Generic Visual Simulation of Manufacturing Equipment

Ch. Sweta Dhaveji  
Department Of Manufacturing Engineering  
Missouri University of Science and Technology  
Rolla, Missouri, USA

Todd E. Sparks  
Department of Mechanical and Aerospace Engineering  
Missouri University of Science and Technology  
Rolla, Missouri, USA

Jianzhong Ruan  
Production Innovation and Engineering LLC  
Rolla, Missouri, USA

Frank W. Liou  
Department of Mechanical and Aerospace Engineering  
Missouri University of Science and Technology  
Rolla, Missouri, USA

ABSTRACT

Simulation of a machine is very important before laser metal deposition is performed, as a tool to check collision detection and validate deposition result. There are several kinds of machines that are used for laser deposition and hence there is a need for a generalized concept for visual simulation of all kinds of machines. This paper presents the research conducted on describing each machine configuration in a generic format. A parent - child list and a dependency list obtained from the machine configuration are utilized to form a generic format. Such a format can be used to describe linear and rotational motion of the machines parts. This method has been tested on various examples to demonstrate its robustness and efficiency.

1 INTRODUCTION

Layered Manufacturing (LM) technology, also known as Rapid Prototyping (RP) is a process that involves adding raw material, in layers, to create a solid part directly from a CAD model instead of removing the material as in the case of traditional subtractive manufacturing processes such as machining. LM processes fabricate a physical part in an additive fashion, layer by layer. Metal rapid prototyping is a technique that can produce fully functional parts directly from a CAD. Direct laser deposition is capable to fabricate fully dense metal parts directly from the CAD model. Such a process has drawn interest from aerospace, heavy machinery and other industries. Due to its complexity, this process requires an automatic planning system to drive.

Laser Aided Manufacturing Process (LAMP) lab is using a Fadal 5-axis CNC machine (model VMC3016) as shown in figure 1, as the motion driver. The deposition nozzle is mounted at the side of the spindle, which forms a hybrid manufacturing system on a single workstation. These are used to restore a damaged part to its original geometry.

FIGURE 1. LASER METAL DEPOSITION PROCESS

The final tool path of a machine for laser aided deposition is in the form of G and M codes, which needs to be verified before part repair is performed [12], since the parts are expensive and a small error[1] in the repair action will cost heavy. Thus visual simulation of the entire repair process is done before any practical application of the codes generated. [2 to 8,14 to 15]

This paper aims to address visual simulation of different manufacturing equipment. The paper will summarize how each machine can be represented in a generic format, derive a parent - child list and a dependency list from it and how this helps in visual simulation of linear and rotational motion. In addition, the paper presents some examples on linear motion simulation using the above format.

2 RELATED WORK

Visual simulation is still very much focused on a single machine or equipment and or is specific to a group of machines. Even though visual simulation of machines is being done since a long time, it has been having a very simple representation of the simulation environment. Also, the solids used in simulation are simple and not sophisticated. A tridimensional box is used to represent each link with
dimensions and positions and the procedure to change the relative position between links in order to correspond to the given trajectory is to transform the tridimensional representation to perspective position to make it look like motion happened. [13]

Following are some other simulation softwares available:

- LINCAGES(Nelson and Erdman 1994), Sphinx (Larochelle et al.1993), SPADES (Larochelle 1998) and Synthetica (Suet al. 2002) : Mechanism design software focus on the dimensional synthesis of planar, spherical or spatial linkages. They do not have interface for constructing mechanisms and simulating their motion.
- WATT by Heron Technologies Inc., SyMech by symech.com and Ch Mechanism Toolkit by SoftIntegration Inc. (Cheng and Trang 2006): These focus on the analysis and synthesis of planar 4-bar, 6-bar or 8-bars. None of them provide a full dynamic simulation for general mechanisms
- Working-Model (Design Simulation Technologies Inc.) and ADAMS(MSC Software Co.): Provide versatile functionalities in modeling and simulating mechanisms. WorkingModel does not support spatial mechanisms, and it does not provide an advanced visualization and interaction functionality. ADAMS does not provide a user-friendly interface.
- MATLAB’s SimMechanics: Provides a powerful multi-body dynamics solver. However, its building process is based on block diagram, which is not intuitive (see what is built), and is overly complicated.
- VRJuggler, Vizard Virtual Reality, Virtual Reality Peripheral Network (VRPN), Virtual Assembly Design Environment (VADE), MIVAS:A Multi-modal Immersive Virtual Assembly System, SHARP: a Dual-Handed Haptic Assembly System, Virtual environment for assembly training : However, relatively less work has been done for applying VR techniques in supporting interactive design especially the conceptual design stage.
- Virtual environment (VE): Their focus is not on the construction of mechanical systems but on the dynamic simulation aspect.
- Stan et al. (2008) have developed a VR: for a particular type of parallel robot by using the MATLAB’s VR toolbox, which does not provide a 3D stereoscopic immersive visualization.
- VR programs dedicated to the design of spherical mechanisms (Furlong et al. 1999) and spatial mechanisms: The focus of these works is on a particular kind of mechanisms, spherical 4R or spatial 4C, etc., not on providing an intuitive and user-friendly interface for interactively building mechanisms.

Currently no mechanism design tool simultaneously has the following features:

1. Focusing on the conceptual design stage
2. Providing an intuitive and user-friendly interface,
3. Supporting arbitrary mechanism topologies and
4. Flexible to support all joint types. [10]

In this paper, a generic method has been developed to represent a machine in the form of numbers and letters in a text document. This could be used for any given machine configuration. Also, a CAD model [16] of the machine can be imported and hence complex structures can be visually simulated.

3 GENERALIZED SIMULATION OF VARIOUS MACHINES

Laser deposition for part repair can be done using other manufacturing equipment as well, some of them having more than five axes. Other than laser processing, there are different kinds of additive manufacturing processes like aerosol jetting, electron beam melting that need to be visualized before practically applying them for fabrication of various parts. Thus, there is a requirement for a generalized visual simulation of the different machines, such that the same algorithm can be applied in case of different equipment. Figure 2 shows an overview of how generalized simulation works.

There are three inputs to the algorithm- one is a 3D CAD model of the machine, second is the tool path document and the other is a configuration text file, the details of which would be discussed later in the paper. The flow of data from the configuration text file goes through a series of transformations. The first transformation converts raw data into more usable data lists. Data lists are relational depictions of this data.

![Figure 2: Flow Chart of the Simulation Process](image)

The raw data is nothing but descriptive information about the machine. From here, the data is transformed into data lists such as parent - child list and dependency list. This model also points out the need to transform raw data into something more usable [9].
Using the 3D CAD model of a equipment, data needed for the configuration file can be obtained as specified. Information from this data can be used to generate a parent - child list and a dependency list, the details of which shall be discussed later in the paper. From the dependency list, the parts to be moved are obtained and also, the type of motion – linear or rotational is noted. These two are inputs to simulate the machine.

The 3D CAD model as shown in figure 4 of a machine is converted into stereo lithography (STL) format. This format describes only its surface model, which is widely used for rapid prototyping and computer aided manufacturing. The STL object is transformed into a faces object as shown in figure 5, which is an array primitive consisting of one sided triangles.

**FIGURE 4. CAD MODEL OF ABB IRB ROBOT**

**FIGURE 5. FACES MODEL OF ABB IRB ROBOT**

4 MACHINE CONFIGURATION DATA

This is a configuration text file that is used in the generic representation of a machine such that all data related to the machine like axes, origin etc are present in it. Machine representation is done in the form of numbers and words. This text document must contain data in columns with each column separated by a tab space. Examples are given for a X and Y table of a CNC machine.

1. Should contain the label for the part(like X or Y or A)
2. Should contain the relation between each part of the machine. The part that is connected to the world space is given -1. The other parts are related by numbers (1, 2,3,4 etc.) where in a part given the relation 1 signifies that it is the parent to the part given the relation 2 and so on. This is used to generate the parent – child list which is explained in detail, later on in the paper.
3. Specify the origin of the part in local co-ordinates(X, Y, Z), that is the co-ordinate frame of the part. The origin is (0, 0, 0) by default for STL files that are imported from the assembly of the machine. For a random file, the origin is the offset distance between the parts with respect to their local part origins.
4. Must contain the orientation of the each part in the form of degrees in order to place them properly in the world space.
5. Must contain the type of motion (translation or rotation) for the particular part.
6. Must contain the limits of motion for each part. There arise 3 cases in this. Either the center or left or right hand side of the whole assembly can be taken as the origin. For these cases, the number line is followed with negative numbers on the left side and positive numbers on the right side.
7. Must contain the vectors of axis of motion in X, Y, Z coordinates for each part. Always 1 is used to depict motion and zero is used to depict that there is no motion. Example: (1, 0, 0) depicts that there is motion along X axis.
8. Must contain the center for rotational motion in local coordinates
9. Contains the axis along which rotation should take place.
10. Must contain the names of the geometry files to be imported with the path that is the folder in which they are present.
11. Must contain the colors in the form of vectors. For example: (1,0,0)-red, (1,1,0)-yellow, (0,0,0)-black,(0,1,0)-green,(1,0,5,0)-orange, (1,1,1)-white,(0,0,1)-blue,(0,1,1)-cyan,(1,0,1)-magenta.

For example, consider two tables X and Y, one over the other and a configuration file for the same would be as follows.

**FIGURE 6. ILLUSTRATION OF X AND Y TABLE**

The text document for the figure 6 would contain the data as shown in table 1.

The parts are labeled as Y and X respectively. The Y table is connected to the world and therefore it is designated by -1. The X table is connected to the Y table and hence is designated by 0. The coordinates of the origin for both the parts is (0, 0, 0) as the parts are imported from the assembly of the machine. The
orientation of each part in the form of degrees is zero as well. Both the parts have linear motion. The vector of axis of motion for X table is (1, 0, 0) and for Y table it is (0, 1, 0). Since there is no rotation, the rotational part of the text document is not shown. The names of the .STL files to be imported are given as X.STL and Y.STL. Colors are not represented as the default color is used.

### TABLE 1. EXAMPLE OF MACHINE CONFIGURATION

<table>
<thead>
<tr>
<th>Label</th>
<th>Relation</th>
<th>Origin</th>
<th>Orientation</th>
<th>Motion</th>
<th>Axis of motion</th>
<th>File name</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1, 0, 0</td>
<td>Y.STL</td>
</tr>
<tr>
<td>Y</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0, 1, 0</td>
<td>X.STL</td>
</tr>
</tbody>
</table>

### 5 UTILITY DATA STRUCTURES

There are two kinds of data lists that are derived from the configuration text file. One is the parent - child list and the other is a dependency list.

**Parent-Child list:**

In order to simulate a machine, there is a need to know the parts that move in tandem with each other and the parts that move independent of each other, such that they can all be put in a moving and a non moving frame respectively while performing simulation. This is nothing but the dependency list. In order to derive the dependency list, the parent – child relationship for each part is to be known. This gives the dependency information required.

Considering the case of X and Y table from figure 6, the X table is connected to the world space and Y table is connected to X. Therefore Y is a parent to X. Thus the parent - child list in this case would be:

\[[2, 1]\]

But, there are bound to be complex cases. For example, in a CNC machine as shown in figure 7, the column for relation between each part would contain the following data:

\[-1, 0, 9, 10, 8, 1, 9, 3, 0, 7, 4.\]

### FIGURE 7. FACES MODEL OF THE 5 AXIS CNC MACHINE

This is depicted in figure 8.

### FIGURE 8. Illustration of the parent – child and dependency list

Then the parent- child list would be:

\[[[1, 5], [5], [1, 7], [10], [1], [9], [9], [4], [2, 6], [3]]\]

The empty arrays show that their corresponding parents do not have any more children.

**Dependency list**

For figure 6, the dependency list would be: [2, 1]. The motion of the X table is independent of any other part. That is when Y table moves, X table does not need to move. The Y table, however has a motion that depends upon X, that is, when X table moves, Y moves as well. Hence the dependency list generated shows that Y is dependent on X in a numerical format of [2, 1].

For figure 8, the dependency list would be:

\[[[1, 5] [8, 4, 10, 3, 7, 9, 2, 6]]\]

### 6 TOOL PATH DATA

This is another text document that contains the tool path motion in axial coordinates for n number of axes and the time step for the same in the form of distance points. This text document must contain data in columns with each column separated by tabs.
For a linear 3 axes motion:

- The first column contains the motion of the machine in X direction.
- The second and third columns contain the motion of the machine in Y and Z directions respectively. The Z direction is just the addition of layers over the same path, thus leading to additive manufacturing.
- The fourth column consists of the time step.

This data represents the path for laser deposition. One of the examples is the path for LAMP logo as shown in figure 9.

![LAMP Logo](image)

**FIGURE 9. SIMULATION OF THE LAMP LOGO DEPOSITION**

7 VISUALISATION

The input for this is the dependency list and type of motion. The dependency list helps in giving data regarding parts that move along with a particular one.

The type of motion can be either linear or rotational. Linear motion is a straight line motion that can be described using only one spatial dimension. Rotational motion for each joint differs. There are three types of rotational motions:

- **Revolute** – 1 DOF
- **Cylindrical** – 2 DOF
- **Spherical** – 3 DOF

When the same joint has more than one DOF, then it is represented as many times as the number of degrees of freedom in the configuration text file except that its geometry is represented only once and the rest of the representations are considered virtual.

The faces primitive is represented in one of the columns in the configuration file. The tool path text document gives information regarding axial movements. All the above information is used to generate visual simulation of the equipment by placing the parts to be moved at one time step, in tandem in a moving frame and the rest of the parts in a non-moving frame and updating the moving and non-moving frame for each time step.

8 RESULTS

The simulation results for linear motion are shown in figure 10 and figure 11.

**FIGURE 10. ILLUSTRATION OF LINEAR MOTION SIMULATION ALONG X AXIS**

**FIGURE 11. ILLUSTRATION OF LINEAR MOTION SIMULATION ALONG Y AXIS**

9 CONCLUSIONS AND FUTURE WORK
3D visual simulation software has been developed, that is capable of simulating a tool-path of any machine. The software provides 3D graphical environment to manipulate (translate, rotate, zoom) the tool path. The simulator evaluates a correct position of the tool in space, i.e. the location and the orientation of the tool for an arbitrary sculptured surface. Currently, the simulator provides preliminary results that can be used to estimate the errors graphically.[11] Future work includes

- Generic representation of parallel mechanisms is complex as the end effector may have n degrees of freedom and is connected to the base by n independent kinematic chains. For example, a hexapod.
- Rotational motion at joints.
- Collision detection as shown in figure 12, detecting the intersection of two or more objects in space.

![Image of collision detection](image)

**FIGURE 12. ILLUSTRATION OF COLLISION OF TOOL INTO THE PRODUCT**

10 ACKNOWLEDGEMENT

This research was supported by the National Science Foundation grants IIP-0822739 and IIP-1046492. The support from Boeing Phantom Works, Product Innovation and Engineering, LLC, Missouri S&T Intelligent Systems Center, and the Missouri S&T Manufacturing Engineering Program, is also greatly appreciated.

REFERENCES


