Journal of the Nigerian Association of Mathematical Physics
***Volume 10 (November 2006),* 275 - 292**
© J of NAMP

Fairing NURBS curve by dual parameter optimization

John A. Akpobi and Ufuoma D. Egbedi
Department of Production Engineering, University of Benin, Benin City, Nigeria.

Abstract

The curve fairing problem has seen many innovations especially in Computer-Aided Design (CAD) applications where product design depend largely on aesthetic, producibility and functional requirements. A major factor for evaluating these requirements is the geometric fairness of the product being modelled. This paper addresses the geometric fairing problem in which we model the shape of the product using Non Uniform Rational B-Splines (NURBS). The concept of curvature plot is used to interrogate the curve for defects and the corresponding knot and weight (at the defective regions) are sequentially modified in a sense that a fair curve ultimately results. Finally, results of our implementation are presented to show the validity of the proposed scheme.

Keywords*:* NURBS, virtual array, convexity, knot vector, homogeneous coordinate vector, inflection point, curvature discontinuity, curve fairing.

1.0 Introduction

As a result of high customer awareness in addition to frequently varying expectations, there is the need for product developers to keep abreast with the current trend to produce customer compliant products. In order to satisfy the need for products to assume complex shapes, shape (curve) optimisation techniques like Non Uniform Rational B-Splines (NURBS), has emerged. With the development of high-speed computers, form design (an aspect of product design that examines the shape and appearance of a product) has become an essential process by which the producibility, aesthetic and functional characteristics of a product are investigated. Typically, such characteristics may include the hydrodynamic properties of a car body, an aeroplane fuselage or a ship hull among others. It is practically possible to improve on these properties, if the designer is able to model the object correctly using its geometric (curve and surface) representations. This has motivated the field of geometric modelling. Geometric modelling is a wide field of study and includes 2D and 3D geometry, wire frame, surface and solid modelling, and parametric, variational and relational geometry etc. An important requirement in geometric modelling of a product is its geometric fairness. This characteristic can be investigated by modelling the product surface. The surface of a product may be viewed as a set of smoothly attached patches derived by well blended intersecting curves. The fairness of these curves is based partly on the subjective judgement of the designer and partly on the current situation in which the designer is working. However to ensure that products meet their aesthetic, producibility and functional requirements, their geometric representations should be free of all unwanted features [1] which may exist as cusps, inflection points, wiggles and excessive flat regions etc.

The vast research work that has been performed so far in the curve fairing environment may be grouped into two categories, namely: local fairing and global fairing. Local fairing is the process by which a curve is made subjectively better in a sense that only those segments of the curve that contain unwanted features are modified and as a result the curve becomes optimum and feasible every where. In global fairing, a curve is faired globally based on the assumption that every segment of the curve can be potentially made better. The disadvantage of this strategy is that some initial good features may be lost. Thus, the resulting curve is optimum everywhere but may not necessarily be feasible everywhere. In
practical terms, it could therefore be inferred without much loss of generality that the former scheme is characterized by a more even trade-off (between aesthetics and functionality) while the latter is characterized by a "tilted trade-

off" (between aesthetics and functionality), in this case more in the aesthetic direction. We therefore in this paper, adopt a local fairing strategy in which the defective curve segment(s) is/are identified using the curvature plot concept and subsequently modified using a nonlinear optimization technique [2].

Non Uniform Rational B-Splines (NURBS) presentation can be found in several computer- aided design (CAD) and computer graphics (CG) literatures. Rogers and Adams [3] gave a detailed mathematical presentation of B-spline curves and surfaces, generally while Burger and Gillies [4], and Watt [5] gave a rather detailed theoretic presentation. Chalmovinsky and Jüttler [6] developed methods for the variational design of algebraic curves based on geometric criteria such as elastic bending energy. They formulated the fairing problem as a constrained optimization problem and used the Sequential Quadratic Programming (SQP) technique to attack the problem. In Huang et al [7] an algorithm for raising the degree of B-splines was proposed. The algorithm was based on first, computing derivatives from control points, next, knot vector re-sampling and finally, computing new control points from derivatives. Goldenthal and Bercovier [8] introduced a novel method of unified optimal control over knot vectors of non-uniform B-splines, which was applied to cases that violate the Schoenberg-Whitney condition. Poliakoff et al [9] proposed an automatic algorithm for fairing B-spline curves. Their formulation was an extension of the Kjellander's algorithm (which was used to adjust "bad" data points on a uniform parameterized curve so as to create a jump of the third derivative of the curve at the point to be equal to zero). Hahmann [10] proposed a local method for automatic fairing of bi-cubic B-spline surfaces. In this method, a local fairness criterion detects the region where the spline needs to be faired and the control net is modified there. Eck and Hadenfeld [11] proposed a local fairing technique for automatic fairing of B-spline curves based on minimizing me integral of the squared l^{th} derivative of a given curve interactively by changing a single control point where the highest improvement of the energy integral is to be expected in every step. Moreton and Sequin [12] applied nonlinear programming techniques to minimize a fairness functional based on curvature variation. Liu et al [13] employed a constrained smoothing B-spline curve fitting technique for mesh curves of ships by minimizing energy functional as fairness metric. Nowacki et al [14] presented an approximation scheme based on minimization of the strain energy sum of mesh lines and potential energy of strings attached to the data points. McCallum and Zhang [15] proposed an automatic smoothing algorithm based on B-spline curvature property and its application in ship design. Pigounakis and Kaklis [16] presented an automatic fairing scheme for fairing cubic parametric B-splines under convexity, tolerance and end constraints. In Juhasz and Hoffmann [17], and Speer et al [18], a geometric approach to curve fairing was presented such that the knot vector placement problem is considered a non-optimization problem. In Renner et al [19], a genetic algorithm approach was used to optimize the knot vector. Sapidis [20] proposed an automatic algorithm for fairing B-spline curves using uniform and non-uniform parameterized knot vectors. His key idea is to fair the curve at the knot value that corresponds to the highest curvature discontinuity. This knot is known as the offending knot and is faired first. Aszódi et al [21] presented a NURBS fairing algorithm based on knot vector optimization. The approach used in their work is to optimize the knot vector by the method of simulated annealing such that the process tries to find the knot value for which the quality of the curve is optimum. In a practical sense, the method of simulated annealing is based on the metaphor of the cooling process of a material composed by particles. The inherent problem in this technique is that if cooling is allowed to proceed too fast, the algorithm may miss the global minimum and end up in a local minimum. Alternately, if the rate is slow, then a global minimum is very possible though this will increase computation time. The reason for this behavior can be explained thus (Otten and van Ginneken [22]): minimum energy states are called ground states in condensed matter physics though experiments showed that an extremely low temperature does not guarantee that a system is in its ground state or close to that state. A technique called annealing was therefore developed to bring a substance into one of its ground states. It starts from a state in which the substance is melted. Then the temperature is slowly lowered, slowly enough to keep the system in a state of quasi-equilibrium. When the temperature is lowered too fast, the resulting crystal may have many defects, or even lack all crystalline order. To apply this concept in simulation and modelling of curves, a trade-off between fairness and computation time has to be made. This paper features an automatic scheme for fairing NURBS curves. Our work uses the Sapidis [20] criteria for identifying defects in the curve segment. Our contributions include: (a) the development of *a novel but simple method for* computing both the rational B-spline functions and the parametric curve in each segment, and (b) the extension of Sapidis fairing criteria to the seemingly complex NURBS, where we optimize the homogeneous coordinate vector in addition to the knot vector. The presentation here includes *a novel but simple approach to* computing the rational B-spline basis functions. *Section 2* is focussed on NURBS formula. . In *Section 3*, details of the fairing scheme are presented. *Section 4* features examples and results, while conclusions and future work are presented in *Section 5*

2.0 NURBS Formula

NURBS stands for non – uniform rational B-splines. They are equations used to define curves or surfaces that simulate the designer's pattern in terms of stiffness and continuity [23]. In other words a NURBS curve is a rational B-spline curve that is defined over a non-uniform knot vector. For over two decades ago, NURBS [3] has been used as an initial graphics exchange standard {IGES) through which interchange of design data can be made between two CAD systems or between CAD and CAM systems. NURBS has been well known and used as a versatile tool in shape modelling due to its mathematical capability to represent all types of curves, planes, conic sections and free-form shapes. It is thus regarded as a unified surface geometry. NURBS curves are polynomials, which are smoothly joined together by knots. The curve has a non – decreasing knot vector, a homogeneous coordinate vector and rational B – spline basis functions. Each weight in the homogeneous coordinate vector is attached to a vertex point. These weights affect the shape of the curve locally in the vicinity of the vertex point.

A cubic NURBS curve may be defined parametrically by the equation:

$$C(t) = \sum_{i=1}^{n+1} B_i R_i^k(t) \tag{2.1}$$

$$R_i^k(t) = \frac{h_i N_i^k(t)}{\sum_{i=1}^{n+1} h_i N_i^k(t)} \tag{2.2}$$

$B_i = [B_{xi} \quad B_{yi}] =$ control points

$h_i =$ weights

$t =$ curve parameter

$N_i^k(t) =$ B-spline basis function

$R_i^k(t) =$ Rational B-spline basis function

In this paper, we present a novel but simple approach for computing the parametric NURBS curve. The approach here is based on the concept that the curve is made up of smoothly joined segments; which implies that it is piecewise continuous. To exploit this concept we compute each segment of the curve and sum them up discretely. A NURBS curve of order k defined by $n+1$ control points has the following primary characteristics:

***Property* 1**

There are a total of $n+k+1$ knots for which $n-k+1$ of the knots are interior and "controllable" or "active" knots, leaving k-multiplicity of knots at both ends of the knot vector as "uncontrollable" or "passive" knots.

***Property* 2**

The number of curve segments which make up the curve is given by the expression, $w = n-k+2$.

***Property* 3**

The basis function dependencies for the curve are given by the array:

$$\begin{array}{llllllll}
N_1^k & N_2^k & \ldots & N_{n+1}^k & & & & \\
N_1^{k-1} & N_2^{k-1} & \ldots & \ldots & N_{n+2}^{k-1} & & & \\
. & . & \ldots & \ldots & \ldots & . & & \\
. & . & \ldots & \ldots & \ldots & \ldots & . & \\
N_1^1 & N_2^1 & \ldots & \ldots & \ldots & \ldots & \ldots & N_{n+k}^1
\end{array}$$

2.1 Forming Virtual Arrays

The basis function dependencies in ***Property 3*** above are decomposed into w numbers of $k \times k$ "virtual arrays" as follows:

1st virtual array (segment 1, w = 1):

$$\begin{array}{llll} N_1^k & \dots & \dots & N_k^k \\ . & \dots & \dots & . \\ . & \dots & \dots & . \\ N_1^1 & \dots & \dots & N_k^1 \end{array}$$

2nd virtual array (segment 2, w = 2):

$$\begin{array}{llll} N_2^k & \dots & \dots & N_{k+1}^k \\ . & \dots & \dots & . \\ . & \dots & \dots & . \\ N_2^1 & \dots & \dots & N_{k+1}^1 \end{array}$$

(n+2-k)th virtual array (segment n+2-k, w = n+2-k):

$$\begin{array}{llll} N_{n+2-k}^k & \dots & \dots & N_{n+1}^k \\ . & \dots & \dots & . \\ . & \dots & \dots & . \\ N_{n+2-k}^1 & \dots & \dots & N_{n+1}^1 \end{array}$$

Summarily, the above formulation shows that, for a curve of w segments, $n+1$ control points and order k, virtual arrays are possible from $w = 1$ to $w = n\text{-}k+2$ and may be recursively computed from order $c = 1$ to $c = k$. Hence the w^{th} virtual array may be de fined as:

$$\begin{array}{llll} N_w^k & \dots & \dots & N_{w+k-1}^k \\ . & \dots & \dots & . \\ . & \dots & \dots & . \\ N_w^1 & \dots & \dots & N_{w+k-1}^1 \end{array}$$

2.2 Computing B-spline basis functions

The B-spline basis functions in each segment are computed based on the following rules:

***Rule* 1**

Locate the element labelled N_{w+k-1}^1 in each virtual array and assign it a value 1.

***Rule* 2**

Perform a hill climbing motion (↑, ←, ↑, ← etc) starting at position N_{w+k-1}^1 and stopping at position N_w^k. For instance, considering a fourth level curve with six control points, the following could be derived for its *2nd virtual array:*

$n+1=6$

$w=2$

hence $c=1$ to $c=4$

$w+k-1=2+4-1=5$

In this sense the *2nd virtual array* becomes:

$$\begin{array}{cccc}
N_2^4 & N_3^4 & N_4^4 & N_5^4 \\
N_2^3 & N_3^3 & N_4^3 & N_5^3 \\
N_2^2 & N_3^2 & N_4^2 & N_5^2 \\
N_2^1 & N_3^1 & N_4^1 & N_5^1
\end{array}$$

***Rule* 3**

Assign a value of zero to all elements on the left of the hill-climbing boundary and outside the range; while elements on the right of the hill-climbing boundary are computed using the Cox-de Boor formula:

$$N_i^k(t) = \frac{(t - x_i)N_i^{k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N_{i+1}^{k-1}(t)}{x_{i+k} - x_{i+1}} \qquad (2.3)$$

For instance, in the *2nd virtual array* of a cubic NURBS curve defined by six control points, the computation of $N_5^2(t)$ may result in the term $N_6^1(t)$. Obviously, this latter term is outside the range of this virtual array and must therefore be assigned a value of zero. Terms such as $N_4^1(t)$, $N_3^2(t)$, $N_2^2(t)$, $N_2^1(t)$, $N_3^1(t)$ and $N_2^3(t)$ must all be assigned a value, 0 and $N_5^1(t)$, $N_4^2(t)$, $N_5^3(t)$, $N_4^3(t)$, $N_3^3(t)$, $N_5^4(t)$, $N_4^4(t)$, $N_3^5(t)$ and $N_2^4(t)$ are computed using equation (2.3) above.

***Rule* 4**

Any computation using equation (2.3) that results in $\frac{0}{0}$ must be assigned a value of zero.

2.3 Computing rational B-spline basis functions

The rational B-spline basis functions that define each segment are computed using only those elements in the top row of the corresponding virtual array.
Generally, for an arbitrary segment, *w*

$$R_i^k(t) = \frac{h_i N_i^k(t)}{\sum_{i=w}^{w+k-1} h_i N_i^k(t)} \qquad (2.4)$$

NURBS contains an additional blending tool called the homogenous coordinate vector. It is a row vector containing values called weights, each attached to a vertex point. In specifying the homogenous coordinate vector, the external weights h_i and h_{n+1} are each assigned a value, 1 while the interior weights may be intuitively assigned values in the range 0 to n+1. Although each control point has an attached weight, the amount of change in the shape of the curve cannot be precisely predicted by the user. To ensure controllability of the curve when using weights, we intuitively recommend the range $0.25 \le h_i \le 1$.

3.0 NURBS fairing

This paper adopts a local faring strategy to control the shape of the curve. This strategy recommends that the curve be modified only at “defective segments”. To achieve this objective the curvature plot of the curve is investigated and all defects in the curve are identified. To identify defective segments, we investigate the behaviour of the curve from one knot to another, along the curve. Since the knot vector defines how the polynomial pieces are blended together, a slight change in the value of a knot will have a significant effect on the shape of the curve in that vicinity. For this reason, the behaviour of the curve with respect to the knots can better be predicted by considering the knots. We therefore use a scheme that exposes the defective segments as we investigate the curve from one knot to another. Although the knots can be used to achieve shape control of the curve, NURBS has an additional shape control tool (the homogeneous coordinate vector) which contains weights, each attached to a vertex point. If these weights are well specified and manipulated concomitant with the corresponding knot, a better shape control of the curve can be achieved [1] and [23]. To this end, we propose that for any defective knot, x_j , the corresponding weight should be h_{j-k+2} since this weight is capable of fine tuning (causing minimal shape changes in) the curve at the point, x_j .

The behaviour of the curve with respect to a particular control point when the associated weight is assigned values (h_i < 1 and h_i > 1, $\forall\ 0 \le h_i \le n+1$) is of interest. The curve tends to increase its positive convexity relative to a vertex point when the corresponding weight is decreased from h_i = 1 to h_i = 0. Alternatively, when h_i is increased from 1 to say n+1, positive convexity tends to decrease. At both extremes (h_i = 0 and h_i = n+1), positive convexity is maximum (making the curve touch the vertex point), and zero (making the curve flat), respectively.

3.1 Identification of defective curve segments

A fair curve is one that satisfies its aesthetic, producibility and functional requirements. In other words, the curve should look visually pleasing as well as preserve the expected performance characteristic. A product is objectively beautiful if it is aesthetically compliant as may be judged by every user. To apply this concept to shape modelling of objects, it is necessary that a suitable quality measure be specified. This quality measure should be defined mathematically so as to be able to automate the fairing process. Automatic faring has two major advantages. First, it makes possible high-speed computation, thus reducing labour cost/time. Second, the results obtained are more consistent, thus more acceptable universally.

Burchard et al [1] theorised that: beautiful objects are free of unessential features and simple in design. Based on this concept, a product will meet its aesthetic requirements if it is characterised by a smooth and visually pleasing shape. With the development and use of technologies such as CAM, CNC, robotics etc., it is essential that products be sufficiently investigated for geometric fairness during their design. In this way such products will meet their aesthetic, producibility and functional demands. Geometric modelling has revealed that unessential features in products may include cups, curvature extrema, loops, and unwanted inflection points which in turn require different mathematical techniques for identifying them. For convenience, these unwanted features (defects along the curve segments) may be grouped into the following two classes:

Class I Curve Defects

These defects are due to the presence of inflection points. Basically an inflection point, where it exists in the curve may not represent a defect if it is implied by the vertex points in that vicinity. In this case the convexity of the curve is prescribed. On the other hand an unwanted inflection point (in a point segment) is not usually implied by the vertex points in that vicinity and as such the curve must be modified in that region so as to minimize the effect of the unwanted feature.

Class II Curve Defects

These defects are due to the presence of such features as loops, curvature extrema, cusps (discontinuities) etc. It is necessary in geometric modelling that these features be identified, and the curve modified where they exist so as to optimize the curve.

This paper features the use of curvature plot to identify defective curve segments. Curvature plots are good tools for investigating the behaviour of a curve with respect to the knots. From the curvature plot, it is possible to infer whether an inflection point exists within a knot internal or not. In this sense it can be used to derive a suitable fairness indicator. The curvature plot could therefore be regarded as an "interior property" as it is capable of revealing defective segments which ordinarily may not have been noticed cosmetically. For convenience, the 3D NURBS curve is here modelled parametrically in 2D as:

$$C(t) = \left[x(t), \quad y(t)\right] \tag{2.5}$$

To investigate this curve for fairness using the curvature plot, we compute the plane curvature, given by the relation:

$$\kappa(t) = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{[\dot{x}(t)^2 + \dot{y}(t)^2]^{3/2}} \tag{2.6}$$

The term $\kappa(t)$ is a signed quantity; as a result changes in the sign of $\kappa(t)$ as we progress from one knot to another may be given different interpretations. For a typical curve, a change in sign of $\kappa(t)$ from one reference point (knot) to another signifies the presence of an inflection point within the said interval. However, because of the multiplicity of knot values at both ends, the curvature plot will be a plot of $\kappa(t)$ against x_i in the range $x_k \le x_i \le x_{n+2}$. The existence of an inflection point in a curve segment is characterized by a curvature sign change. Mathematically, this makes possible the product of adjoining curvature values with opposite signs to be a negative value. This concept can reveal the presence of an inflection point. Hence the criterion for existence of an inflection point is defined thus:

***Criterion* I**

For a curve segment characterized by end knots x_i and x_{i+1}, an inflection point exists for this interval if the condition given by equation (2.7) holds:

$$\kappa(x_i)\,.\,\kappa(x_{i+1}) < 0 \tag{2.7}$$

Although, *Criterion I* can be used to identify a point of inflection, it can not show whether it is desired or unwanted. The question then is: *when is an inflection point desired or unessential?*

An inflection point when it exists in a curve segment is desired when and only when it is implied by the defining vertex points. To prove whether an inflection point is implied by the defining vertex points it is essential that we examine the change in convexity of the polygon as we move from segment to segment within the region of interest. In our formulation in Sections 2.2 and 2.3, we noticed that a single segment is defined by four vertex points. Typically, segment 1 is defined by vertex points: B_1, B_2, B_3 *and* B_4, segment 2 is defined by vertex points: B_2, B_3, B_4 *and* B_5. Applying this notion, we determine the vertex points for (the last segment), segment $n-k+2$ as $B_{n-k+2}, B_{n-k+3}, B_{n-k+4}$ *and* B_{n-k+5}. To determine whether there is a change in convexity between any two segments we consider the positions of the other three vertex points with respect to the initial vertex point. For instance for segment 1, the positions B_2, B_3 *and* B_4 are investigated relative to B_1. This derivation can be summarized in Criterion II below.

Criterion II

An unwanted inflection point exists in a curve segment when the inflection point is not implied by the vertex points defining that segment and is characterized by a constant sign in the "relative gradients". When *Criterion II* is applied to the defining vertex points of a segment with inflection point, if there is a sign change in relative gradient within the segment, then the inflection point is 'implied'.

This can be explained by the change in convexity. For instance, for a cubic NURBS curve with five segments, the relative gradients in each curve segment are:

***Segment* 1**

$$\Delta_{11} = \frac{B_{y2} - B_{y1}}{B_{x2} - B_{x1}}$$

$$\Delta_{21} = \frac{B_{y3} - B_{y1}}{B_{x3} - B_{x1}}$$

$$\Delta_{31} = \frac{B_{y4} - B_{y1}}{B_{x4} - B_{x1}}$$

***Segment* 2**

$$\Delta_{12} = \frac{B_{y3} - B_{y2}}{B_{x3} - B_{x2}}$$

$$\Delta_{22} = \frac{B_{y4} - B_{y2}}{B_{x4} - B_{x2}}$$

$$\Delta_{32} = \frac{B_{y5} - B_{y2}}{B_{x5} - B_{x2}}$$

***Segment* 3**

$$\Delta_{13} = \frac{B_{y4} - B_{y3}}{B_{x4} - B_{x3}}$$

$$\Delta_{23} = \frac{B_{y5} - B_{y3}}{B_{x5} - B_{x3}}$$

$$\Delta_{33} = \frac{B_{y6} - B_{y3}}{B_{x6} - B_{x3}}$$

***Segment* 4**

$$\Delta_{14} = \frac{B_{y5} - B_{y4}}{B_{x5} - B_{x4}}$$

$$\Delta_{24} = \frac{B_{y6} - B_{y4}}{B_{x6} - B_{x4}}$$

$$\Delta_{34} = \frac{B_{y7} - B_{y4}}{B_{x7} - B_{x4}}$$

***Segment* 5**

$$\Delta_{15} = \frac{B_{y6} - B_{y5}}{B_{x6} - B_{x5}}$$

$$\Delta_{25} = \frac{B_{y7} - B_{y5}}{B_{x7} - B_{x5}}$$

$$\Delta_{35} = \frac{B_{y8} - B_{y5}}{B_{x8} - B_{x5}}$$

If there is a sign change in relative gradients in any of the segments, then the inflection point (if it exists in that segment, as identified by *Criterion I*) is desired. *Class II defects* in the curve can be identified if we investigate the behaviour of the curvature change with respect to arc length of the curve. In *Class II curve defects*; we have unwanted features such as cusps, loops, and flat regions.

Cusps

These are discontinuities in the curve. A discontinuity in the curve is characterised by a sharp change in the curvature. This can be identified by investigating the *change in curvature with respect to a corresponding change in arc length.*

Loops

These are characterised by a monotone bending of the curve to one side. In other words a curve with monotone convexity or monotone concavity has the tendency of forming a loop.

Flat regions

Flat regions in the curve are characterised by small change in curvature with respect to a large change in arc length.

To investigate a curve for these features we compute the change in curvature with respect to a corresponding change in arc length. This computation has to be investigated for knots x_k to x_{n+2}.

The curvature change with respect to a corresponding change in arc length is given by the expression:

$$\kappa_p(t) = \frac{d\,\kappa(t)}{d\,c(t)} \tag{3.6}$$

$$\kappa_p(t) = \frac{d\,\kappa(t)}{dt} / \frac{d\,c(t)}{dt}$$

$$\therefore \kappa_p(t) = \frac{\dot{\kappa}(t)}{\dot{c}(t)}$$

Where $\dot{\kappa}(t)$ *and* $\dot{c}(t)$ are first differentials of $\kappa(t)$ *and* $c(t)$ respectively with respect to the curve parameter, t.

For a better approximation of the quantity $\dot{c}(t)$ over the range of the entire curve we take its norm, hence [20]:

$$\kappa_p(t) = \frac{\dot{\kappa}(t)}{\left\| \dot{c}(t) \right\|} \qquad (3.7)$$

Equation (3.7) can be used to investigate the curve for cusps, loops and flat regions.

Basically, when $\left| \kappa_p(t) \right|$ is very large, it means $\left| \dot{\kappa}(t) \right| >> \left\| \dot{c}(t) \right\|$ and the corresponding curve segment will give a bulging or looping tendency. However, if $\left| \kappa_p(t) \right|$ is very small, then $\left| \dot{\kappa}(t) \right| << \left\| \dot{c}(t) \right\|$ may likely result, leading to the formation of flat regions. Cusps represent sharp changes in the curvature plot. To identify cusps in the curve, we investigate the term $\kappa_p(t)$ for two adjoining knots. The key idea here is that the curvature distribution should be made more even from one knot to the other. This can be achieved by taking the difference of $\kappa_p(t)$ for two adjoining knots, and minimizing this difference. For any arbitrary knot, x_i, the difference in curvature distribution leftwards and rightwards can be minimized by taking the modulus of the difference between $\kappa_p(x_{i-1})$ *and* $\kappa_p(x_{i+1})$. This quantity, if minimized can be used to ensure local shape correction, hence the local fairness indicator [20]:

$$L_i = \left| \Delta \kappa_p(t) \right| \qquad (3.8)$$

Equation (3.8) is the general form of the following three equations:

$$L_i = \left| \kappa_p(x_{i+1}) \right|, \quad \forall \quad i = k \qquad (3.9\text{ a})$$

$$L_i = \left| \kappa_p(x_{i-1}) \right|, \quad \forall \quad i = n+2 \qquad (3.9\text{ b})$$

$$L_i = \left| \kappa_p(x_{i+1}) - \kappa_p(x_{i-1}) \right|, \forall\; k+1 \le i \le n+1 \qquad (3.9\text{ c})$$

Equations (3.9b) and (3.9 c*)* are intuitively specified as stated above so as to exclude any modification (whenever it results) at the boundary knots, x_k and x_{n+2}. If we minimize the discrete summation of L_i (for all points with respect to the defective point), it is possible to have an approximately better curve. In this sense, the term could be assumed as a global fairness indicator, hence the definition [20]:

$$G = \sum_{i=k}^{n+2} L_i \qquad (3.10)$$

Equation (3.10) is a measure of the change in the curvature distribution of the curvature plot. The task here is obviously to minimize this change as this can be tantamount to a modification of the global shape of the curve. We therefore adopt a suitable nonlinear optimization technique [24] and [2] which is capable of searching for the appropriate knot and weight that could return a smoother curve.

3.2 Nonlinear Optimization Process

The optimization process [24] and [2] finds the best knot and weight values at the defective curve segment using the quality measures discussed in Section 3.1. The curve fairing problem is a highly nonlinear problem and can best be attacked by a nonlinear programming technique. The first step to solving this problem using a suitable nonlinear program technique is to properly formulate the problem as a nonlinear program.

In formulating the curve fairing problem as a non-linear program to make suitable for particular application, we notice that the problem has three parts:

Objective Function

$Min\ F(X) = f(x_1, x_2, \ldots, x_n)$

The objective function is the global fairness measure G defined earlier by equation (3.10).

Design Variables

$X(x_1, x_2, \ldots, x_n)$

There are only two variables in this problem; the knot, x_j and the corresponding weight, h_{j-k+2}. These are the pre-defined values at the defective curve segment.

Constraints

$g_j(X) \geq 0, \quad j = 1, 2, \ldots, n$

The constraints are (a) a non-decreasing knot vector with fixed end values, and (b) a homogeneous coordinate vector with fixed end values. The key idea in this NLP technique is to evaluate the objective function at a series of points which form a carefully directed search across the feasible region (the pool of possible values, based on the step length, that satisfy the above constraints). The search procedure is described in terms of base points and temporary positions (which represent original and perturbed values of x_j and h_{j-k+2} respectively). We start at some initial feasible points known as the first base point, denoted by: $X = B^{(0)} = (b_1^{(0)}, b_2^{(0)}, \ldots, b_n^{(0)})$

A step length δ_j (which is the j^{th} component of the vector D_j, with all other components zero) is chosen for each variable x_j. we now vary each variable in turn by amounts $+\delta_j$ or $-\delta_j$ each time accepting the change if it leads to improvement. Having varied each variable we reach a new base point $B^{(1)}$. Evaluate $F(B^{(0)})$, then vary the variable x_j and then, evaluate $F(B^{(0)} + D_1)$. If $F(B^{(0)} + D_1) < F(B^{(0)})$, then the point $(B^{(0)} + D_1)$ is called the temporary position and we denote it by $T_1^{(0)}$. If $F(B^{(0)} + D_1) \geq F(B^{(0)})$ then we evaluate $F(B^{(0)} - D_1)$ and if $F(B^{(0)} - D_1) < F(B^{(0)})$ then $(B^{(0)} - D_1)$ is the temporary position. If this also gives no improvement then $B^{(0)}$ is designated the temporary position. Next, we vary the variable x_2 about the temporary position $T_1^{(0)}$ instead of the original base point $B^{(0)}$, and $T_2^{(0)}$ is computed as the new temporary position as $T_0^{(1)} = 2B^{(1)} - B^{(0)}$. We then repeat the procedure for computing $T_j^{(0)}$ with the superscript zero replaced by 1. If the final temporary position is an improvement on the objective function value at $B^{(1)}$, this is established as the new base point, $B^{(2)}$. Then we compute a new temporary position $T_0^{(2)} = 2B^{(2)} - B^{(1)}$ and carry out exploratory moves around $T_0^{(2)}$. However, if this action leads to $F(T_n^{(1)}) \geq F(B^{(1)})$ we abandon the pattern search and continue with a sequence of exploratory moves about $B^{(2)}$.

3.3 Fairing methodology

In this section, we present the general methodology for fairing a typical NURBS curve. Two basic and very important conditions which the final curve must satisfy are:

(i) Shape preservation,

(ii) Performance characteristics.

To ensure that these conditions be satisfied, all curves must be faired based on the quality measures discussed in Section 3.2 above.

This paper features a fairing methodology that is based on concepts from two fields: Boolean arithmetic and set theory. Defects are faired according to their correction index, Γ. The correction index is set based on boolean arithmetic in a sense that only the selected defective point to be faired at a particular instance is set at $\Gamma = 1$ and all others set at $\Gamma = 0$. After correction, the previous point (corrected point) is set at $\Gamma = 0$ alongside others except the immediate next point in the rank order (which is now set at $\Gamma = 1$). *Class I* defects carry a higher rank than Class II defects and as such are faired before the latter, Q is a set of m number of *Class I* defects defined by: $Q = \{q_1, q_2, \ldots, q_m\}$ and ranked in the following decreasing sequence: $q_1 > q_2 > \ldots > q_m$ where $m \leq n - k + 2$. R is a set of *Class II* defects (*defects are lumped*) defined by: $R = \{r_1\}$. The universal set of defects existing in the curve is given by the expression: $\xi = Q \cup R.$. From the above, it follows that

$n(Q) =$ number of *Class I* defects identified

$n(R) =$ number of *Class II* defects identified

$n(\xi) =$ total number of defects identified

Assuming that $n(R) = 0$ is always false, the following are true:

If $n(\xi) > n(R)$, then $n(\xi)$ - $n(R)$ number of *Class I* defect(s) exist(s).

If however $n(\xi) = n(R)$, there exist no *Class I* defect.

Based on the above argument, the following steps describe the fairing methodology.

***Step* 1:** If $Q \neq 0$, (else go to *Step 6*) and $m > 1$; $\Gamma_{q1} = 1$; $\Gamma_q = 0, \ \forall \ q \in Q - \{q_1\}$; $\Gamma_r = 0, \ \forall \ r \in R$

***Step* 2:** Minimize G with respect to q_1.

***Step* 3:** Repeat *Step 1*, by setting $\Gamma_{q2} = 1; \Gamma_q = 0, \forall \ q \in Q - \{q_2\}; \Gamma_r = 0, \forall \ r \in R.$

***Step* 4:** Minimize G with respect to q_2.

***Step* 5:** Perform this operation till $\Gamma_{q_m} = 1$ and afterwards go to *Step 6*.

***Step* 6:** If $\xi = R$, then $Q = \varnothing$, hence set $\Gamma_r = 1, \forall \ r \in R$ and $\Gamma_q = 0, \forall \ q \in Q$.

***Step* 7:** Minimize G with respect to r_1

***Step* 8:** Repeat *Step 6,* and *Step 7* subsequently, until the algorithm converges.

4.0 Implementation

In this work a software has been developed specially (using Microsoft) to implement the proposed curve fairing scheme. To use the software, the designer inputs the order, k of the curve, the number of defining control points, $(n+1)$, the control points, $B_i = [B_{xi} \quad B_{yi}]$ and the homogeneous coordinate vector, [H]. The program then automatically computes local variables such as the knot vector, [X], the B-spline basis functions, $N_i^k(t)$ and the rational B-spline basis functions, $R_i^k(t)$. Further, the program proceeds to compute the parametric NURBS curve and uses values of the parametric curve to plot the initial curve. The program thereafter computes the third derivatives of the rational B-spline basis functions and uses them to compute the curvature values (which are in turn used to generate a curvature plot). The program then uses *Criteria I and II* in conjunction with the global fairness indicator to identify all defects in the curve. The variables to be optimized are then recommended for optimization by a nonlinear optimization method [24] and [2]. The optimization scheme finds the best knot that could give a minimum value of the objective function. The values are automatically used to plot the final, fair curve. Finally the software displays the initial curve, curvature plot and final (optimum) curve on its form, and also computes and displays on the form, the number of iterations, computer time (for the optimization process) and the percentage improvement on the curve.

4.1 Examples

***Example* 1**

The following control points define a fourth level curve which represents the feature line of the right arm of a chair.

$$B_1 = 1.5, \ 0$$
$$B_2 = 0, \ 2$$
$$B_3 = 1, \ 5$$
$$B_4 = 4, \ 6$$
$$B_5 = 9, \ 4$$
$$B_6 = 12, \ 14$$

In realizing his design objective, the designer specifies the following homogenous coordinate vector based on intuition [H] = [1, 0.5 0.75 0.75 0.25 1]. Investigate his design process using the proposed scheme.

***Example* 2**

A shoe stylist has to design a shoe by modelling the feature line of the front upper section of a sample shoe. He concluded that he could realize his objective by specifying the defining control points and inputting a homogenous coordinate vector. If he specifies the following control points:

$$B_1 = 0, 0$$
$$B_2 = 0, 2$$
$$B_3 = 2, 2$$
$$B_4 = 4, 2$$
$$B_5 = 6, 3$$
$$B_6 = 7, 5$$

and homogenous coordinate vector

[H] = [1 0.71 1 1 1 1].

Investigate his approach using the proposed shape control scheme.

***Example* 3**

A SCARA robot palletizer is to be installed in the BBH of an existing beer brewing facility. The firm has decided that the existing installation in the BBH be unaltered due to the huge costs that may be involved. However for the robot end effector to transfer objects from its station frame (the roller conveyor) to its goal frame (the pallet), a path defined by the following control points has to be accurately followed.

$$B_1 = 2.5, 0$$
$$B_2 = 0, 1$$
$$B_3 = 0, 4$$
$$B_4 = 4, 4$$
$$B_5 = 5, 6.5$$
$$B_6 = 2.5, 8.5$$

If a homogenous coordinate vector

[H] = [1 1 1 0.95 1 1]

is intuitively specified by the designer; realize the following objectives using the proposed shape control scheme: (a) accurate path following, and (b) minimum discontinuities along the travel path of the robot end effector.

4.2 Results

The results of the curve fairing solution to the 3 Example are shown in Figures 1-3.

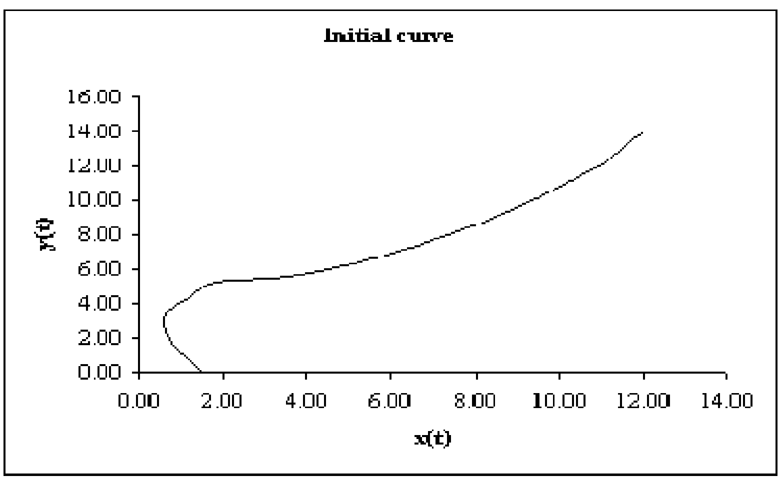

Figure 1a: Initial Curve of Y(T) against X(T) for Example 1(Chair right arm)

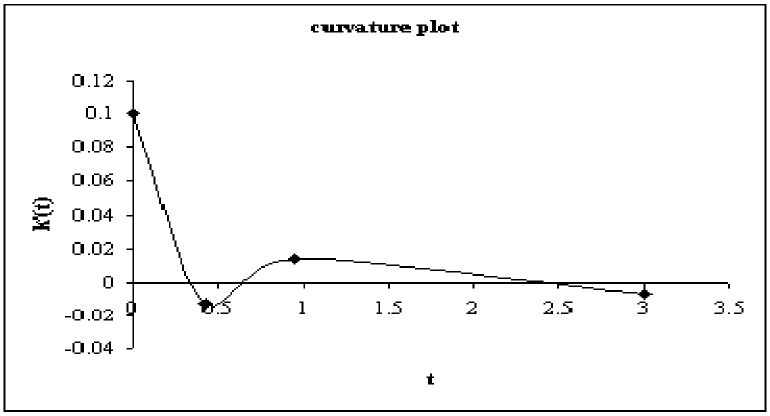

Figure 1b: Curvature Plot of K'(T) against T for Example 1(Chair right arm)

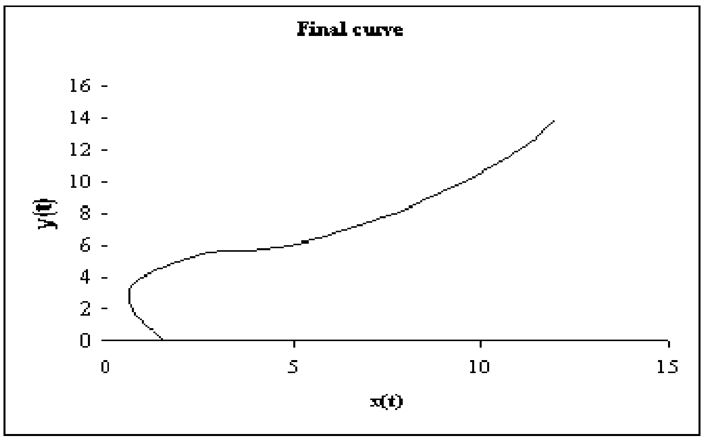

Figure 1c: Final Curve of Y(T) against X(T) for Example 1(Chair right arm)

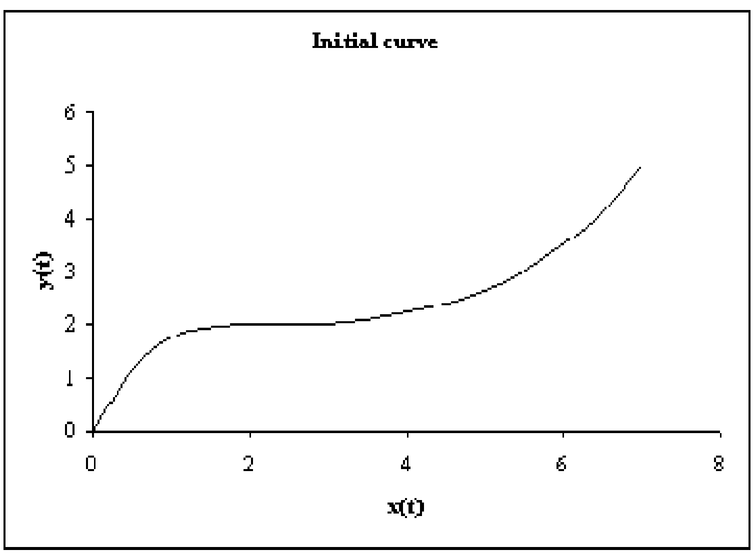

Figure 2a: Initial Curve of Y(T) against X(T) for Example 2 (Shoe upper section)