Development of a Variational Part Model Using In-Process Dimensional Measurement Error

Shane A. Carlson[1], W. Edward Red[2] and C. Gregory Jensen[3]

[1]Brigham Young University, shane.carlson@ssp.navy.mil
[2]Brigham Young University, ered@byu.edu
[3]Brigham Young University, cjensen@byu.edu

ABSTRACT

To improve the precision of CNC machined parts, the dynamic machining errors due to tool deflection, tool wear, heat deformation, etc. should be accounted for. This paper presents a systematic approach for automatically compensating for these errors based on a new closed-looped machining scheme. The new scheme introduces a Variational Part Model which inherently possesses the dynamic machining error. To create the Variational Part Model, the machined part is measured on the machine (in-process inspection) and compared to the CAD model's theoretical data. The data is then used in conjunction with modeling functions contained in Unigraphics' Application Programming Interface (API) to interact with the CAD model and modify it's surface geometry. The validity and effectiveness of the methods are presented as well as results from experimental testing.

Keywords: in-process inspection, closed-loop machining, dynamic error, CAD/CAM.

1. INTRODUCTION

With the increasing precision of CNC machining centers and CMM technology, in-process part inspection and closed-loop machining are being realized by using the process machine as an inspection machine. Closed-loop machining has been a critical research topic over the past 15 years. Closed-loop machining is basically a system for manufacturing parts with automatic compensation of machining errors such as worn tools, thermal deformation and other process variations. A gage or CMM touch probe captures a part's dimensional data, deviations outside pre-set limits are fed back to the CNC controller and the tool or axis is offset to make up for the deviation. Fig. 1 shows the steps involved with the closed-loop machining scheme: 1) to take a part modeled in a CAD/CAM system, 2) generate tool paths, 3) machine the part, 4) perform inspection, and 5) accept, reject or re-machine the part by feeding back error offsets to the CNC controller. [1]

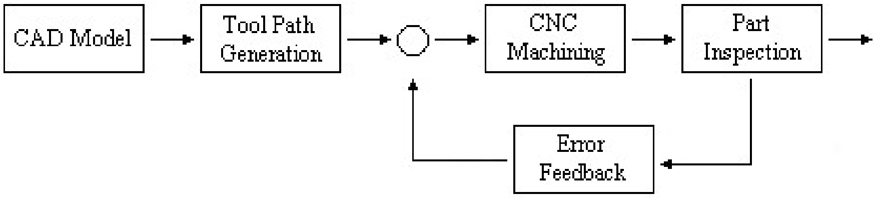

Fig. 1. Closed-loop machining scheme.

For quite some time now, the closed-loop machining scheme has incorporated the concept of in-process machining and inspection while the part is fixtured on the machine tool. The biggest advantage of closed-loop machining is the ability to keep track of process variations and make corrections to the manufacturing process when needed. The ultimate goal is to have 100% machined parts accepted after inspection. The difficulty is that machines are not 100% accurate, thus process control is extremely important. When in-process machining and inspection are included, closed-loop machining practices allow adjustments to be made dynamically as parts are being machined, thus improving part precision and reducing the amount of scrap.

The accuracy of the finished part is almost entirely dependent on the tool paths generated by the CAM system. The goal of the CAM system is to provide a sufficient number of cutter location data points that will produce the part with an acceptable amount of error. Cutter location points are provided in a discrete manner and errors are unavoidable due to the point-to-point operations of the CNC machine. Some of the errors likely to emerge are interference with the fixture, gouging and under cutting. To compensate for these errors external computer simulation software packages have been developed to check for machining errors developed by other CAD/CAM systems.

All machine tools inherently possess some type of error. Errors in roll, pitch and yaw develop over time and need to be monitored frequently to ensure the quality of manufactured parts. Analytic compensation procedures such as volumetric error modeling [2] have been developed to compensate for these errors. Measurements of each error are accomplished by using a laser interferometer, electronic level or ball bar system. This approach, however, is limited because the accuracy is obtained under a load free condition. [3]

Machining process disturbances occurring during the machining process include: tool deflection, thermal deformation, controller following errors, mechanism compliance, etc. Many researchers have attempted methods to compensate for such errors. They have also tried to incorporate analytical modeling techniques to model cutting forces, chatter, vibration, heat deformation and tool deflection which occurs during machining [4-9].

One of the more advanced methods developed recently is a geometrical error feedback scheme. In the proposed scheme, the machined part is: 1) measured on the machine, 2) compared with the CAD model, and 3) the tool path is corrected repeatedly until the predefined geometric accuracy is obtained. The method described is placed in an iterative part programming procedure, repeatedly making corrections after each finish machining operation. The critical steps (2 and 3) involve taking measured data points and adjusting the NC tool paths stored in the nominal CL data file to reflect the machining error. In addition to the method described above, the error feedback scheme includes compensation for the machine's volumetric error using a laser interferometer and ball bar system. [10]

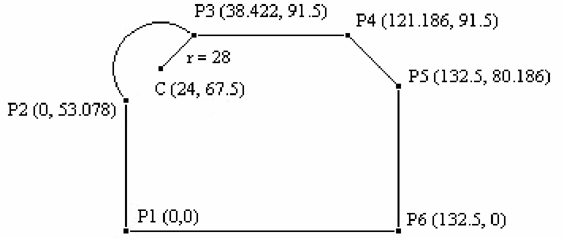

Fig. 2. Test part used for geometrical error feedback validation.

Testing of the geometrical error feedback routine was conducted using an aluminum part with the configuration shown in Fig. 2. The test included the following cutting conditions: HSS end mill, cutting width of 2.5 mm, cutting depth of 4 mm, feed rate of 60 mm/min, and a spindle speed of 900 rpm. Various points along the test part were measured as shown in Tab. 1. The average error (distance between the theoretical and measured values) without compensation measured 0.2177 mm. After compensation (first iteration), the average error measured 0.0443 mm.

	Theoretical values		Measured values without feedback			Measured values with feedback		
	x (mm)	y (mm)	x (mm)	y (mm)	Error (mm)	x (mm)	y (mm)	Error (mm)
P1	0	0	-0.147	-0.117	0.188	-0.027	-0.039	0.047
P2	0	53.078	-0.226	52.827	0.338	-0.046	52.994	0.096
C	24	67.5	24.279	67.634	0.31	24.01	67.503	0.01
r	28		28.163		0.163	28.053		0.053
P3	38.422	91.5	38.599	91.629	0.219	38.454	91.52	0.038
P4	121.186	91.5	121.306	91.639	0.184	121.209	91.524	0.033
P5	132.5	80.186	132.633	80.302	0.176	132.533	80.212	0.042
P6	132.5	0	132.618	-0.114	0.164	132.524	-0.025	0.035

Tab. 1. Error comparison at various manufacturing points.

The geometrical error feedback routine showed a significant improvement (79.7% reduction in error) in the precision of the machined part after the first iteration. The results indicate that the approach can be used as a means for correcting both static and dynamic machining process disturbances without having to count on the analytical modeling techniques. In addition, the proposed scheme is meaningful in that it can be implemented as a feedback mechanism for geometric control. Although this method compensates for the static and dynamic error, the error adjustments were applied to the NC tool paths instead of the feature model. The problem with this method is the adjusted tool paths are no longer associated with the CAD model's geometry.

The research in Direct Machining And Control (DMAC) at Brigham Young University recently developed methods to automatically compensate for dynamic machining error and to associate the errors directly on the CAD model [11]. The CAD model is consequently called a Variational Part Model (VPM) because it inherently possesses the machines dynamic error. This paper discusses the techniques developed to automatically alter part geometry and discusses the results obtained through testing using the inspection software package PC-DMIS [12], a Sugino 3-axis machine tool and a Renishaw touch trigger probe.

2. NEW CLOSED-LOOP MACHINING SCHEME

To demonstrate the ability to automatically correct dynamic machining errors through closed-loop feedback, techniques were developed and implemented in a Unigraphics plug-in (computer program run from inside Unigraphics). The techniques developed take measurement information from PC-DMIS's inspection report and use them to create the VPM. With the addition of these techniques, a new closed-loop machining scheme was developed. The following in-process machining, inspection and error feedback routine shown in Fig. 3 shows where the VPM fits into the closed-loop machining scheme. Prior to performing the final finishing passes, the part is inspected and its geometric dimensions and tolerances are evaluated. In so doing, when the CMM software reports dimensions/tolerances outside pre-defined limits, modifications are automatically made to the VPM and the process plan updated. When the part is re-machined, a more accurate part is made, thus eliminating scrap.

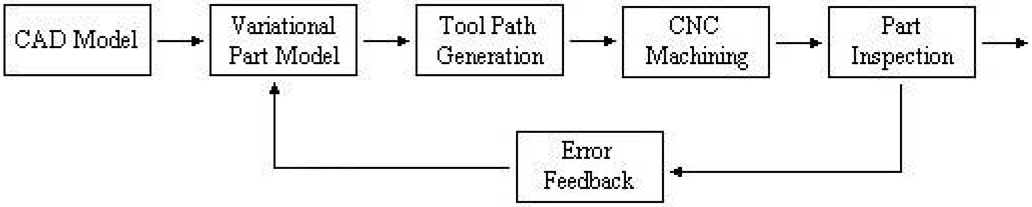

Fig. 3. New Closed-loop Machining Scheme.

3. VARIATIONAL PART MODEL METHODOLOGY

The goal of this research was to develop the methods and techniques to modify part geometry by using measurement data. The modifications are made possible by using Unigraphics' Application Programming Interface (API) [13]. The API contains the functionality available to users to write customizable software programs and to interact with the CAD/CAM model. Examples of modeling functions include: creating points, lines, arcs, cylinders, blocks etc. Even the more advanced operations are available such as creating surfaces with the through curves function. This section will discuss the methodology used to create the VPM and describe each of the modeling functions used in this research. Fig. 4 illustrates the VPM methodology.

The general idea is to first take the part's measured data and compare it to the part's theoretical data. This is accomplished by measuring the part using PC-DMIS. The second task is to take the data and create points that represent the new VPM surface. Third, a series of splines are created through the VPM points. The splines are then used in conjunction with a through-curves feature operation. This operation creates a sheet body through the collection of splines and has zero thickness. The final task is to replace the existing surface with the new, through-curves sheet body.

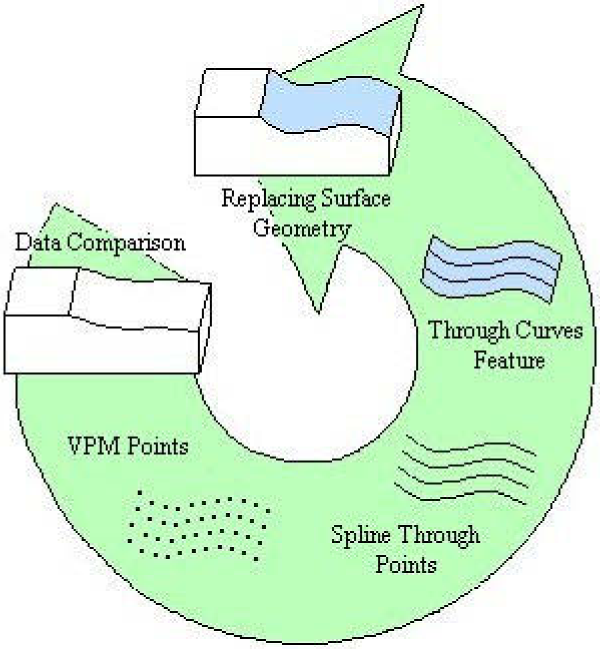

Fig. 4. Diagram illustrating the VPM methodology.

One of the key issues regarding the development of the VPM is the ability to automatically identify part features such as lines, arcs, curves, faces and other CAD features. In order to use API modeling functions the user has to know the surface's object identifier. PC-DMIS has a feature called the DirectCAD Interface and through this interface, PC-DMIS has the ability to obtain object identifiers. Because PC-DMIS currently has not implemented this functionality, an interactive, Graphical User Interface (GUI) built in Unigraphics was created to provide the needed functionality. This interface enables the user to select the desired feature and obtain the surface's object identifier. This research uses the surface's handle (consistent object identifier) because its identity never changes. The user copies the handle and pastes it into PC-DMIS and from that point on, the appropriate object identifier is linked to the surface being measured. This type of functionality could easily be integrated into PC-DMIS, because PCDMIS has access to the CAD systems API.

3.1 Data Comparison and VPM Points

Comparisons between the measured and theoretical values are made to determine what the VPM should look like. At each measured data point, the distance between the measured points and the corresponding theoretical points is calculated. This calculation (Eqn. 1) describes how far each point along the surface deviates from its intended location. The API modeling function UF_VEC3_distance_to_plane( ) is used to obtain the sign (±) of the distances. The inputs to this function are the theoretical and measured data points and their corresponding surface normals, which are all contained in the inspection report.

$$\text{Distance} = \sqrt{\left(x_{meas} - x_{theo}\right)^2 + \left(y_{meas} - y_{theo}\right)^2 + \left(z_{meas} - z_{theo}\right)^2} \tag{1}$$

To create the VPM points, the theoretical points are offset by the distances calculated in Eqn. 1. The directions of the applied offsets are the surface normals. These directions are also the directions the probe travels when taking a hit. It is important to note that the offset could result in self-intersecting curves depending on the curvature of the object and the offset distances applied during translation. However, this is very unlikely because the offset distances are extremely small and was therefore disregarded in this research. Fig. 5 illustrates how the VPM points are created.

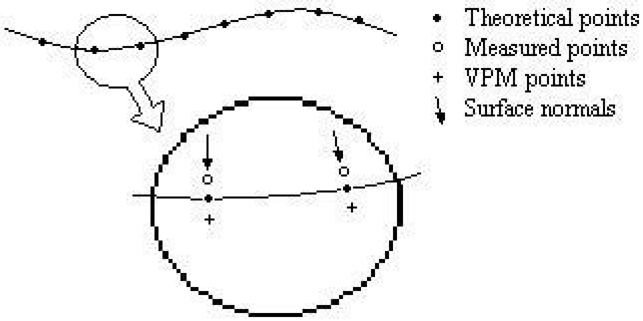

Fig. 5. Creating the VPM points.

After the VPM points are calculated, two additional points are added to each row of points. One point is added to the beginning and another to the end. These points are calculated to ultimately be located on the edges of the surface. They are intended to make sure the entire surface is modeled and appropriately represented. To create these points, the theoretical locations on the edges as well as the surface normals are needed.

To obtain the needed information a few API modeling functions are accessed. The first function obtains the surface's maximum and minimum locations that bound the surface. These locations theoretically construct a box wherein the surface resides. Fig. 6 shows a surface's bounding box.

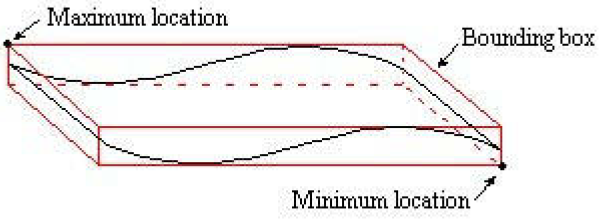

Fig. 6. Bounding box max/min locations.

The API modeling function used to obtain these max/min locations is called *UF_MODL_ask_bounding_box( )*. The input to this function is the object identifier. The output contains the max/min locations. The second API modeling function used returns a point on a surface given a reference point. The point on the surface is actually expressed as a u,v parameter. This parameter describes the point's location on the surface. The function is called *UF_MODL_ask_face_parm( ).*This function has two inputs, the surface's object identifier and a reference point. The reference point is used to locate the u,v parameter, which is the output of the function. The u,v parameter is the closest location on the surface to the reference point. The other output, which we will call *face_pnt* contains the x,y,z location of the point on the surface described by the u,v parameter.

The calculation of the reference point involves using the row's first and last theoretical points, the max/min locations of the bounding box and a dot product operation. To be simplistic, the row's first and last theoretical points will be referred to as row-end-points. The dot product is used to find the perpendicular distance from the row-end-points to the max/min locations. Two vectors are needed in the dot product calculation. The first is a unit vector that describes the probe's direction of travel as it moves along the row of points. It is calculated by subtracting the two row-end-points. The second is the vector from one of the row-end-points to the max/min location. A reference point is calculated by offsetting one of the row-end-points by the smallest calculated distance obtained through the dot product operation. The direction of the offset is the unit vector describing the probe's direction of travel. Offsetting in this direction allows the entire row of points to be aligned in the same direction.

With the surface point (*face_pnt*) acquired, the next step is to obtain the point's surface normal. The API modeling function used to accomplish this is called *UF_MODL_evaluate_face( ).* The function output is a pointer to the data structure containing the needed surface normal. The function has three inputs, the surface's object identifier, type of derivative to be computed, and the u,v parameter value that was obtained previously. The desired derivative value is the unit normal and is set equal to *UF_MODL_EVAL_UNIT_NORMAL*.

With all the needed information, the two additional points can be calculated and added to the series of VPM points. To calculate the new points, the same offset applied to the row-end-points is applied to the calculation of the new points as is shown in Eqn. 2. Fig. 7 shows the addition of a VPM point to the edge of a surface.

$$new\_pnt_{x,y,z} = face\_pnt_{x,y,z} \pm dis\tan ce \cdot surf\_norm_{i,j,k} \qquad (2)$$

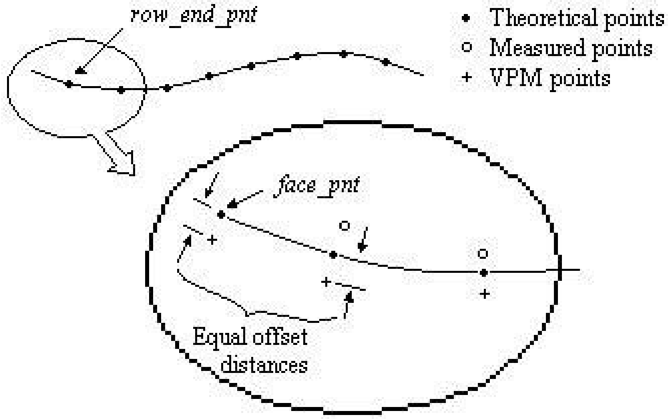

Fig. 7. Adding a VPM point to the edge of a surface.

3.2 Creating the Spline Through Points

The next step is to create a spline through each row of VPM points. The API modeling function, *UF_CURVE_create_spline_ thru_pts( )*, allows the user to create a spline, which passes through a number of points specified by the user. The function call has a number of necessary inputs and outputs the spline's object identifier. To use this function the spline's degree, periodicity (periodic or non-periodic), and the number of points are input to this function. In addition, the data structure containing the points, and other spline defining parameters are input. Fig. 8 shows the creation of three splines through their corresponding row of points.

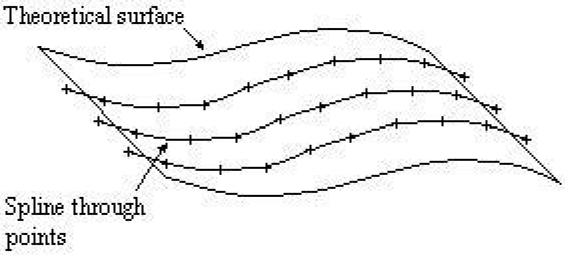

Fig. 8. Creating splines through points.

Since the goal is to completely represent the entire surface, two additional splines are created at the outer edges. The splines created at the outer edges are actually copies of the edge's nearest splines. This operation is carried out first by performing a dot product operation to calculate the distance from each edge to the nearest spline. Each of the spline's points are copied and translated to the edge's border. The dot product calculation uses the vector from the max/min location to a row-end-point and the unit vector perpendicular to the probe's direction of travel (vector from *last_end_pnt* to *first_end_pnt*). The unit vector is calculated using a cross product operation and the right-hand rule. The cross product uses the probe's direction of travel along the path and the probe's orientation (shown in Fig. 9). The cross product gives the vector perpendicular to the other two vectors. Fig. 10 shows the two additional splines translated to each edge.

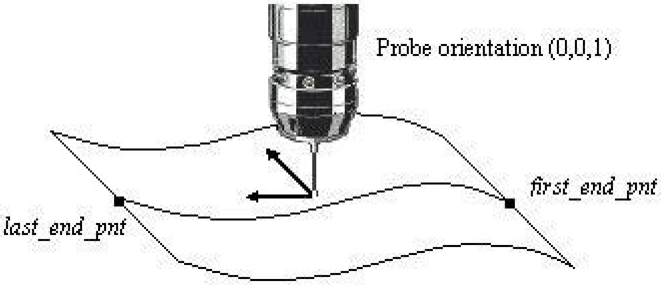

Fig. 9. Illustration showing the cross product operation.

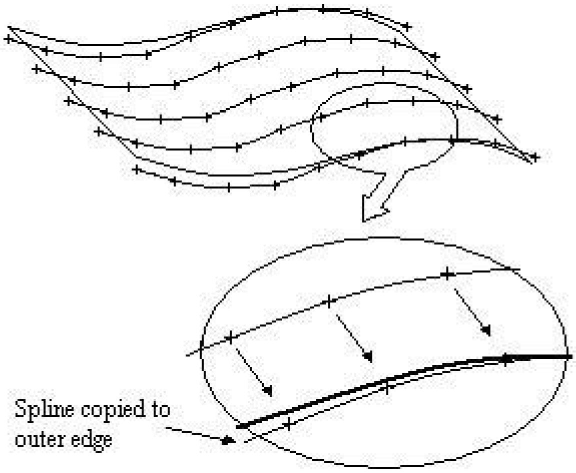

Fig. 10. Copying splines to the outer edges.

3.3 Creating the Through Curves Feature

The next step is to create a surface that encompasses each of the splines created in the previous step. The API modeling function used to perform this step is called *UF_MODL_create_thru_ curves( )*. This function allows the user to create a sheet body through a collection of curve outlines. The curve outlines are called section strings and can either be a curve, edge, or a face. The data structure containing the curve outlines (splines) is specified and input into the function. The other important inputs include: the type of patch, the degree of the surface, the type of body to be created (sheet body), and the surface's tolerance. The function output is the through curves feature identifier. Fig. 11 shows the creation of the through curves feature.

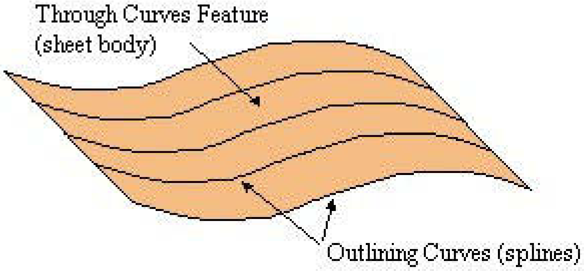

Fig. 11. Through curves feature showing outlining curves.

3.4 Replacing the Existing Surface

The final step is to replace the old surface with the new surface. The *UF_MODL_create_replace_ face( )* modeling function allows the user to replace a set of faces with another face. The critical inputs are the *target_ faces*, which represent the faces that will be replaced and the *tool_face*, which represents the new face. The *tool_face* is the through curves feature created in the previous step. Fig. 12 gives an illustration of a face being replaced by another.

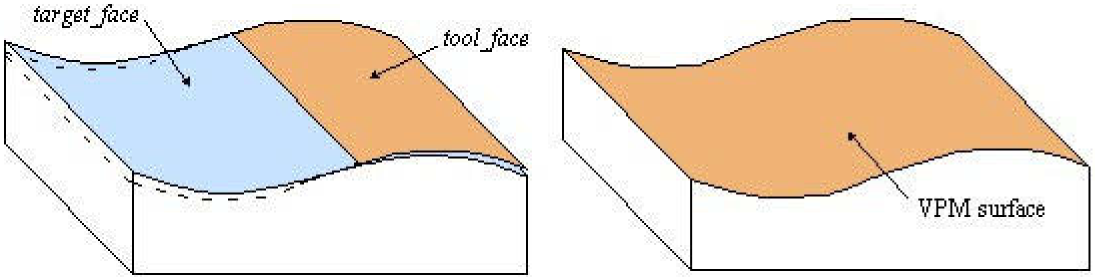

Fig. 12. Example showing the replace face modeling function.

4. VPM EXPERIMENTAL VERIFICATION

To validate the methods discussed in this paper, two tests were performed; one where a single row of points was measured along a non-planar surface, and the second where several rows of points were measured. The measurements were then evaluated against a line/surface profile tolerance. The first test used a linear open scanning procedure that included 1 row of 75 points. The second test used a UV Scanning Procedure to measure the surface profile. The scan included 10 rows of 50 points. A Sugino 3-axis machining centre was used to perform both the machining and inspection tasks. The parts were made out of aluminum and measured 80.45 x 63.44 x 30 mm. The CAD/CAM model (same for each test) was created in NX and PCDMIS was used for inspection planning. In order to bias some error into the system, a radial offset of 0.1 mm was applied to the part's non-planar surface (see Fig.13). A radial offset was chosen in order to appropriately compare theoretical and measured values. With this offset, the values correlate with each other; meaning the surface normals are identical along either surface. In theory, after machining, the surface would be 0.1 mm above its originally intended location and still have material overstock to correct for the radial offset and machine the intended profile.

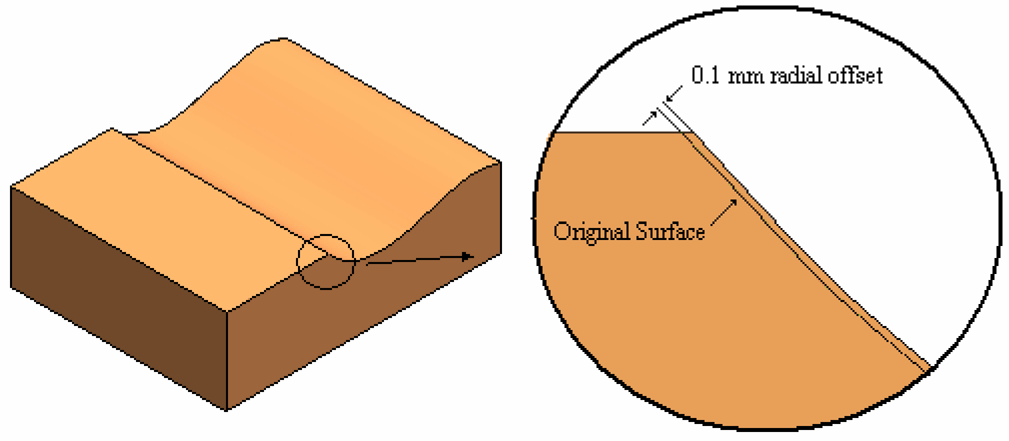

Fig. 13. Part used in Variational Part Model test.

The part was machined using the generated tool paths from NX and measured on the machine with a touch probe. The radial offset CAD/CAM model was used for machining and the original CAD model was used inside PCDMIS to compare theoretical and measured values. Tab. 2 lists the machining parameters used for tool path generation as well as the tool's type and size for each machining operation.

	Rough Cut	Semi-Finish Cut	Finish Cut
Part Intol	0.03 mm	0.03 mm	0.003 mm
Part Outol	0.12 mm	0.03 mm	0.003 mm
Part Stock	1.0 mm	0.3 mm	0.000 mm
Tool Type	Flat End Mill	Ball End Mill	Ball End Mill
Tool Diameter	7.9375 mm	3.1750 mm	3.1750 mm
% Step Over	30%	30%	10%

Tab. 2. Machining parameters used for tool path generation

After machining the first part, the theoretical and measured values of the part were compared. The error values (calculated as the distance between the theoretical and measured values) ranged from 0.10028 mm to 0.14725 mm. The average error was 0.11323 mm. After the VPM was created, the tool paths were regenerated and the part was re-machined. After re-machining, the part was measured and then the theoretical and measured values were compared. The smallest error measured on the VPM was 0.00283 mm (0.00011 in) and largest error measured was 0.04727 mm (0.00186 in). The average error measured was 0.03103 mm (0.00122 in). Fig. 14 shows a histogram illustrating how the measured error was distributed.

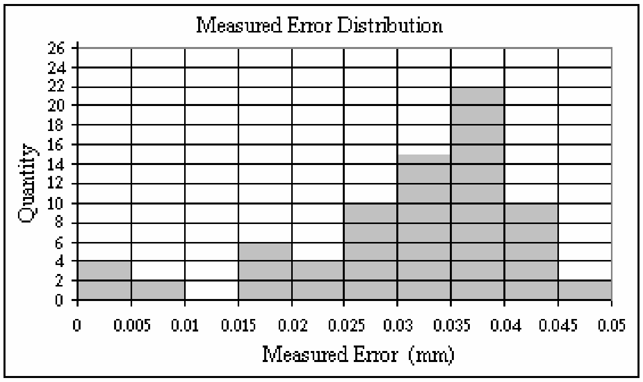

Fig. 14. Line profile VPM measured error distribution.

The error values of the second part ranged from 0.10395 mm to 0.16051 mm. The average error measured was 0.12484 mm. After creating the VPM and re-machining, the part was measured and the theoretical and measured values were compared. Fig. 15 shows a histogram illustrating the measured error distribution for the VPM. As can be seen, the smallest error measured on the VPM was 0.00849 mm (0.00033 in) and largest error measured on the VPM was 0.06637 mm (0.00261 in). The average error measured was 0.04278 mm (0.00168 in).

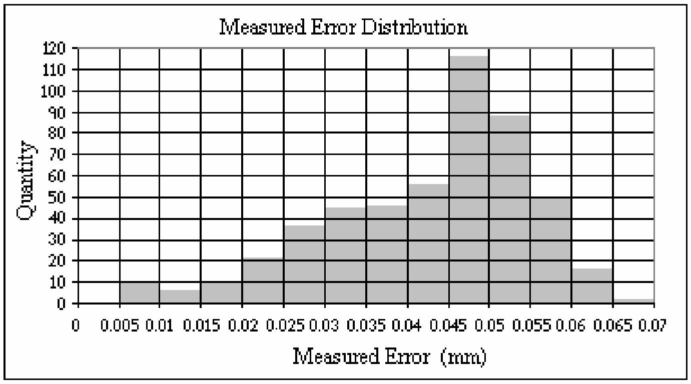

Fig. 15. Surface profile VPM measured error distribution

5. CONCLUSIONS

In this paper, a new closed-looped machining approach was presented to deal with dynamic machining error. In essence, the approach is to create a Variational Part Model that inherently possesses the dynamic errors associated with machining. The two tests demonstrate the validity of the Variational Part Model concept and prove that automatic closed-loop machining is possible. In addition, the proposed method could easily be integrated in an iterative part programming routine as well as be used with Geometric Dimensioning and Tolerancing (GD&T) principles to account for errors when related features are used such as orientation tolerances. Several methods for improving the accuracy of machine tools exist. Volumetric error modeling like that used in the geometrical error feedback routine compensate for a machine's ball screw errors, as well as a workspace's perpendicularity errors. Although none of these methods were implemented on the Sugino test mill, the results presented in this paper compare very well if not better than the referenced geometrical error feedback routine results, which did include volumetric error modeling [10]. With the application of accuracy improving methods, even better results could be obtained.

6. REFERENCES

[1] Koepfer, Chris, Consider Closing the Loop, *Modern Machine Shop*, 1999, pp 1-3.

[2] Cho, J., Cho, M., and Kim, K., Volumetric Error Analysis of a Multi-Axis Machine Tool Machining of a Sculptured Surface Workpiece, International Journal of Production Research, Vol. 32, No. 2, 1994, pp 345-363.

[3] Suh, S., Sohn, J., and Jung, S., A Close-loop Method For Reducing Total Machining Errors: Experiment and Analysis, *Transactions of NAMRI/SME*, , Vol. XXV, 1997, pp 311-316.

[4] Donmez, M., Blomquist, D., Hocken, R., Liu, C., and Barash, M., A General Methodology For Machine Tool Accuracy Enhancement by Error Compensation, *Precision Engineering*, Vol. 8, No. 4, 1986, pp 187-195.

[5] Kreng, V., Liu, C., and Chu, C., A Kinematic Model For Machine Tool Accuracy Characterization, *International Journal of Advanced Manufacturing Technology*, Vol. 9, 1994, pp 79-86.

[6] Klein, W., The Prediction of Cutting Forces in Endmilling with Application to Cornering Cuts, *MTDR*, Vol. 22, No. 1, 1982, pp 7-22.

[7] Fan, K., Lin, J., and Lu, S., Measurement and Compensation of Thermal Error on a Machining Center, *Proc. MATADOR*, 1991, pp 261-268.

[8] Suh, S., Cho, J., and Hascoet, J., Incorporation of Tool Deflection in Tool Path Computation: Simulation and Analysis, *Journal of Manufacturing Systems*, Vol. 15, No. 3, 1996, pp 190-198.

[9] Mou, J. and Liu, C., A Method For Enhancing the Accuracy of CNC machine Tools For On-Machine Inspection, *Journal of Manufacturing Systems*, Vol. 11, No. 4, 1995, pp 229-237.

[10] Suh, S., Lee, E. and Sohn, J., Enhancement of Geometrical Accuracy via an Intermediate Geometrical Feedback Scheme, *Journal of Manufacturing Systems* Vol. 18, No. 1, 1999, pp 12-21.

[11] Davis, T., Red, E., Carlson, S., Jensen, G., Sipfle, K., Flexible In-Process Inspection Through Direct Control, *Measurement*, In Press, Corrected Proof 2005.

[12] PC-DMIS 3.5 Reference Manual. Brown & Sharpe Manufacturing Corporation and Wilcox Associates Incorporated. 2002.

[13] Unigraphics Open C Reference Manual. EDS PLM Solutions.