Direct Generative Vehicle Machining

Zhenyu Cheng[1], C. Greg Jensen [2] and W. Ed Red [3]

[1]Brigham Young University, chengzy@email.byu.edu
[2] Brigham Young University, cjensen@byu.edu
[3] Brigham Young University, ered@et.byu.edu

ABSTRACT

Direct vehicle generative machining is a new method to machine the vehicle globally and directly. In most CAM systems the surface drive method or area mill method is used to generate the tool paths. No matter what method is used, the user must select the surface patches one-by-one to generate the tool path sequence. Usually, a vehicle model has more than 1000 surfaces; hence, generating tool paths for the vehicle is extremely time-consuming. Using the direct generative vehicle machining method described in this paper, the processing time of manufacturing plan can be greatly reduced. The details of the direct generative vehicle machining methods are described as applied to an SSR model. Using a global vehicle machining method, the SSR model represented by nearly 2000 surfaces was reduced to a model with only 37 surfaces.

Keywords: Generative machining, Direct machining

1. INTRODUCTION

The creation of a concept vehicle by machining a clay or foam prototype can be very time consuming. Once the designer has completed the vehicle's surface definition, it can take days, even weeks, before the clay model is machined to visual satisfaction by a multi-axis machine tool. Investigating why it takes so long to develop the tool paths necessary to cut the car body, one quickly discovers that organizational structures and resistance to change limit the introduction of disruptive new technologies. In many automotive design and styling studios the task of tool path generation is relegated to union employees bent on maintaining status quo, which unfortunately for US manufacturers is not a limitation of their competition. Second, the tool path planner is content to use old inefficient planning strategies that machine the surface patch by patch, as opposed to sweeping the tool across large sections of the car. Third, when design iterations of the clay model are required the path planner usually starts from scratch, rather than modifies (morph) the original surface geometry and regenerate previously planned paths.

What design studios need is a process that is more like printing and less like the union laden production milling shop that function autonomously to produce the physical car components. How much design cycle-time would be saved or how many design iterations would occur if design personnel could send their model to a mill in much the same way as they send them to a high photo quality color plotter?

This paper identifies the five technologies that designers and digital sculptors could use to directly generate vehicle prototypes. First, the designer would want the path planning interface to be as straightforward and easy to use as the printing dialog. Second, there must be generative path planning algorithms that support this simple interface that can apply best-in-class path planning strategies to aggressively machine the surfaces. Third, when the path planning is complete, if the designer so desires, he/she can call for a machine preview which will simulate the machining of the vehicle. Fourth, the file must be sent to a Direct Machining and Control (DMAC) buffer or queue to await processing. The final technology is the reuse of prior tool paths when slight refinement of the model occurs. The main focus of this paper is the generative path planning phase.

Over the last two decades there has been a significant transition from the days of hand sculpting of concept cars to today when having the smallest detail represented in math and cut on a computer controlled mill is commonplace. Yet, even with all of this automation we still struggle with reducing the time it take to produce the first full-size vehicle prototype because of all the hand-offs and unnecessary department delays that are nothing more than a throw back to the days before CAD/CAM automation.

No one would argue that computer controlled machining plays an important role in what can be designed and manufactured by automotive OEMs. Also, no one would argue that in the concept phase full-size clay models are still required for verification of design intent. It is equally important to note that car definitions originate either in the form of virtual surfaces within a CAD system or as scanned point cloud data. Often when models originate in CAD and then machined, the resulting clay model is frequently worked by hand to correct surface definitions and add features and details. This recourse is taken due to delays (days to weeks) because of the time required to correct the CAD model and regenerate tool paths. Today, scanned changes are taken back into the CAD model through reverse engineering methods to once again establish the CAD master model.

All machining researchers would agree that automotive exteriors have brought about increased challenges and demands. More vehicle exteriors are exhibiting the characteristics of an organic form like a sculpture. Complex transitions from convex to concave areas and rapid change of curvatures are commonplace. This requires a new adaptive strategy for rapidly modifying shapes.

2. BACKGROUND

In this section we review pertinent research and provide appropriate definitions and derivations to allow the reader to understand and extend our work.

2.1 Generative Machining

In Computer-integrated manufacturing (CIM), computer-aided process planning (CAPP) is the function within a manufacturing facility that establishes the processes, process parameters, and machines used to make a part or shape. Two forms of automated process planning are in use – retrieval or variant type, and generative type.

In generative type CAPP, various process information such as knowledge of component geometry, component material, specifications of the machine tools, cutting tools and work holding devices, operation sequencing, and production costs are all synthesized to create a process plan. Some generative type CAPP systems are PMPS, GAPP, and XPLANE [1-4]. The limitations of these CAPPs include (i) selective geometry of the part, (ii) lack of important functional modules of planning systems, such as tool selection, and (iii) can't be machined directly. A methodology to support planning and machining capability of feature-based design and automatic tool selection was proposed [5], but no record could be found of its implementation.

2.2 Machining Algorithms

Modern sculptured surface designs provide strong arguments for the benefits of 5-axis milling. The two additional degrees of freedom in a 5-axis CNC machining has brought new advantages and added flexibility to the process of machine sculpting. With new opportunities come new challenges, in that conventional CAD/CAM systems still lag in the acceptance and implementation of proven 5-axis curvature catering or matching algorithms. Since 1989 many researchers have focused their efforts on improved flat and filleted end mill machining. The obvious benefit of all this research has been the calculation of much larger stopovers, thus reducing the overall machining times. Vickers and Quan (1989) reported that using flat-end mills has twenty times the productivity of sphere-end mill cutting [6-9].

A variety of algorithms for five-axis machining have been developed and implemented within some commercial CAD/CAM software packages. Curvature matching techniques are used in some industries for machining open complex sculptured surfaces with fewer passes for a specified tolerance [10].

2.3 Direct Machining and Control (DMAC)

A major limitation of the current CNC manufacturing methods is the form of data transfer between the driving CAD model and the machine tool controller—M&G code. There is no association between the tool path data and the master CAD model. This means that any modification to the original CAD model requires a regeneration of all subsequent intermediate files as well as the M&G codes. This traditional design-to-manufacturing process increases the time and costs of creating a physical model from a conceptual design.

Over the past six years a new method called Direct Machining and Control (DMAC) has been developed at Brigham Young University. DMAC allows the machine controller to directly interpret CAD/CAM process instructions, based on the original CAD part model, without need for post-processing into M&G code and other intermediate forms. Instead, DMAC uses the original CAD mathematics to drive the machine. Because the mathematical representation of the CAD model is universal, the DMAC controller can be configured to communicate directly with any CAD/CAM system, given the right interface, and is not dependent on any machine-specific M&G code [11-12].

Direct machining eliminates many of the modern problems associated with the current design-to-manufacture processes. There are many advantages of using the true geometry instead of representative ASCII tool paths. Because machining is based on the original design geometry, there is no geometric information loss [13].

3. METHODS

Direct generative vehicle machining is an approach that machines the vehicle globally and directly. In most CAM systems either the surface drive method or the area mill method is used to generate the tool path. No matter what method is used, the user must choose the surfaces one-by-one to develop the machining paths. Vehicle models commonly have in excess of 1000 surfaces; hence, generating tool paths for the whole vehicle is really time-consuming. Using the direct generative vehicle machining method, the planning process is automated and the time is reduced.

The global vehicle machining method can be described by the following steps:

1) Apply generative path planning to adjacent smooth surfaces
   a) Select a surface, and choose another surface adjacent to it.
   b) Judge whether the two surfaces are smooth or not. If they are smooth, sew the surfaces together defining a new surface.
   c) Select the next surface that is adjacent to the new surface; if smooth, sew a new surface; repeat until all adjacent surfaces have been checked.
   d) Loop back to step 1 selecting two new surfaces; if none exist, only single non-smooth patches remain.
   e) Generate tool paths on smooth surfaces and patch mill the non smooth.
2) Use virtual tool path simulation to check the overall path continuity and quality.
3) Send the tool paths to the direct machine controller.

3.1 Generative Path Planning

3.1.1 Selection of Adjacent Surfaces

The bounding box technique was used to select adjacent surfaces. The surface information and the boundary points can be derived from CAD systems and sent to a database. The sum of the boundary points forms a point cloud. Hence after one surface is selected, the point cloud data is searched. If another surface has the same boundary, then the two surfaces are adjacent. We compare the two surfaces to judge whether they are smooth or not.

3.1.2 Surface Smoothness

A tensor product B-spline surface patch is defined

$$b(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} b_{i,j} B_i^m(u) B_j^n(v) \tag{1}$$

where $b_{i,j}$ are Bezier points, and

$$B_i^m(u) = \binom{m}{i} u^i (1-u)^{m-i} \tag{2}$$

Let x(u, v) and y(u, v) be two patches. They are r times continuously differentiable across their common boundary curve x(u_I, v) = y(u_I, v) if all u-partials up to order r agree there:

$$\frac{\partial^r}{\partial u^r} x(u,v)\,|_{u=u_I} = \frac{\partial^r}{\partial u^r} y(u,v)\,|_{u=u_I} \tag{3}$$

That is,

$$\left(\frac{1}{\Delta_{I-1}}\right)^r \sum_{j=0}^{n} \Delta^{r,0} b_{m-r,j} B_j^n(v) = \left(\frac{1}{\Delta_I}\right)^r \sum_{j=0}^{n} \Delta^{r,0} b_{m,j} B_j^n(v) \tag{4}$$

where $\Delta_I = u_{I+1} - u_I$. Since the $B_j^n(v)$ are linearly independent, we can compare coefficients:

$$(\frac{1}{\Delta_{I-1}})^r \sum_{j=0}^{n} \Delta^{r,0} b_{m-r,j} = (\frac{1}{\Delta_I})^r \sum_{j=0}^{n} \Delta^{r,0} b_{m,j} \quad (5)$$

This is the C^r (r=0, 1, 2 …) condition for Bezier curves, applied to all n+1 rows of the composite Bezier net. We thus have the C^r condition for composite Bezier surfaces. Two adjacent patches are C^r across their common boundary if and only if all rows of their control net vertices can be interpreted as polygons of C^r piecewise Bezier curves.

We define surface smoothness of two adjacent patches as G^2 geometric continuity. Hence from the surface smoothness condition, all rows of their adjacent Bezier curves should have G^2 geometric continuity. The algorithm for checking smoothness is given in the following pseudo code fragment.

```
Get surfaces S, T
If they are adjacent
{
      Get the their common edge Ls
      Create a plane V perpendicular to Ls at C
      Get two surface curves Ps, Qs formed by V intersection with S and T surfaces
      Fit Ps, Qs to Bezier curves
      Compute the curvature k1 of Ps at the common point with Qs
      Compute the curvature k2 of Qs at the common point with Ps

      If k1=k2
      {
            surface smooth = true
      }
}
```

Let S and T denote two adjacent surfaces. To judge whether two surfaces are smooth, first the common line Ls of the two adjacent surfaces is extracted. Then a plane is created through C, the meddle point of Ls and normal to Ls. This plane intersects the S and T surfaces, creating two surface curves Ps, Qs as shown in Fig. 1.

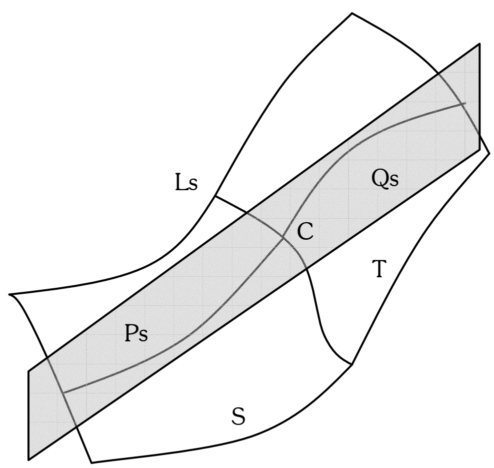


Fig. 1. Use plane V to cut the surface

The spline curves for Ps and Qs can be derived from the CAD geometry. The spline curve can be expressed by Lagrange polynomials in the explicit form:

$$P(t) = \sum_{i=0}^{n} p_i L_i^n(t) \quad (6)$$

where p_i are the spline points, and $L_i^n(t)$ are Lagrange polynomials,

$$L_i^n(t) = \frac{\prod_{\substack{j=0 \\ j\neq i}}^{n}(t - t_j)}{\prod_{\substack{j=0 \\ j\neq i}}^{n}(t_i - t_j)} \tag{7}$$

G^2 geometric continuity is used to judge adjacent surface smoothness, meaning tangency and curvature continuity as shown in Fig. 2.

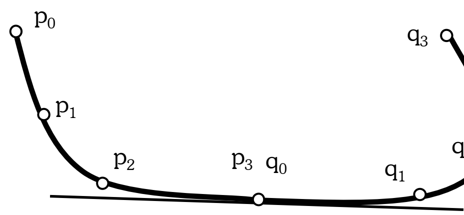


Fig. 2. Geometric continuity

Because Ps and Qs are in the same plane, they can be treated as 2-D spline curves. The curvature of a spline curve can be computed as

$$\kappa(t) = \frac{\ddot{x}(t)\,\dot{y}(t) - \ddot{y}(t)\,\dot{x}(t)}{[(\dot{x}(t))^2 + (\dot{y}(t))^2]^{3/2}} \tag{8}$$

x(t), y(t) are the values of P(t) in X and Y directions. Hence if Ps and Qs have the same tangent line and curvature, surfaces S and T are smooth, and have G^2 geometric continuity.

3.1.3 Sew Surfaces

If two surfaces are geometrically continuous, they can be sewed together. Fig. 3 is the decision flow chart. The sew option lets you join together two or more surfaces, thus creating a single surface.

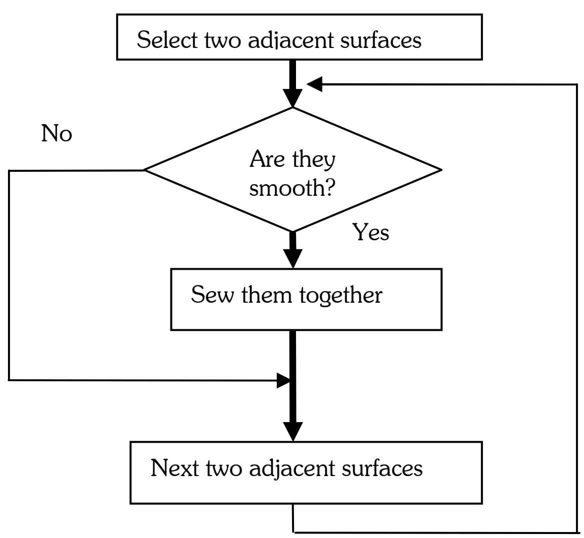


Fig. 3. Flow chart for sewing surfaces

3.1.4 Selection of Next Adjacent Surface

Once two surfaces are sewed they become a new surface with an expanded boundary. This process is repeated until there are no adjacent surfaces. The process is finished when there are no patches left in the database. All resulting surfaces are smooth across their interior regions. The next step is to generate the tool path across the stitched surface.

3.1.5 Generate Tool Path

Tool paths are created by the manufacturing module of CAD/CAM systems. This method offers many advantages because the manufacturing modules have interfaces to set up all necessary parameters for the tool path geometry. , Even gouge detection and tool path optimization routines are embedded into the system, and all the manufacturing information can be accessed and passed to the CNC machine.

In this paper, Unigraphics (UG) NX 2.0 was used to create the tool path. The surface area drive method was used to define the 5-axis tool path. This drive method was selected because of the complexity of the part surfaces and the required control of the tool axis. Tool paths are created by projecting drive points to part geometry. Drive points are generated from drive geometry, such as curves, boundaries, faces, or surfaces, and projected along a specified projection vector to the part geometry. The tool then positions relative to the part geometry to generate the tool path. The tool path is created using the output cutter location point at the tip of the tool. Both the projection vector and the tool axis are variable and are defined as normal to the drive surface. Fig. 4 shows the surface area drive method [14].

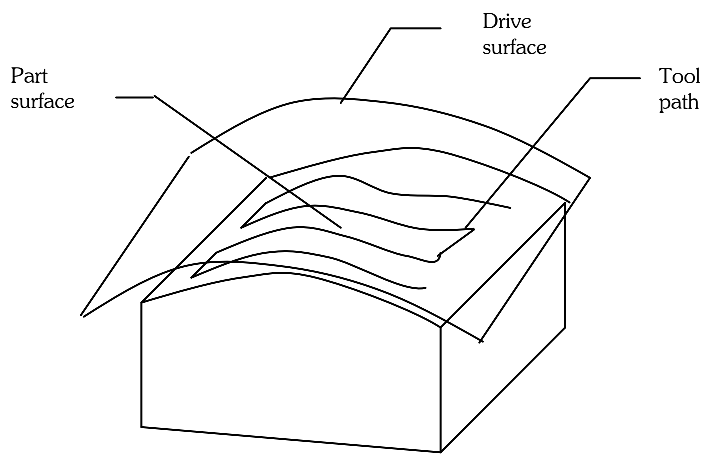


Fig. 4. Surface area drive method

3.2 Simulation of Planned Path

The DMAC controller can be placed in simulation mode as shown in Fig. 5. The motion set points derived from the motion planner module can be sent back to the move engine, and then back to the simulation module. Hence, real time simulation is realized.

3.3 Send tool path to DMAC Controller

The DMAC architecture is configured on a dual-processor platform. Fig. 6 shows the current DMAC architecture [15]. The top layer of the DMAC system is a non real-time Windows CAD/CAM application that runs on one processor. All the real-time control applications, such as motion planning and servo-control loops, run on the second processor, with feedback between the motors and I/O occurring over a high speed network. The tool paths are generated in CAD/CAM applications. Once tool path data have been created, they are passed directly to the motion planner as machine moves. The servo controller receives the torque set points from the Motion Planner and performs the servo control functions for each actuator in the machine.

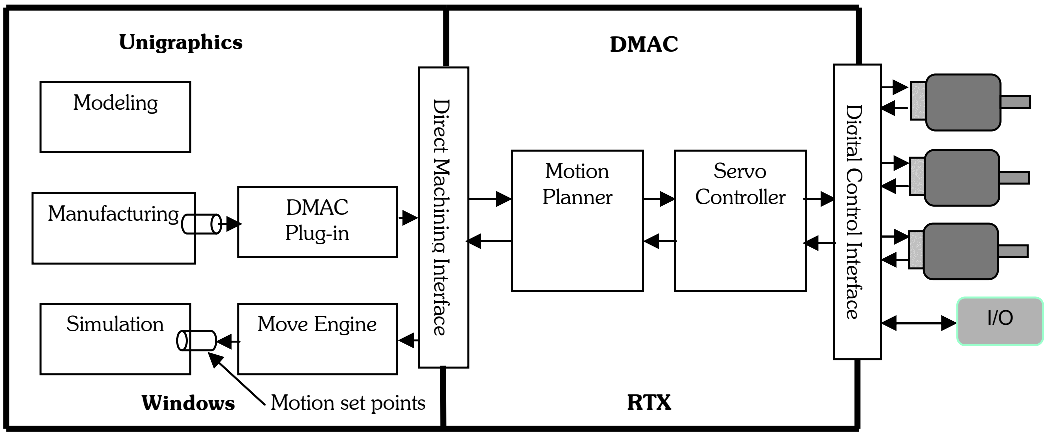


Fig. 5. Simulation Architecture

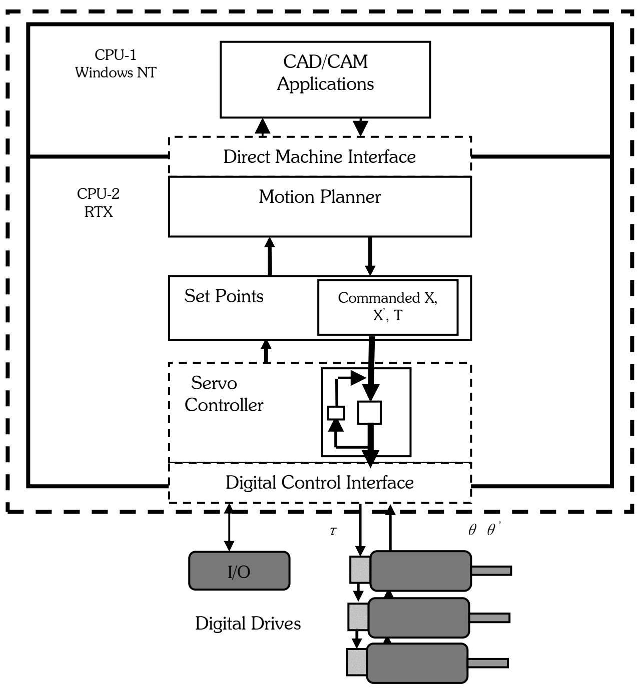


Fig. 6. DMAC Overall Architecture

Under this new paradigm, different machine tools are controlled in a similar way printers are controlled by a personal computer. All machine tools are directly connected to CAD/CAM applications through a software interface that essentially acts as a driver. The tool paths and process plans generated from the CAD/CAM applications can be sent directly through this software interface. This software driver can then invoke and control the machine tool that is connected through this driver.

Direct generative vehicle machining is possible with the DMAC controller. When the tool paths for the vehicle are ready, they can be sent to the DMAC controller directly without any postprocessor or M&G code. The DMAC controller will control the machine tools and finish the machining.

4. SSR TEST CASE AND RESULTS

In this paper, a GM SSR car model was used as a test case. SSR is a representational model of an automotive exterior; hence, it was used to test global vehicle machining methods. Fig. 7 is the SSR wire frame model.

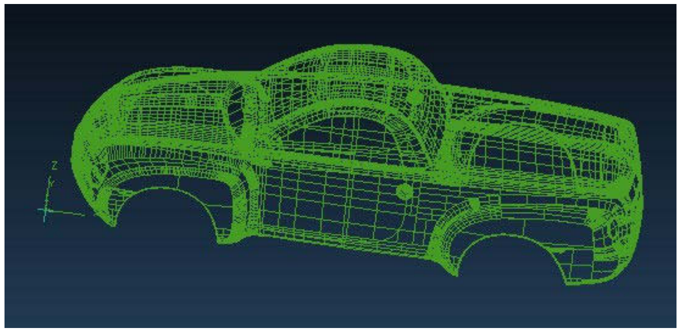


Fig.7. SSR wire-frame model

There are nearly 2000 surfaces in the SSR model. Using global vehicle machining, the surfaces can be grouped into larger areas while still maintaining an acceptable tolerance for machining the prototype in clay. Using the methods described in this paper the SSR model was reduced to 37 areas. Fig. 8 shows the roof areas of the SSR. There were 223 small surfaces in the roof area. Using global machining method, now there are only 3 surfaces: Windows, Windows Frame, and Ceiling.

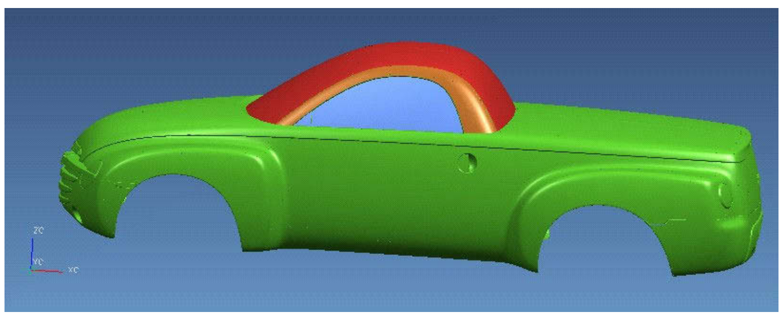
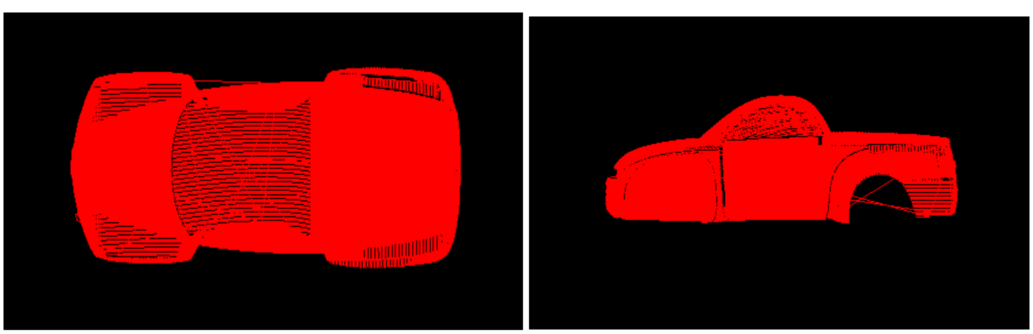


Fig. 8. Top area of SSR

Similar methods were applied to the full vehicle leading to the generation of tool paths over the entire surface as shown in Fig. 9.



a) b)

Fig. 9. a) Top view of the full vehicle tool paths b) Side view of the full vehicle tool paths

5. CONCLUSIONS

Direct generative vehicle machining is implemented in this paper. The SSR model was used as the test case. By this method, the surfaces in the vehicle and the processing time of manufacturing plan were greatly reduced, and tool path was sent directly to machine tool without any M&G code.

In the future, curvature matched machining method will be used to generate the tool path, and the vehicle will be machined on a Tarus 5-axis mill to test the tool path, and compared with traditional machining.

6. REFERENCES

[1] Utpal Bose, "A Cooperative Problem Solving Framework for Computer-Aided Process Planning", Proceedings of the 32nd Hawaii International Conference on System Sciences – 1999

[2] Levin J.B., Dutta D., "PMPS: A Prototype CAPP System for Parallel Machining," *Journal of Manufacturing Science and Engineering*, 118, pp. 406-414, 1996.

[3] Laperriere L., ElMaraghy H.A., "GAPP: A Generative Assembly Process Planner," *Journal of Manufacturing Systems*, 15(4), pp. 282-293, 1996.

[4] Van Erve A.H., Kals H.J.J., "XPLANE: A generative computer aided process planning system for part manufacturing", *CIRP Annals*, 35, 1986.

[5] Lee Y.S., Daftari D., "Feature-composition Approach to Planning and Machining of Generic Virtual Pockets," *Computers in Industry*, 31, pp. 99-128, 1996.

[6] Choi, BK, Park, JW, and Jun, CS, "Cutter-Location Data Optimization in 5-axis Surface Machining", *Computer Aided Design*, Vol.25, No.6, pp.377-386, June 1993

[7] Elber, Gershon, and Cohen, Elaine, "Tool Path Generation for Freeform Surface Models", Proc 2 Symp Solid Model Appl, ACM, New York, NY, pp. 419-428, 1993

[8] Jensen, C.G., and Anderson, D.C., "A Review of Numerically Controlled Methods for Finish-Sculptured-Surface Machining", *IIE Transactions*, Vol.28, pp. 30-39, 1996

[9] Yang, D.C.H. and Kong, T., "Parametric Interpolator Versus Linear Interpolator for Precision CNC machining", *Computer Aided Design*, Vol.26, pp. 225-234, 1994

[10] Zhenyu Cheng, Wei Li, Robert Cheatham, Jianguo Wang, C.G. Jensen, W.E. Red, "Developing A Direct Machining System For Commercial CAD/CAM Packages", Proceedings of 2004 ASME International Mechanical Engineering Congress, Anaheim, California, USA, Nov 13-19, 2004

[11] Zhenyu Cheng, Robert Cheatham., Jianguo Wang, C.G. Jensen, Yifan Chen, Brian Bowman, "A Machining Benchmark Comparison of CAD 5-axis machining packages", Proceedings of ASME-International Design Engineering Technical Conferences & Computers And Information In Engineering Conference, Salt Lake City, USA, September 28 – October 2, 2004

[12] Bassett CP, Jensen CG, Red WE, Evans MS, "Direct Machining: a New Paradigm for Machining Data Transfer", Proceedings of DETC2000/DFM-14298: ASME 5th Design for Manufacturing Conference, Baltimore, Maryland, September 10-13, 2000.

[13] Bassett CP, Jensen CG, Bosley JE, Luo Y, Red WE, Evans MS, Direct Machining Architectures Using CADCAM Generative Methods", Proceedings of the IASTED International Conference on Control and Applications, 2000.

[14] Unigraphics NX2.0 Documentation, EDS, 2004.

[15] Wei Li, W.E. Red, C.G. Jensen, Tyler Davis, "Dynamic Reconfigurable Machine Tool Controller", Proceedings of 2004 ASME International Mechanical Engineering Congress, Anaheim, California, USA, Nov 13-19, 2004