A 3D-Compensating Algorithm for LNG Shipbuilding Profiles Robot Cutting System

Q.K. Chen, X.Y. Li
Hudong Zhonghua shipbuilding (Group) Co. Ltd.
Shanghai, China

L. Kong, Y.X. Wu
Shanghai Jiao Tong University
Shanghai, China

H.J. Wang, Z.Q. Ji
Shanghai Leader Soft Information Technology Co. Ltd.
Shanghai, China

Abstract —**The LNG shipbuilding profiles' deformation leads to lots of challenges for hi-precision shipbuilding, in order to achieve hi-precision manufacturing for deformed LNG shipbuilding profiles, a 3D-compensating algorithm for shipbuilding profile deformation is developed based on the profile deformation happening principles. The spatial coordinates on edge curve of the profile are measured and collected by a moving 2D laser sensor. Basing on the collected profile edge coordinates, an formula of polynomial curve is defined and fitted by the method of least squares, which takes accumulated chord length as parameters. Giving coordinates on a standard profile, the target coordinates on deformed profile can be calculated, eventually the deformation of the profile is compensated and the hi-precision manufacturing for deformed shipbuilding profile is fulfilled. Furthermore, the compensation algorithm has been implemented with VC++ and MATLAB mixed programming technology, which has been used in a robot flexible cutting system for serial production a famous shipbuilding company. The results show the 3D-deformation of shipbuilding profiles get compensated as expected, it also verified the algorithm is effective and efficient.**

Keywords-LNG ship; compensation algorithm; profiles cutting; laser measurement; MATLAB

I. INTRODUCTION

LNG shipbuilding profile cutting is one of most important and basic process in shipbuilding factory. Profile cutting means to cut the long raw metal material into different shapes of sections, and to make holes, assembly lines on them, these sections or profiles will be used at the assembly and welding process at the later stages. In practice, the shipbuilding profiles are very prone to deform during the shipping and storage period, and the shipbuilding profiles are always very long and narrow(0.5m*12m), this makes the deformation more serious [1], as shown in Figure 1.

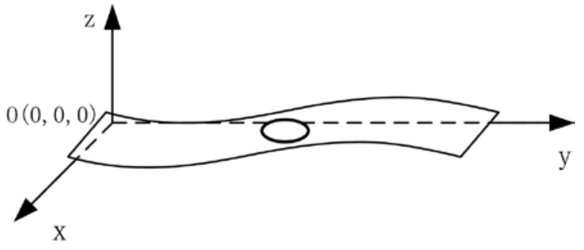

FIGURE I. DEFORMED PROFILE (SAMPLE).

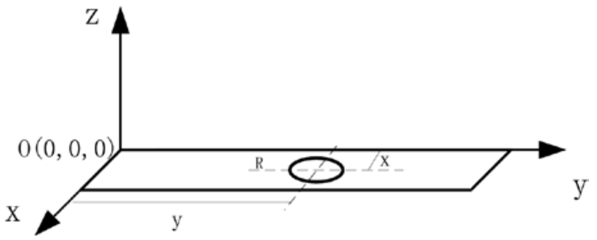

FIGURE II. STRAIGHTENED AND FLATTENED PROFILE.

The profile deformation makes lots of troubles in hi-precsion manufacutring process which is automatically fulfilled by shipbuilding robot cutting system. So it is certainly worth and necessary to solve this issue, to ensure the robot flexible cutting system can achieve hi-preccision graphic elements manufacturing even the profile is in deformation status, to make the coordinates of the graphic elements on the deformed profile equal to the intial desin values when the deformed prolife is straightened and flattened.[2]

The purpose of this paper is to investigate and implement a 3D deformation compensation cutting algorithm basing on the spatial curve of the profile edge; it achieves hi-precision manufacturing for the graphic elements on LNG shipbuilding profiles even if the profiles are deformed. The algorithm takes the mount of profile deformation into consideration, to compensate the coordinates for deformed profile cutting, to let the coordinates on straightened profile assembly statisfes the design commands, evevntually meets the hi-precision profile manufacturing requirements in the morden shipbuiding industry. This agrithom has been used and verified in a block assembly shop of a very famous Chinese shipbuiding company, the results show that the agrithom worked very well, and it improved the precision of profile manufacturing dramatically, it saves the cost and gains a big success, this agrithom has a large market prospective and good practice application values.

II. THE 3D-COMPENSATING CUTTING ALGORITHM BASING ON PROFILE EDGE CURVE

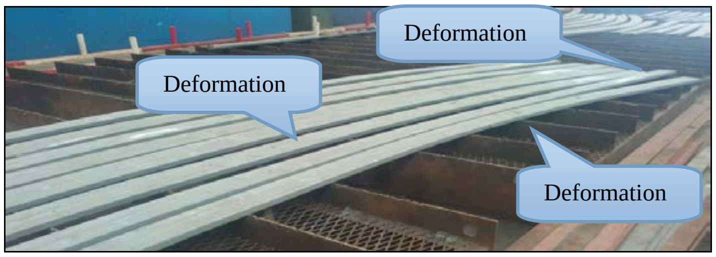

FIGURE III. SAMPLES OF PROFILE EDGE DEFORMATION.

As shown in Figure 3, shipbuilding profiles are prone to deform during operations such as transportation, storage, hoisting, and the deformation mechanism is very complex, summarizing the general requirement of the hull construction, the position compensation along the profile edge curve is every effective, so this paper presents a 3D-compensating algorithm basing on the deformed profile's edge curve; The algorithm effectively compensates the assembling errors caused by profile deformation, in the period of robot automatic cutting manufacture operations, it dramatically improves the cutting precision and manufacture quality.

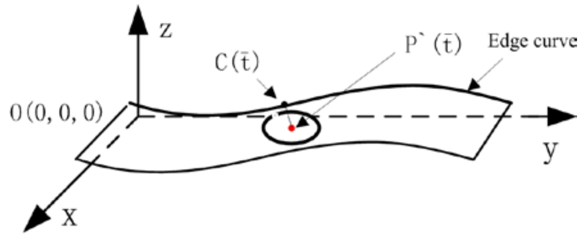

FIGURE IV. 3D MODEL OF PROFILE DEFORMATION.

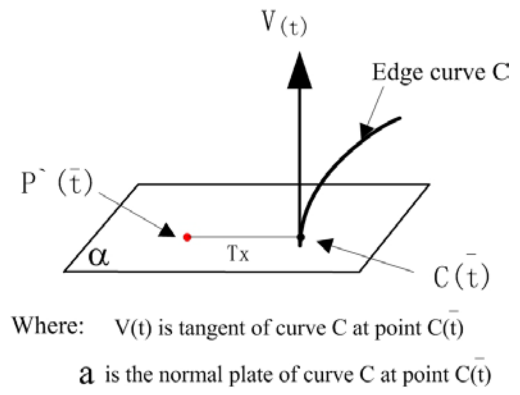
V(t)
Edge curve C
P` (t̄)
α
Tx
C(t̄)
Where: V(t) is tangent of curve C at point C(t̄)
a is the normal plate of curve C at point C(t̄)

FIGURE V. RESTRICTIONS BETWEEN CENTER OF CIRCLE COORDINATE $P'(\bar{t})$ AND SPACE CURVE POINT $C(\bar{t})$.

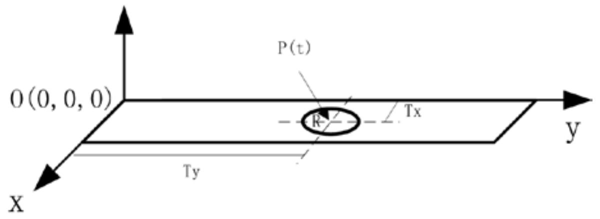

FIGURE VI. STRAIGHTENED AND FLATTENED PROFILE.

FIGURE VII. LASER SCANNER MEASURING.

Figure 4 shows the case that deformation occurred on the profile edge but need to keep the accuracy of the graphic elements coordinators on the profile. Meanwhile, even if the graphic elements are manufactured on the profile in deformed situation, it is necessary to make sure that all the coordinates of the graphic elements on the deformed profiles should equal the standard design value, when the deformed profile is straightened and flattened. Let's take manufacturing a circle hole on the profile as an example, Figure 4 shows the graphic element (the circle) on a deformed profile, comparing to the same graphic element on the standard profile Figure 6 (the straightened and flattened profile which will be finally assembled). Suppose a round hole is designed and cut on a standard profile, the round hole center coordinate is $P(t)=(T_x,T_y,0)$, the radius is R; in practice the hole should be manufactured on the deformed profile with high accuracy demand, and after this deformed profile gets straightened and flattened, the position of the hole center should equal the design value, as shown in Figure 6, means it should satisfy the following constrains, the horizontal distance from center of circle to the original point $O(0,0,0)$ should be T_y, the vertical distance to Y axle should be T_x and the radius should be R. As shown in Figure 5; if the normal distance from circle center point $P'(\bar{t})$ on the deformed profile to the profile edge curve equals T_x, and the curve length from the intersection point of normal plate α and edge curve to the origin equals T_y, then it is guaranteed that the coordinate of the circle center equals initial design value, when the deformed profile is straightened and flattened.

The graphic coordinate compensating algorithm descripted in this paper can effectively transfer any coordinate $P(t)$ on the standard profile to the according coordinate $P'(\bar{t})$ on the deformed profile, the transfer information will be fed to the robot flexible cutting system to compensate the deformation. In the shipbuilding robot automatic profile cutting system, there's a dedicated profile deformation detecting and measuring sub-system responsible for the deformation pre-measure. Basing on the deformation data, the compensation of the graphic elements CAD model for the coordinates can be fulfilled by the compensating algorithm, then the computer will transfer the coordinates for the compensated profile graphic elements, forms the robot spatial working coordinates, and communicates with the robot online, finally conducts the profile lining and cutting, implements the correct cutting and lining for deformed profiles. The profile deformation usually happens in the scope of space cubic curve, not happens in a higher curve for the deformation.

The coordinates P_i on the profile edge curve are measured and collected with a laser scanner, operation of the sensors is based on the principle of optical triangulation (Figure 8); radiation of a semiconductor laser 1 is formed by a lens 2 in a line 3 and projected to an object 7. Radiation scattered from the object is collected by the lens 4 and directed to a two-dimensional CMOS-array 5. The image of object outline thus formed is analyzed by a signal processor 6, which calculates the distance to the object (Z-coordinate) for each point of the set along the laser line on the object (X-coordinate).

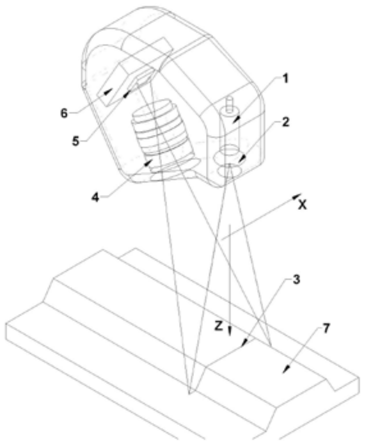

FIGURE VIII. THE LASER SCANNER STRUCTURE AND PRINCIPLE.

The fundamental principles for the compensating algorithm are: Detecting and gathering deformed profile edge space curve coordinates with the laser scanner, as shown in Figure 4 and Figure 8, fitting them with a cubic space curve. Given a serial points P_i on the profile edge space curve, the space curve can be presented by a third order polynomial, which takes cumulative chord length of ordered points P_i as parameters, as shown in polynomial (1), this method truly reflect the situation the data (the coordinates) are distributed according to the chord length. It is always regarded as the best parameterization method, and this method overcomes the shortages of the sampling points distribute unevenly along the chord but are parameterized evenly; in most cases this method can achieve a satisfied result [3].

$$C(t) = Q_0 + Q_1 t + Q_2 t^2 + Q_3 t^3 \tag{1}$$

In the equation Q_0, Q_1, Q_2, Q_3 are fitting parameters for the deformed profile edge curve, it can get by least square fitting method (equation 2).

$$\begin{bmatrix} N & \sum t_i & \sum t_i^2 & \sum t_i^3 \\ \sum t_i & \sum t_i^2 & \sum t_i^3 & \sum t_i^4 \\ \sum t_i^2 & \sum t_i^3 & \sum t_i^4 & \sum t_i^5 \\ \sum t_i^3 & \sum t_i^4 & \sum t_i^5 & \sum t_i^6 \end{bmatrix} \begin{bmatrix} Q_0 \\ Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} = \begin{bmatrix} \sum P_i \\ \sum t_i P_i \\ \sum t_i^2 P_i \\ \sum t_i^3 P_i \end{bmatrix} \tag{2}$$

Where Q_i is a vector, $Q_i = (Qx_i, Qy_i, Qz_i)$, and $t_i = \sum_{k=1}^{i} | P_k - P_{k-1} | / \sum_{k=1}^{n} | P_k - P_{k-1} |$ (i=1, 2,…n) $t_0 = 0$

P_i is the i's measuring point on the edge curve.

The first-order derivative of curve $C(t)$ is:

$$C'(t) = Q_1 + 2Q_2 t + 3Q_3 t^2 \tag{3}$$

Then the tangent vector of any point on the curve is:

$$V(t) = C'(t) / | C'(t) | \tag{4}$$

The unit vector from point $C(\bar{t})$ on the curve to the target point $P'(\bar{t})$ is:

$$V_1(t) = Z \times V(t) / | Z \times V(t) | \tag{5}$$

Finally, the compensating algorithm for converting the coordinate $P(t) = (T_x, T_y, 0)$ on the standard profile to the according point $P'(\bar{t})$ on deformed profile is:

$$P'(\bar{t}) = C(\bar{t}) + T_x V_1(\bar{t}) \tag{6}$$

Meanwhile, the equation should satisfy constrain formula:

$$\int_0^{\bar{t}} \sqrt{C'(t)^T C'(t)} dt - T_y = 0 \tag{7}$$

III. Algorithm implementation

Because there are lots of complex mathematic computations to fit the space edge curve on the deformed profile, calculate the arc length and get the slope of the normal, in order to improve the programming efficiency, the VC++ and MATLAB mixed programming technology is introduced in this paper, the MATLAB computation engine is responsible for all main computations [4-6], the 3D-compensating algorithm for deformed profile is implemented eventually.

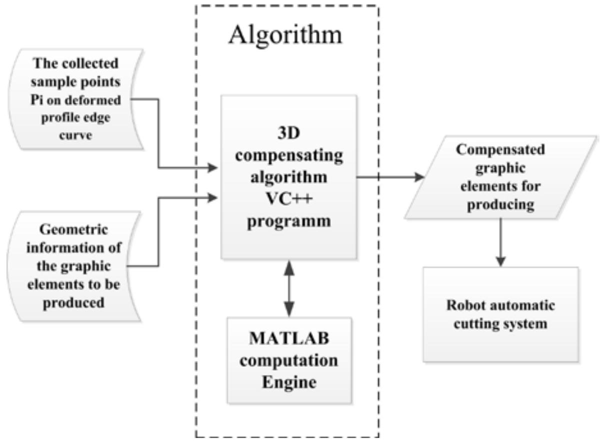
Algorithm
The collected sample points Pi on deformed profile edge curve
Geometric information of the graphic elements to be produced
3D compensating algorithm VC++ programm
MATLAB computation Engine
Compensated graphic elements for producing
Robot automatic cutting system

FIGURE IX. ARCHITECTURE OF 3D COMPENSATING ALGORITHM.

Algorithm implementation basing on VC++ and MATLAB mixed programming technology.

```
//
// Program start
//
//
//Measured points on the edge curve
//
Point3D arrayPoint[n] = {CPoint(p1), CPoint(p2),
CPoint(p3),..., Cpoint(pn)};
double arrayPointx[n];
double arrayPointy[n];
double arrayPointZ[n];
//Parameter t
double arrayParameterT[n];

void InitializeParameters(Point3D& arrayPoint[])
{
        double dLength_N = 0;
        doulbe dLength_I = 0;
        arrayParameterT[0] = 0;
        for (int i=1; i<n; i++)
        {
            arrayPointx[i] = arrayPoint[i].x;
             arrayPointy[i] = arrayPoint[i].y;
             arrayPointz[i] = arrayPoint[i].z;
             dLength_N += dLength_N + sqrt ( norm
             ( arrayPoint[i] - arrayPoint[i-1] ) );
        }
        for (int i=1; i<n; i++)
        {
            dLength_I = dLength_I + sqrt(arrayPoint[i]-
            arrayPoint[i-1]);
           arrayParameterT[i] = dLength_I/dLength_N;
        }
}
//
//The compensating algorithm
//
void AjustPositionData(Point3D& arrayPoint[], Point3D*
targetPoint)
{
        ASSERT ( targetPoint == NULL );
        mxArray *Px=NULL, *Py=NULL, *Pz=NULL,
        *t=NULL, *Cx=NULL, *Cy=NULL, *Cz=NULL,
        *Fx=NULL, *Fy=NULL, *Fz=NULL, *derCt=NULL,
        *Vt=NULL, *M=NULL, *G=NULL; *t1=NULL,
        *T1x=NULL, *T1y=NULL, *T1z=NULL,
        *Vt1=NULL, *Pt1=NULL, *C=NULL;
        //
        //Open MATLAB computation engine
        //
        Engine* pEng;
        If (!(pEng =engopen(NULL))
        {
            AfxMessageBox(“cann’t start matlab”);
            Exit(-1);
        }
        //
        //Initialize parameters, Pi, ti
        //
        initializeParameters(listPoint);
        Px=mxCreateDoubleMatrix(1,500,mxREAL);
        Py=mxCreateDoubleMatrix(1,500,mxREAL);
        Pz=mxCreateDoubleMatrix(1,500,mxREAL);
        t=mxCreateDoubleMatrix(1,500,mxREAL);
        mxSetName(Px,"Px");
        mxSetName(Py,"Py");
        mxSetName(Pz,"Pz");
        mxSetName(t,"t");
        mxSetData(Px, arrayPointx);
        mxSetData(Py, arrayPointy);
        mxSetData(Pz, arrayPointz);
        mxSetData(t, arrayParameterT);
        engPutArray(pEng,Px);
        engPutArray(pEng,Py);
        engPutArray(pEng,Pz);
        engPutArray(pEng,t);
        engPutArray(pEng,Y);
        //Curve fitting – x vector
        engEvalString(pEng,"Cx=polyfit(Px,t,3);");
        // Curve fitting - y vector
        engEvalString(pEng,"Cy=polyfit(Py,t,3);");
        // Curve fitting - z vector
        engEvalString(pEng,"Cz=polyfit(Pz,t,3);");
        //Ployder - x vector
        engEvalString(pEng,"Fx=polyder(Cx);");
        // Ployder - y vector
        engEvalString(pEng,"Fy=polyder(Cy);");
        // Ployder - z vector
        engEvalString(pEng,"Fz=polyder(Cz);");
        //C'(t) - C(t) first order derivative (the tangent vector)
        engEvalString(pEng,"derCt=[Fx, Fy, Fz];");
        //Vt – the unit vector at the point t on the curve
        engEvalString(pEng,"Vt=derCt/norm(derCt);");
        engEvalString(pEng,"M=sqrt(cross(derCt',derCt));");
        //Curve length
        engEvalString(pEng,"G=polyval(M,t)-polyval(M,0);");
        //parameter value t1
        engEvalString(pEng,"t1=solve('G-Ty=0');");
        //x value of point t1 on curve C
        engEvalString(pEng, "T1x=polyval(Cx, t1);");
        //y value of point t1 on curve C
        engEvalString(pEng,"T1y=polyval(Cy, t1);");
        //z value of point t1 on curver C
        engEvalString(pEng,"T1z=polyval(Cz, t1);");
        engEvalString(pEng,"Vt1=cross([0,0,1],Vt)/norm(cros
        s([0,0,1],Vt));");
        //The target point Pt1
        engEvalString(pEng,"Pt1=[T1x,T1y,T1z] + Ty*Vt1;");
        //The compensated coordinator
        C=engGetArray(pEng,"Pt1");
        targetPoint.x=(double*)mxGetData(C);
        targetPoint.y=(double*)mxGetData(C);
        targetPoint.z=(double*)mxGetData(C);
        //
        //Close Matlab computation engine
        //
        engClose(pEng);
}
//
//Program end
//
```

IV. Application of the algorithm

The 3D-compensating cutting algorithm presented in this paper has been successfully applied to the fabricating shop of a famous shipbuilding company in China. The main steps to use this profile cutting software system are, firstly, the mechanical design engineer exports the MGF file from a PLM (Product Lifecycle Management) system, and it will be loaded to profile cutting software system afterwards, with this inputted information combined with the collected coordinates on the deformed profile edge curve, the compensated positions on the deformed profile will be calculated for manufacturing. In the experiment, taking a 12m standard shipbuilding ball flat steel profile as a testing sample, the cutting information is imported to the system via MGF file format, the result shows the compensating algorithm is effective and applicable.

The graphic elements manufactured on the experimental profiles are shown in Figure 10, the size of the profile is: 12000mm (length) * 370mm (width); there are five round holes spotted evenly along the profile, the circle center is (100mm, 0), the radius is 50mm; the cutting result is shown in the following picture.

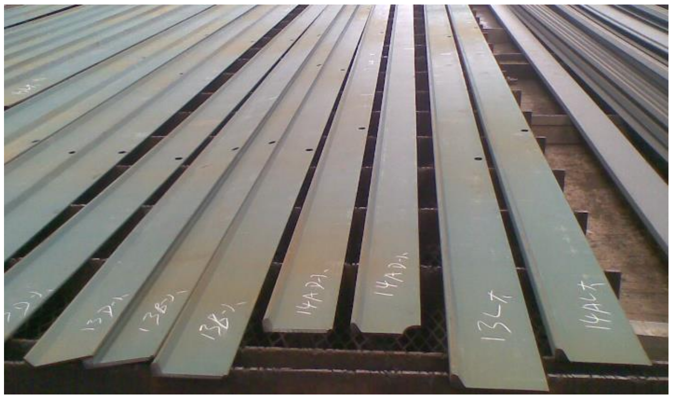

FIGURE X. PROFILE SAMPLES FOR CUTTING EXPERIMENT.

The target manufacturing materials for this compensating algorithm are shipbuilding ball flat steel profile, flat steel, angle steel and T-bar; basing on the material and thickness of the shipbuilding profiles, Kejllberg Hifocus 440i is adopted as the cutting machine, and ABB IRB2600 is introduced as manufacture robot. The cutting speed of the robot is 40mm/s, the cutting height is 6mm, cutting electric current is 40A~60A, the whole cutting cycle time is 400s. The system has been running for three months for the pilot production in the shipbuilding company, the results show the compensating algorithm improves the manufacturing precision dramatically for the deformed shipbuilding profile cutting, the system can compensate the deformation very well, this algorithm makes up to the defects which existed on the shipbuilding profile manufacturing process, it satisfies the demands of precision shipbuilding [7], and has a high productivity.

V. Conclusions

The cutting system basing on the LNG shipbuilding profile 3D-compensating algorithm, can automatically achieve the compensation for deformed shipbuilding profiles during the cutting process, it satisfied the profile cutting process requirements from shipbuilding industry, it realized the hi-precision manufacturing for shipbuilding profiles, it provided an efficient, reliable, and applicable solution for hi-precision manufacturing in the shipbuilding industry. The experiments show this algorithm can make up to the precision defects which was existing during manufacturing the deformed shipbuilding profiles before, it meets the hi-precision manufacturing demands in modern shipbuilding field.

References

[1] Mingyue Fan, 3D models driving robot off-line programming profile cutting system [D]. Tianjin：Tianjin University，2011:24-26

[2] Hongjie Wang, Guoqing Ding, Yao Shun.Robot auto-marking and auto-cutting of shipbuilding panels based on a compensation algorithm[J].Industrial Robot: An International Journal, 2001, 28(5):425-425

[3] Zhifeng Zhang, Numerical comparison between subdivision curve parameterization and choral parameterization [D]. Dalian：Dalian University of Technology，2009:23-26

[4] Yidong Mu, Ling Zhao, Research on mixed programming of VC++ and Matlab [J].Measurement & Control Technology.2013,(09):111-114.

[5] Stephen J.Chapman, MATLAB Programming with Applications for Engineers [M]. CL-Engineering, 2012, :-

[6] Joe Pitt-Francis, Jonathan Whiteley, Guide to Scientific Computing in C++ (Undergraduate Topics in Computer Science) [M]. Paperback, 2012, :-

[7] Yunqing Han,Tanhong Tu,Ping Tong, Shipbuilding precision control method[J].Journal of Heilongjiang Science and Technology.2013,(15):90.