Journal of Computational Design and Engineering 4 (2017) 192–202

Contents lists available at ScienceDirect

Journal of Computational Design and Engineering

journal homepage: www.elsevier.com/locate/jcde

# Introducing a novel mesh following technique for approximation-free robotic tool path trajectories ☆

Carmelo Mineo [a,*], Stephen Gareth Pierce [a], Pascual Ian Nicholson [b], Ian Cooper [b]

[a] Department of Electronic and Electrical Engineering, University of Strathclyde, Royal College Building, 204 George Street, Glasgow G1 1XW, UK
[b] TWI Technology Centre (Wales), Harbourside Business Park, Harbourside Road, Port Talbot SA13 1SB, UK

ARTICLE INFO

*Article history:*
Received 7 November 2016
Received in revised form 18 January 2017
Accepted 25 January 2017
Available online 16 February 2017

*Keywords:*
Tool path generation
Mesh following technique
Triangular meshes
Robotics
NDT

ABSTRACT

Modern tools for designing and manufacturing of large components with complex geometries allow more flexible production with reduced cycle times. This is achieved through a combination of traditional subtractive approaches and new additive manufacturing processes. The problem of generating optimum tool-paths to perform specific actions (e.g. part manufacturing or inspection) on curved surface samples, through numerical control machinery or robotic manipulators, will be increasingly encountered. Part variability often precludes using original design CAD data directly for toolpath generation (especially for composite materials), instead surface mapping software is often used to generate tessellated models. However, such models differ from precise analytical models and are often not suitable to be used in current commercially available path-planning software, since they require formats where the geometrical entities are mathematically represented thus introducing approximation errors which propagate into the generated toolpath. This work adopts a fundamentally different approach to such surface mapping and presents a novel Mesh Following Technique (MFT) for the generation of tool-paths directly from tessellated models. The technique does not introduce any approximation and allows smoother and more accurate surface following tool-paths to be generated. The background mathematics to the new MFT algorithm are introduced and the algorithm is validated by testing through an application example. Comparative metrology experiments were undertaken to assess the tracking performance of the MFT algorithms, compared to tool-paths generated through commercial software. It is shown that the MFT tool-paths produced 40% smaller errors and up to 66% lower dispersion around the mean values.


## 1. Introduction

Modern Computer-Aided Design (CAD) is used extensively in composite manufacture. Where it was once necessary to construct large items from many smaller parts, Computer-Aided Manufacturing (CAM) now allows these large items to be produced easily from one piece of raw material (through traditional subtractive approaches, or built up using more recent additive manufacturing processes (Gibson, Rosen, & Stucker, 2010). As a result, large components with complex geometries are becoming very common in modern structures.

Production engineers often face the problem of generating optimum tool-paths to perform specific actions on curved surfaces through numerical control machinery or robotic manipulators. The requirements of surface spray painting, surface coating, or Non-Destructive Testing (NDT) can be quite challenging to meet for complex shapes, especially when 100% coverage of the surfaces is required (Andulkar & Chiddarwar, 2015; Martin, 1967).

When working directly from available CAD models, there are a large number of Off-Line Programming (OLP) software packages available commercially that satisfy the requirements for such tool-path generation (e.g. MasterCAM®, Delcam®, Delmia®). However, when the original CAD model of the components is not available, photogrammetry or laser scanning must be used to create a point cloud of the surfaces of interest (Chikofsky & Cross, 1990; Varady, Martin, & Cox, 1997). There are also situations with composites

*Abbreviations:* CAD, Computer Aided Design; CAM, Computer Aided Manufacturing; NDT, Non-Destructive Testing; OLP, Off-Line Programming; STL, Standard Tessellation Language; MFT, Mesh Following Technique; CNC, Computer Numerical Control; NURBS, Non-Uniform Rational Basis Spline; CMM, Coordinate Measuring Machines; GUI, Graphical User Interface; SD, Standard Deviation; TOF, Time-Of-Flight.

☆ Peer review under responsibility of Society for Computational Design and Engineering.

* Corresponding author.
*E-mail address:* carmelo.mineo@strath.ac.uk (C. Mineo).

http://dx.doi.org/10.1016/j.jcde.2017.01.002
2288-4300/

manufacturing, compared to conventional light alloy materials, where the use of surface mapping metrology is required, even when the original CAD model is available. This situation can arise due to the inherent process variability associated with composites manufacture. Parts that are designed as identical may be affected by distortions when removed from the mould and may exhibit significant deviations from CAD (Zahlan & O'neill, 1989). These effects present a significant challenge for the execution of successive production operations.

When considering such parts, surface mapping leads, through the collection of point cloud data, to the generation of meshed CAD models of the parts (Fabio, 2003). Such models differ from precise analytical models, where all geometrical entities and spatial relationships are described analytically. Meshed CAD models are usually saved as Standard Tessellation Language (STL) files. The STL format is widely used for rapid prototyping and computer-aided manufacturing (Szilvśi-Nagy & Matyasi, 2003). The format only describes the surface geometry of a three-dimensional object without any representation of colour, texture or other common CAD model attributes. Whilst the conversion of analytical geometries into meshed surfaces is straightforward, the reverse process of conversion of a STL file into an analytical CAD model is challenging and time-consuming (Fabio, 2003).

Meshes represent 3D surfaces as a series of discreet facets, much as pixels represent an image with a series of coloured points. If the facets or pixels are small enough, the image appears smooth. Yet, if the surface is zoomed in enough, it is possible to see the pixelization or granularity and that the object is not locally smooth and continuous. This can lead to problems in the robot path creation (e.g. discontinuities and gaps) and explains the reason why existing commercial path-planning software requires precise part models, where the surfaces are mathematically represented.

This paper presents a novel algorithm based on a Mesh Following Technique (MFT) for the generation of tool-paths from STL models, suitable to overcome the difficulties encountered with current software applications that rarely support tessellated models as the input format for their embedded path-planning options.

## 2. Standard analytical approach

There is a wide range of algorithms suitable to generate boundary-conformed tool-paths for curved surfaces. Several studies have produced tool path generation methods for Computer Numerical Control (CNC) machining from STL models (Choi, Lee, Hwang, & Jun, 1988; Hwang, 1992; Jun, Kim, & Park, 2002; Ren, Yau, & Lee, 2004). However, these methods generate tool paths by approximating the polyhedral models and most of them focus on tool-paths for CNC machinery with limited number of axes (e.g. three-axis). A typical standard approach (Wang, Zhang, Scott, & Hughes, 2011) to convert tessellated STL surfaces into analytic surfaces is to approximate using Non-Uniform Rational Basis Spline (NURBS) or polynomial reconstruction. NURBS surfaces are mathematical representations of curves and surfaces; they are capable of representing complex free form surfaces that are inherently smooth. NURBS can be easily converted to meshes at any time, in the same way that one can easily take a digital image of an object with a camera. Conversely, going from meshes to NURBS is like trying to reconstruct the object from a pixelated digital image – it is a much more difficult task. NURBS surfaces are generated by a series of NURBS curves in two directions (called *U* and *V*) interpolated to create a surface. There are no quick automatic methods to convert tessellated surfaces to NURBS. Some CAD applications (e.g. Rhinoceros® by McNeel) include conversion tools, but consider only the simplest case of NURBS surfaces – the so called *bilinear surfaces* defined by 1 degree NURBS curves (i.e. lines) in both directions (Piegl & Tiller, 2012).

Curve fitting is the process of approximating a pattern of points with a mathematical function (Arlinghaus, 1994). Fitted curves can be used to infer values of a function where no data are available (Johnson & Williams, 1976), overcoming the discretization of pixelated or tessellated models. Regression analysis provides robust statistical tools to estimate how much uncertainty is present in a curve that is fit to discreet data points (Freund, Wilson, & Sa, 2006). The goal of regression analysis is to model the expected value of a dependent variable $y$ in terms of the value of an independent variable (or vector of independent variables) $x$. In general, the expected value of $y$ can be modelled as a $n$th degree polynomial function, yielding the general polynomial regression model based on the truncated Taylor's series:

$$y = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n + \varepsilon \quad (1)$$

where $\varepsilon$ is a random error with mean zero. Conveniently, these models are all linear from the point of view of estimation, since the regression function is linear in terms of the unknown coefficients $a_0, a_1, \ldots, a_n$. Therefore, for least squares analysis, the computational and inferential problems of polynomial regression can be completely addressed using the techniques of multiple regression. This is done by treating $x, x^2, \ldots, x^n$, as being distinct, independent variables in a multiple regression model.

An initial attempt was made to approximate meshes with polynomial analytical surfaces. A MATLAB® surface fitting toolbox (D'Errico, 2010) was used. Given the vertices of the tessellated model and the order of the target polynomial function, *the fitting algorithm* provides the coefficients of the fitting function. The order of the polynomial function can be progressively increased until the approximation error falls below a set threshold, at the expense of increasing the computation time. At each iteration, the residual errors and the *R*-squared parameter of the regression can be computed. The *R*-squared parameter is a statistical measure of how close the data are to the fitted regression surface. It is also known as the coefficient of determination. *R*-squared is always between 0% and 100%, and indicates how well the polynomial model explains the variability of the data around its mean. However, it is possible to obtain high *R*-squared values also for a model that does not fit the data well. Therefore, the analysis of the maximum of the residual errors is more useful in practice. Monitoring the residual errors is also useful to assure the stability of the approximation algorithm; the increase of the polynomial order can be ceased when the residual errors begin to diverge.

The use of this approach highlighted some important limitations. Even classic primitive geometric surfaces can be surprisingly difficult to approximate, leading to coarse approximation errors. The iterative polynomial approximation was applied to the tessellated surface of one quarter of an ellipsoid with semi-major axis of 1 m and semi-minor axes of 0.5 m (Fig. 1a). The surface is represented by a triangular mesh with 1914 vertices and 3557 triangles. The maximum approximation error decreases up to the 30th order function, reaching a minimum value of 8.9 mm, before starting diverging to higher values. Fig. 1b and c shows respectively the 3rd order and the 30th order fitting surfaces, superposed to the mesh vertices.

Fig. 2 shows the maximum error, the *R*-squared parameter and the computation time, plotted against the order of the fitting polynomial function. All the algorithms presented in this paper were implemented in MATLAB® codes and tested using a Windows 10 based computer with 2.7 GHz Intel i7 processor. The computation time increased exponentially with the polynomial order, jumping from few milliseconds for the approximations with the lowest