they are positioned arbitrarily with respect to each other. Movements described with homogeneous matrix algebra are not suited as well since rotation axes cannot be easily characterized. Indeed, the dual algebra of quaternions offers the advantage of representing at the same time various possible movements (rotation, translation, helical movements) in a unified manner. This algebra incorporates both the rotation axis and the direction of translation together with the intrinsic parameters of these movements. Based on quaternions, it is possible to represent all the movements needed to remove/generate the contact between adjacent components and to specify the corresponding operator, which justifies their use to describe the transformations.

A general transformation $\hat{T}$ of screw type can be described using a dual angle $\hat{\psi}$ and a dual vector $\hat{w}$ as:

$$\hat{T} = \left\{ \cos\frac{\hat{\Psi}}{2}, \sin\frac{\hat{\Psi}}{2} \cdot \hat{w} \right\}. \tag{1}$$

The transformation $\hat{T}$ is around the dual angle $\hat{\psi} = \psi + \varepsilon \cdot d$, where d is the amplitude of the translation, along the dual vector $\hat{w} = \vec{n} + \varepsilon \cdot \vec{m}$, where $\vec{n}$ is the rotation axis, $\vec{m}$ is the vector $\vec{m} = \vec{p} \wedge \vec{n}$, defining $\vec{n}$ in Plückerian coordinates. $\vec{p}$ is the vector positioning $\vec{n}$ with respect to the origin. It should be noted that this form looks exactly like the formula used for simple quaternions, but using the dual angle and the unit dual vector. The detailed form of eq. (1) is:

$$\begin{aligned} \hat{T} = & \left\{ \cos\frac{\Psi}{2}, \left(\sin\frac{\Psi}{2} \cdot \vec{n} \right) \right\} + \\ & + \varepsilon \cdot \left\{ -\frac{d}{2} \cdot \sin\frac{\Psi}{2}, \left(\vec{m} \cdot \sin\frac{\Psi}{2} + \frac{d}{2} \cdot \vec{n} \cdot \cos\frac{\Psi}{2} \right) \right\}. \end{aligned} \tag{2}$$

In order to characterize the family of trajectories resulting from the combined effect of two different contacts: C_1 and C_2, a bi-quaternion is associated to each elementary contact between a reference component and its neighboring ones. As a consequence, the resulting family of trajectories defines the possible motions of the reference component with respect to the whole set of contacts considered, being mandatory to transform the family of trajectories describing the contact C_2 into the reference frame associated to a reference contact C_1, arbitrarily chosen. Therefore, to define the compatible trajectories among the two bi-quaternions corresponding to C_1 and C_2, a change of reference frame is necessary.

The family of trajectories for the first contact C_1 is defined by the dual quaternion $\hat{Q}.\hat{Q}$ is written in the reference frame $S_1(O_1, \vec{X}_1, \vec{Y}_1, \vec{Z}_1)$ and characterized by the rotation angle θ, with $0 \le \theta \le 2 \cdot \pi$ around the unit vector $\vec{u}(u_1, u_2, u_3)$ to define a finite rotation. The translation v takes place along the vector $\vec{u}(u_1, u_2, u_3)$, the reference point considered for this family is the origin O_1, hence all movements (translations, rotations, helical) have directions passing through O_1.

The bi-quaternion associated to this family of trajectories is:

$$\hat{Q} = \left\{ \cos\frac{\hat{\theta}}{2}, \sin\frac{\hat{\theta}}{2} \cdot \hat{u} \right\} \tag{3}$$

with: $\hat{\theta} = \theta + \varepsilon \cdot v$ and $\hat{u} = \vec{u} + \varepsilon \cdot \vec{0}$, represented as bi-vectors. The detailed form is:

$$\hat{Q} = \left\{ \left\{ \cos\frac{\theta}{2}, \left(\sin\frac{\theta}{2} \cdot \vec{u} \right) \right\} + \varepsilon \cdot \left\{ -\frac{v}{2} \cdot \sin\frac{\theta}{2}, \left(\frac{v}{2} \cdot \vec{u} \cdot \cos\frac{\theta}{2} \right) \right\} \right\} \tag{4}$$

Similarly, the second family of trajectories associated with the contact C_2 is defined by the dual quaternion $\hat{Z}$ in the $S_2(O_2, \vec{X}_2, \vec{Y}_2, \vec{Z}_2)$ reference frame. The rotation of angle δ takes place around the unit vector $\vec{r}(r_1, r_2, r_3)$ and the translation t is applied along the vector $\vec{r}(r_1, r_2, r_3)$.

The bi-quaternion associated to this family is for the same reasons as previously:

$$\hat{Z} = \left\{ \cos\frac{\hat{\delta}}{2}, \sin\frac{\hat{\delta}}{2} \cdot \hat{r} \right\}, \tag{5}$$

where: $\hat{\delta} = \delta + \varepsilon \cdot t$ and $\hat{r} = \vec{r} + \varepsilon \cdot \vec{0}$ are represented as bi-vectors. The detailed form of the second family of trajectories is:

$$\hat{Z} = \left\{ \left\{ \cos\frac{\delta}{2}, \left(\sin\frac{\delta}{2} \cdot \vec{r} \right) \right\} + \varepsilon \cdot \left\{ -\frac{t}{2} \cdot \sin\frac{\delta}{2}, \left(\frac{t}{2} \cdot \vec{r} \cdot \cos\frac{\delta}{2} \right) \right\} \right\} \tag{6}$$

The dual quaternions $\hat{Q}$ and $\hat{Z}$ can be regarded as kinematics transformations, hence the parameters describing the mobilities at C_1 (angle θ and translation amplitude v) and C_2 (angle δ and translation amplitude t) are all functions of a parameter that be considered as the time.

The proposed method considers that the first family of trajectories at C_1 is a predefined one, i.e. belonging to the set of elementary contacts, while the second is a general one. In the present case, it is the bi-quaternion associated to the second family which will be transferred to the C_1 contact reference frame $S_1(O_1, \vec{X}_1, \vec{Y}_1, \vec{Z}_1)$. The family C_2 itself can represent either a predefined family or a family resulting from a previous combination of contact mobilities. However, after the change of reference frame, the dual quaternion $\hat{Z}$, expressed in the reference frame $S_1(O_1, \vec{X}_1, \vec{Y}_1, \vec{Z}_1)$, can be considered as a general one. Indeed, this scheme is the basic one which allows to combine iteratively all the contacts attached to a reference component so that its resulting mobility can be characterized for the simulation purposes.

For the change of reference frame, a general geometric transformation $\hat{T}$ is carried out. The detailed form of the transformation was presented in equation (1).

As mentioned previously, C_1 is defined by the dual quaternion $\hat{Q}$. A point P of the reference component transformed by the dual quaternion $\hat{Q}$ in the reference frame $S_1(O_1, \vec{X}_1, \vec{Y}_1, \vec{Z}_1)$ is:

$$\hat{P}_{S1}^{Q} = \hat{Q} \cdot \hat{P} \cdot \hat{Q}^{*}. \tag{7}$$

The family of trajectories attached to C_2 being defined by the dual quaternion $\hat{Z}$, the same point P of the reference component transformed by the dual quaternion $\hat{Z}$ in the reference frame $S_2(O_2, \vec{X}_2, \vec{Y}_2, \vec{Z}_2)$ is:

$$\hat{P}_{S2}^{Z} = \hat{Z} \cdot \hat{P} \cdot \hat{Z}^{*}. \tag{8}$$

To express the point $\hat{P}_{S2}^{Z}$ in the reference frame $S_1(O_1, \vec{X}_1, \vec{Y}_1, \vec{Z}_1)$, a change of reference frame is necessary, which is materialized by the geometric transformation:

$$\hat{P}_{S1}^{T \cdot Z} = \hat{T} \cdot \hat{P}_{S2}^{Z} \cdot \hat{T}^{*}. \tag{9}$$

The transformations $\hat{P}_{S1}^{Q}$ and $\hat{P}_{S1}^{T \cdot Z}$ are expressed in the same reference frame. In fact, in order to have a compatibility of the movements, both trajectories must be identical, therefore:

$$\hat{P}_{S1}^{Q} = \hat{P}_{S1}^{T \cdot Z}. \tag{10}$$

From the above analysis, it comes the reference equation:

$$\hat{Q} \cdot \hat{P} \cdot \hat{Q}^{*} = (\hat{T} \cdot \hat{Z}) \cdot \hat{P}_{S2}^{Z} \cdot (\hat{T} \cdot \hat{Z})^{*}, \tag{11}$$

which is equivalent to:

$$\hat{Q} = \hat{T} \cdot \hat{Z}. \tag{12}$$

Showing that the compatibility of families of trajectories reduces to the analysis of the product of the dual quaternion of C_2 with the transformation $\hat{T}$ so that it covers the families of trajectories of C_1. In the Eq. (12), the dual quaternions $\hat{Q}$ and $\hat{Z}$ are 'kinematical', while the transformation $\hat{T}$ is a geometric one, the latter being the most general possible.

In order to analyse the movements (rotations, translations, helical movements), the system formed from equation (12), detailed below, is resolved.

Real part equality:

$$\left(\hat{Q}\right)_R = \left(\hat{T}\cdot\hat{Z}\right)_R . \tag{13}$$

Dual part equality:

$$\left(\hat{Q}\right)_D = \left(\hat{T}\cdot\hat{Z}\right)_D . \tag{14}$$

Thus, the analysis of the movements is addressed in order to characterize the compatibility of trajectories for each type of trajectory:

- Rotations – the movements are compatible if the axes are coaxial and origins of the rotation vectors that are coincident or shifted with respect to each other along their common direction.
- Translations – the movements are compatible if they follow the same direction and have identical absolute values.
- Helical movements – in order to have a compatibility between two helical movements, the two vectors defining the axes movements must be coaxial; in addition, the 'speeds' attached to C_1 and C_2, i.e. the tangent to trajectories compatible with C_1 and C_2, must be the same and the two pitches must be equal as well. Also, the rotation angles can be shifted with respect to each other with a fixed value, i.e. the angle ψ.

3.3. Mobility operator

The mathematical model presented in the previous section is able to describe and combine all the families of trajectories associated to the interfaces from different components of a mechanical system.

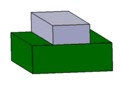
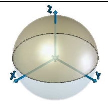
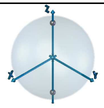
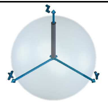
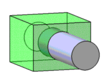
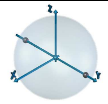
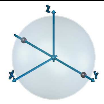
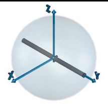
Knowing that the configuration of the contact zone between the surfaces of the assembly components determines the valid trajectories by generating a geometric domain which describes the set of possible movements, a complete geometrical representation of the families of trajectories through a geometrical model is required. The combination between these two models can form, in a real-time simulation environment, the basis for determining and displaying, at each moment, the valid movements between different components or subassemblies [8].

The developed mobility operator can describe and represent all the valid displacements – translations, rotations and helical movements, for a component or for the entire product. It is deployed through the unit sphere (and unit ball) concept, which is a powerful tool for geometrical and graphical representations. The rotations and translations are depicted using a unit sphere (spherical valid domains), while the helical movements are geometrically illustrated using a unit ball (volumetric valid domain).

Table 1 is a graphical representation of the mobility operator for two types of common connections: plane

Table 1

Graphical representation of families of trajectories

description	name	translations	rotations	helical
	planar interface			
	pin-hole interface			

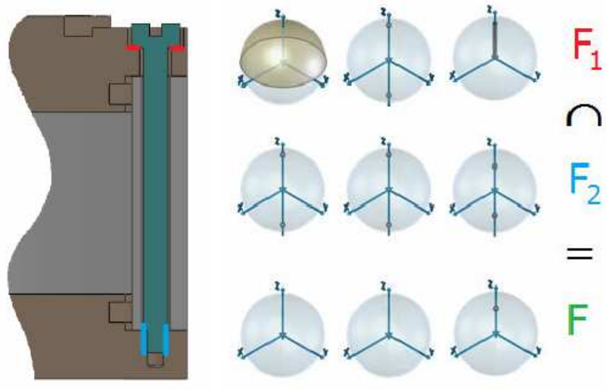


Fig. 4. Implementation example of the mobility operator.

interface and pin-hole interface. This example shows the mobility (removal directions from the assembly) of components, but it can be used as well for generating assembly paths for components.

The operator can be used to represent the mobility of a single interface (complete or partial) or component, or to compute the mobility of a set of components (subassembly or assembly). For the moment, the operator is deployed for standard surfaces, but a generalization of the proposed method is considered by including the general surfaces.

In order to clarify the geometrical method used, a simple example is presented here after (Fig. 4). The assembly consists of four components: a body, two covers and a rod (anchor).

To analyze the mobility of the *rod (anchor)* component, all the existing interfaces have to be defined:

- interface I_1 (red) – planar type contact between the (rod) anchor head and the top cover;
- interface I_2 (blue) – helical pin-hole type contact between the rod (anchor) and the lower cover.

The resulting mobility of the rod (anchor) component, noted F, is determined as the union of the three mobilities corresponding to three types of movements: translations (noted M_T), rotations (M_R) and helical movements (M_H), which are calculated as the intersections of the families of trajectories associated with to each interface: F_1 for I_1 and F_2 for I_2.

The total mobility of the rod (anchor) component can be determined and represented using the unit sphere and the unit ball as:

$$F = F_1 \cap F_2 , \tag{15}$$

$$F = M_T \cup M_R \cup M_H . \tag{16}$$

4. VIPAD plaform

The synthesis of current research [6] showed that an A/D simulation is subjected to different shape representations, i.e. B-Rep NURBS models or polyhedral representations needed for immersive simulations. Although there are some 3D models built with VR tools and used only for VR testing applications, most of the mechanical products are designed using a standard 3D CAD program.

Almost all the CAD assembly modules have a single objective – the relative positioning of components using a set of standard constraints. It is important to note that different combinations of constraints can produce the same result (spatial positioning of the components) and these constraints do not express directly the possible relative motion between components. Moreover, the constraint (mate) concept was developed as a partial solution to build assemblies from parts, but it requires various improvements in order to offer a better description of the assembly process. Thus, the existing modules should be upgrated with new elements regarding the mobility between components.

Furthermore, the information about geometrical constraints is not used to define the relative mobility of the components and the proposed constraints are related to the position of components, but do not explicitly represent the contacts between components. Therefore, this type of information is interesting but its effectiveness is limited and its transfer through data exchange standard formats is not currently possible. The positional constraints refer only to the reciprocal position of components and these data are neither sufficient nor consistent with the requirements for the identification of contacts or interfaces. Moreover, they are not intrinsic to the definition and characterization of the contacts and their commune zone associated.

In this context, the proposed analysis and simulation platform ViPAD is composed of seven modules, as represented in Figure 5, along with the input/output data for each of them. The modules are integrated in an innovative environment (Fig. 6) which provides the design engineers the necessary tools for optimizing the A/D modeling processes, offering useful information for the whole product lifecycle, from design and fabrication to recycling. In order to provide a complete set of data, the concepts mentioned in the previous chapter: connection interface and mobility operator, were implemented in two modules of ViPAD, namely the *Interface* module and the *Mobility* modules.

4.1. Import Module & Data Representation

In the first stage, the 3D CAD model of the product is used to generate the list of components and the associated geometry: surfaces, curves, points, which constitutes the input information for the application.

Based on a series of tests on the main industrial CAD software CATIA and Solid Works developped by Dassault Systemes, NX and Solid Edge produced by Siemens, we can mention that the STEP standard is the most effective way of transfering 3D models of products.

In industry there are other formats like IGES, VRML etc. used for the import-export operations, but they are less robust because they do not describe the topology of a solid model. In this approach, STEP format was also chosen because of its ability to incorporate the description of the analytic surfaces. Using this detailed representation of the components and assemblies, the proposed platform can benefit from all the geometrical data available in the STEP files starting with the first phase – 3D model import.

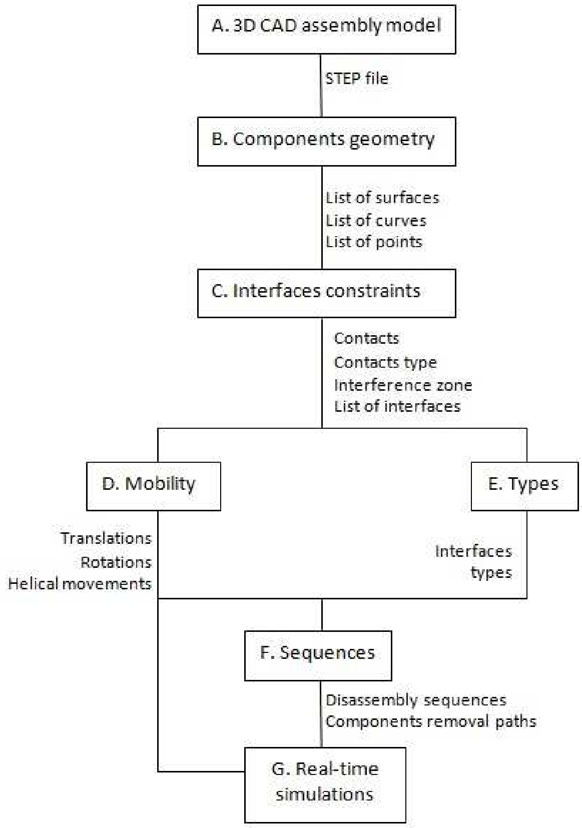


Fig. 5. ViPAD architecture and data flow.

Today, different shape representations and model variants are produced by different CAD modelers even though they are quoted as standard format – exported as STEP files. Thus, in order to handle this diversity and to offer a way for an explicit representation of the semantic information attached to a shape, the platform uses a new data structure.

4.2. Interface Module

The main purpose of this research being to offer an intelligent tool to aid engineers in the design process, the proposed platform offers a set of functionalities currently inexistent. Thus, the *Interface* identification module is able to automatically identify the following information: geometric constraints, contact surfaces relative position, common area, and to combine this data in a complete set of interfaces for an assembly. This information is added to the product data structure and it will be further used in other analysis modules. More, it can be exported, together with the model data, in order to obtain a complete representation of the assembly for the PDP.

The list of interfaces between components contains thirteen basic types based on the functional surfaces of reference: Fixed Fit (FXD - ENC), Revolute (RVL -

PVT), Prismatic (PRS - GLS), Cylindrical (CLD - PVG), Helical (Screw) (HLJ - PGH), Spherique Doigt (SDJ - SPD), Spherical (SPH - RTL), Planar (PLN - APP), Linear Annular (LNA - LNA), Linear Rectiligne (LNR - LNR), Pin (Point) (PNT - PNC), Complex (CPX), Undefined Type Contact (UTC - CND). Mention should be made that the partial interfaces represent a special category which admits additional mobilities due to the partial covering of the surfaces in contact.

In order to have a structured searching algorithm, a classification of the interfaces in five categories is proposed:

- Elementary interfaces → defined by functional surfaces of the same type: {Fixed Fit (FXD), Cylindrical (CLD), Helical (HLJ), Spherical (SPH), Planar (PLN)}
- Standard interfaces → defined by functional surfaces of different type: {Linear Annular (LNA), Linear Rectiligne (LNR), Point (PNT)}
- Partial interfaces → derived type that admits additional mobilities: {Cylindrical partial (CLDp), Helical partial (HLJp), Spherical partial (SPHp), Linear Annular partial (LNAp)}
- Associative interfaces → defined by an association of two or more interfaces: {Revolute (RVL), Prismatic (PRS), Spherique Doigt (SDJ)}
- Complex interfaces → defined by two surfaces of general type having a contact between them: {Complex (CPX), Undefined Type Contact (UTC)}

The output of the *Interface* module consists in the following data: list of interferences between assembly components, contacts type, common zone. All of these are used, in the downstream, as input data for the *Mobility* and *Types* modules or even further for the real-time simulation of the A/D operations in an immersive environment (*Real Time Simulations* module, see fig.5).

The algorithm implemented in the *Interface* module is described here after.

```
SR: create list of body intersections
for each Component in List of components to check
    if (intersection (component (i and j) ) ǂ 0)
    then Add components (i and j) to List of Intersections
SR: search for elementary interfaces for each Element
     in List of Body intersections
    search for cylindrical interfaces (CLD)
        if (same parameters and different orientation)
        then
            Identify common zone
            Add interface to List of CLD interfaces
    search for helical interfaces (HLJ)
        if (helical conditions)
        then
            Identify common zone
            Add interface to List of HLJ interfaces
    search for spherical interfaces (SPH)
        if (same parameters and different orientation)
        then
            Identify common zone
            Add interface to List of SPH interfaces
    search for planar interfaces (PLN)
        if (same parameters and different orientation)
        then
            Identify common zone
            Add interface to List of PLN interfaces
SR: search for standard interfaces
for each Element in List of Body intersections
    search for linear_annular interfaces (LNA)
        if (linear_annular conditions)
        then
            Identify common zone
            Add interface to List of LNA interfaces
    search for linear_rectiligne interfaces (LNR)
        if (linear_rectiligne conditions)
        then
            Identify common zone
            Add interface to List of LNR interfaces
    search for point interfaces (PNT)
        if (point conditions)
        then
            Identify common zone
            Add interface to List of PNT interfaces
SR: search for partial interfaces
for each Interface in List of CLD interfaces
    if (partial cylindrical conditions)
        then
            Add interface to List of CLDprt interfaces
            Remove interface from List of CLD interfaces
for each Interface in List of HLJ interfaces
    if (partial helical conditions)
        then
            Add interface to List of HLJprt interfaces
            Remove interface from List of HLJ interfaces
for each Interface in List of SPH interfaces
    if (partial spherical conditions)
        then
            Add interface to List of SPHprt interfaces
            Remove interface from List of SPH interfaces
for each Interface in List of LNA interfaces
    if (partial linear_annular conditions)
        then
            Add interface to List of LNAprt interfaces
            Remove interface from List of LNA interfaces
SR: search for complex interfaces
for each Element in List of Body intersections
    search for complex interfaces (CPX)
        if (complex conditions)
        then
            Add interface to List of CPX interface
```

A mention should be made related to the searching subroutine for associative interfaces – the code is developed in the *Types* module because the associative interfaces define, generally, the functional role of components.

4.3. Mobility Module

The *Mobility* module was developed to generate all the valid A/D trajectories. It represents a step forward, in comparison with the current approaches, because it is based on an innovative method and operator that can compute and represent all the general movements: translations, rotations, helical movements.

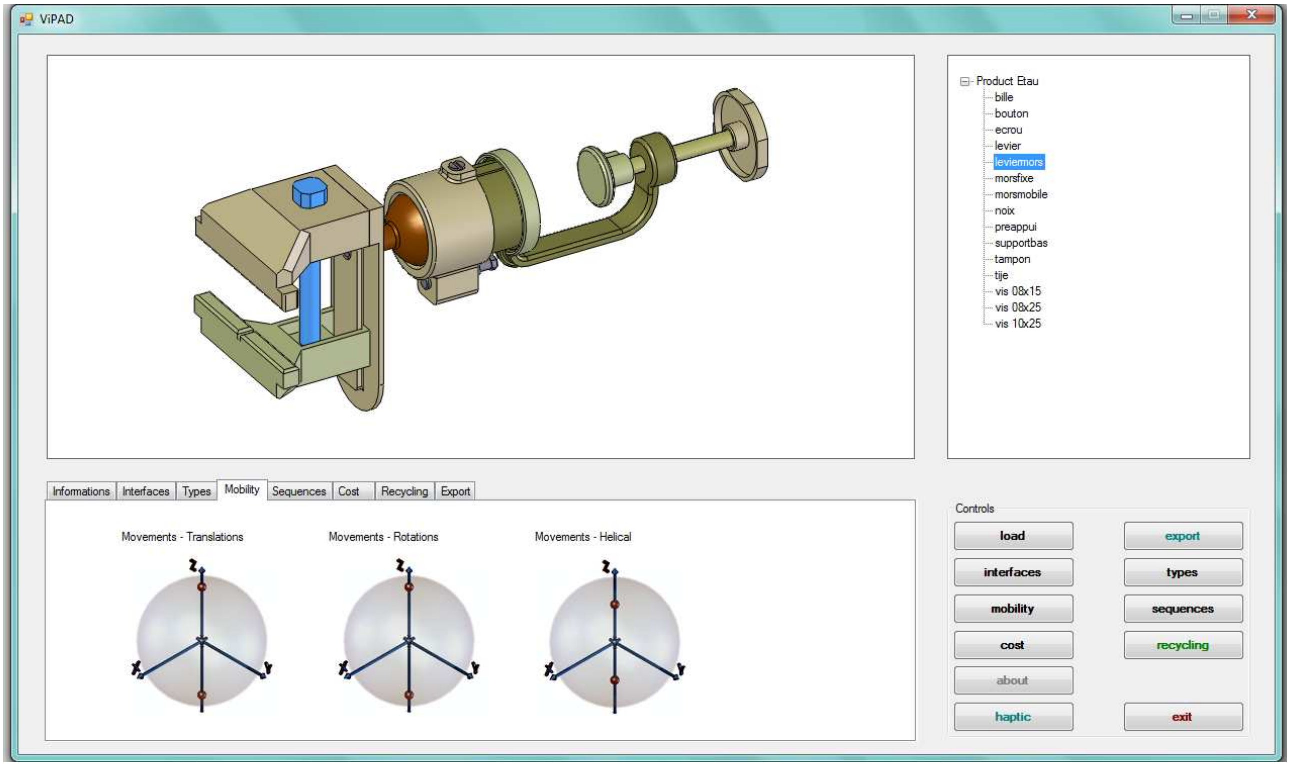


Fig. 6. ViPAD interface.

The interface information represents the input data for the *Mobility* module. The developed operator is based on the combination method previously presented (sections 3.2. and 3.3.) and it can compute the mobility of a component, a set of components or of an entire assembly. For visualization purpose, the unit sphere and unit ball concepts are used.

4.4. Types Module

The *Types* module is able to find all the components that have a specific functional role. Using the information generated by the interface module and a set of rules – defined for different types of components (e.g. screw, washer etc.), the elements with a specific functional role are automatically identified.

For example, a set of rules defined for the rivet fastener will allow the automatic identification of all the rivets from an assembly model. This method can be applied for any type of component, depending on the rule definition. The platform contains an initial predefined set of rules for standard components – fasteners (bolt, nut, rivet, rod, screw), which can be extended, at any moment, through a rule definition method, in order to identify different types of components.

Below are detailed, as example, two standard rules:

- Rivet type rule: a component that has two Planar interfaces [PLN(1), PLN(2)] and two Cylindrical interfaces [CLD(1), CLD(2)], such as the Planar interfaces normal is coaxial with the Cylindrical interfaces axis.
- Screw type rule: a component that has one Planar interface [PLN(1)] and a set of Cylindrical interfaces [CLD(n), with n > 1], such as the Planar interfaces normal is Coaxial with Cylindrical interfaces axis.

All the components of an assembly that have a functional role associated benefit from an entire functional description with interfaces having functions assigned, these functions being clearly identified through faces, edges, and vertices on the boundary of these components. In addition, these components can be rapidily evaluated in the downstream modules.

4.5. Sequence Module

Having all the information related to the components mobility data and the list of all identified fasteners in the assembly, the *Sequence* module will be able to determine the sequence of assembly or disassembly of a component or of the whole product. The application will generate all the feasible sequences and it will identify the best one (or the near-optimal one) according to the criteria set by user.

Researchers have developed different methods based on: wave propagation [13], geometric relationships between components [14], topological information [15], but their efficiency is limited because they used only a part of the geometrical data available in the 3D models and some of the operations still have to be done manually.

The developed disassembly method is based on an improved layer technique – the components are removed one by one from the assembly, starting with those placed in the exterior, in the remotest positions from a base component. This layered disassembly method is augmented with the identification of the type of connection interfaces between components, which establish a clear distinction between components and fasteners. Therefore, it is possible to disassembly a component according to the following rule: a component can be extracted from the assembly by firstly removing the fasteners which connects it with the adjacent components, and then by breaking the connection interfaces (described using standard or combinations of standard interfaces such as plane, pin-hole, dove-tail etc.) with these components.

In comparison to other approaches, which are using different concepts to detect some removal directions, the proposed method is based on the mobility operator for representing translations, rotations and helical movements, thus providing the base for a complete description of the components relative mobility.

Using the information related to the component mobility represents an important advantage, not only for reducing the computational effort and time, but as well for providing an improvement in collision detection algorithms and kinematic constraints management. Therefore, the disassembly sequences can be rapidly identified and their simulation in immersive environments improved, when haptic devices are used.

Another possible application of this *Sequence* module would be to store, on a device attached to the product, the optimal sequence of disassembly for the product or only for a valuable component in terms of recycling. This sequence will be read, in the recycling stage, by a disassembly system (robot), thus creating the frame for an automatic disassembly process. The solution is an extension of the idea proposed in [18].

4.6. Immersive Module

The immersive module, currently under development, using the model data and the generated information, will offer a realistic simulation of the A/D process. It will contain the following main features:

- real-time simulation of A/D operations;
- A/D sequences validation and editing;
- haptic interaction;
- two modes interaction: free mode and kinematic guided movement.

The ViPAD software is developed as an external application that can be attached to CAD software for a better and faster analysis of assemblies. Being a CAD independent platform, it can incorporate different haptic devices requiring only a set of minor changes. Thus, the simulations could be performed using multiple devices.

The complete characterization of the components relative mobility is a key element contributing to A/D simulations. This is a complement to the geometric location of contacts to express the effective relative movements between neighboring components, being very helpful for Virtual Reality (VR) simulations addressing the simulation of maintainability operations, when haptic devices are used. There, it is important to avoid side effects due to configurations where surfaces of two distinct components are close to each other. In such a configuration, collision detection algorithms are based on polyhedral models of the components and many collisions can occur depending on the respective positions of the cylindrical surfaces, thus generating unacceptable vibrations, e.g. during the insertion phase of one component into another when their nominal dimensions are equal.

Complementarily, having the correct kinematic mobilities between the assembly components would allow the servo control of the haptic device to generate the trajectories as specified by the corresponding kinematic joint (planar, cylindrical, spherical etc.), thus avoiding the undesired effects and improving its usage.

The proposed module is largely automated and it will provide a smarter way to manage collisions, using all the information previously computed: list of interfaces, list of fasteners, components mobility, A/D sequences etc.

4.7. Export Module

The main objective of the proposed platform is the simulation of A/D operations in order to offer complete information about the A/D process. However, a very important aspect of any analysis and simulation program is the possibility to share information. Therefore, the platform proposed in this research has to be able to collaborate with other software through an export module. This one will offer the possibility to export the models with different information (semantic data) attached for further use.

The semantic data model for a 3D assembly model it is a conceptual data model in which geometric and semantic information is included. This means that the model describes completely the relations between the components of an assembly. This type of information can be exported through an augmented version of a standard exchange file format.

5. CONCLUSIONS

The research presented describes a solution for analysis and simulation of the A/D processes in a virtual environment – ViPAD platform, which has several advantages compared to other approaches described in the literature. One of these advantages is the integration, in a set of interfaces for a product, of all information regarding the mating constrains, neighboring components, contact surfaces relative position and common zone. Based on this set of interfaces and on a defined set of rules, standard components with functional role can be identified.

Another advantage of ViPAD is the automatic generation of the valid A/D trajectories of a component from an assembly with respect to its surrounding components. Thus, components relative mobility is computed through a mobility operator and used for generating realistic simulations of disassembly paths.

The new concepts of connection interface and mobility operator on which ViPAD is based, as well as its architecture, modules and dataflow are also presented in this paper in order to offer a complete perspective over the proposed platform.

Further work will address the complete development of the remaining modules and a better integration of the concepts and algorithms previously defined.

ACKNOWLEDGEMENTS: This work was supported by CNCSIS-UEFISCSU, project number PN-II

RU 233/2010, project title: "Assembly/Disassembly Process Modeling", project type: "Research projects for stimulation of the founding/forming of young independent research teams".

REFERENCES

[1] T. Dong, L. Zhang, R. Tong, J. Dong, *A hierarchical approach to disassembly sequence planning for mechanical products*, International Journal of Advanced Manufacturing Technology, vol. 30, no. 5-6, 2003, pp. 507–520,

[2] A.J.D. Lambert, S.M. Gupta, *Methods for optimum and near optimum disassembly sequencing*, International Journal of Production Research, vol. 46, no. 11, 2008, pp. 2845–2865.

[3] D. Bobkova, M. Pekarcikova, *Modeling of the disassembly processes for small electric appliances*, Journal of Proceedings of the International Conference on Manufacturing Systems – ICMaS, vol. 3, 2008, pp. 271–274.

[4] S.T. Puente, F. Torres, O. Reinoso, L. Paya, *Disassembly planning strategies for automatic material removal*, International Journal of Advanced Manufacturing Technology, vol. 46, no. 1, 2010, pp. 339–350.

[5] A. ElSayed, E. Kongar, S.M. Gupta, *A genetic algorithm approach to End-Of Life disassembly sequencing for robotic disassembly*, Proceedings of the Northeast Decision Sciences Institute Conference, Virginia, USA, pp. 402-408, 2010.

[6] R. Iacob, D. Popescu, P. Mitrouchev, *Assembly/Disassembly analysis and modeling techniques: A review*, SV Journal of Mechanical Engineering, vol. 58, no. 11, 2012, pp. 653–664.

[7] R. Iacob, P. Mitrouchev, J.C. Léon, *A simulation framework for assembly/disassembly process modeling*, Proceedings of the ASME: IDETC & CIE Conference, Nevada, USA, pp. 1017–1027, 2007.

[8] R. Iacob, P. Mitrouchev, J.C. Léon, *Assembly simulation incorporating component mobility modeling based on functional surfaces*, International Journal on Interactive Design and Manufacturing, vol. 5, no. 2, 2011, pp. 119–132.

[9] P. Jimenez, *Survey on assembly sequencing: A combinatorial and geometrical perspective*, Journal of Intelligent Manufacturing, DOI: 10.1007%2Fs10845-011-0578-5, 2012.

[10] X.F. Zha, S.Y.E. Lim, *Intelligent design and planning of manual assembly workstations: A neuro-fuzzy approach*, Computers & Industrial Engineering, vol. 44, 2003, pp. 611–632.

[11] E. Kongar, S.M. Gupta, *Disassembly sequencing using genetic algorithm*, International Journal of Advanced Manufacturing Technology, vol. 30, no. 5-6, pp. 497-506, 2006

[12] H. Shan, S. Zhou, Z. Sun, *Research on assembly sequence planning based on genetic simulated annealing algorithm and ant colony optimization algorithm*, Assembly Automation, vol. 29, no. 3, 2009, pp. 249–256.

[13] J. Yi, B. Yu, L. Du, C. Li, D. Hu, *Research on the selectable disassembly strategy of mechanical parts based on the generalized CAD model*, International Journal of Advanced Manufacturing Technology, vol. 37, 2008, pp. 599–604.

[14] S.S. Smith, W.H. Chen, *Rule-based recursive selective disassembly sequence planning for green design*, Advanced Engineering Informatics, vol. 25, 2011, pp. 77–87.

[15] R. Vigano, G.O. Gomez, *Automatic assembly sequence exploration without precedence definition*, International Journal on Interactive Design and Manufacturing, DOI: 10.1007%2Fs12008-012-0165-9, 2012.

[16] P. Alba, J.M. McCarthy, *Dual quaternion synthesis of parallel 2-TPR robot*, Proceedings of the Workshop on Fundamental Issues and Future Research on Mechanisms, 2002, pp. 101–108.

[17] J. Rooney, *Multivectors and quaternions in rigid body rotation: Clifford vs. Hamilton*, 12th IFToMM World Congress, Besançon, France, June 18-21, 2007, www.iftomm.org/iftomm.

[18] G. Ostojic, S. Stankovski, D. Vukelic, M. Lazarevic, J. Hodolic, B. Tadic, S. Odri, *Implementation of automatic identification technology in a process of fixture assembly/disassembly*, SV – Journal of Mechanical Engineering, vol. 57, no. 11, 2011, pp. 819–825.