The Control Vector Scheme for Design of Planar Primitive PH curves

Ching-Shoei Chiang, Sheng-Hsin Tsai, and James Chen

Abstract—The PH curve can be constructed by given parameters, but the shape of the curve is not so easy to image from the value of the parameters. On the contract, Bézier curve can be constructed by the control polygon, and from the control polygon, we can image the figure of the curve. In this paper, we want to use the hodograph of Bézier curve to construct PH curve by selecting part of the control vectors, and produce other control vectors, so the property of PH curve exists.

Keywords—PH curve, hodograph, Bézier curve.

I. INTRODUCTION

THE curves and surfaces in computer-aided geometric design can be represented and designed in various ways. The curves and surfaces can be represented in parametric form, or in implicit form. Some of the curve, such as Bézier curve, can be represented by their control points. With smart graphic user interface, the design of the curve can be user friendly with pointer device, such as mouse, so that the control points and the curves can be display animatedly.

The hodograph of a plane parametric curve $r(t)=(x(t),y(t))$ is the locus described by the first derivative $r'(t)=(x'(t),y'(t))$ of the curve. A polynomial parametric curve is said to have a Pythagorean hodograph if there exists a polynomial $\sigma(t)$ such that $x'^2(t)+y'^2(t)=\sigma^2(t)$[1]. With this property, the polynomial PH curve has rational offset [2,3,4,5,6].

The polynomial PH curve can be represented in Bernstein-Bézier form, so the control points of the polynomial PH curve can be displayed. However, we would like to preview the appearance of the curve before we design it. We know the polynomial PH curve must be a Bézier curve, and not vice versa. In this paper, we propose 3 algorithms to find polynomial PH curves by giving part of the control vectors of the hodographs, and analysis the number of the PH curves we find in each algorithms.

C. S. Chiang is a professor in the Department of Computer and Information Science, Soochow University, Taiwan (phone: 011-886-2-23111531 ext. 2230; e-mail: chiang@cis.scu.edu.tw).
S. H. Tsai is a graduate student in the Department of Computer and Information Science, Soochow University, Taiwan.
James C. Chen is an Associate Professor in the Department of Industrial Engineering at Chung Yuan University, Taiwan.

There are 5 sections in this paper. The first section gives the introduction of the polynomial PH curve and the contents of this paper. The second section defines the notation and Theorem used in this paper. The third section explains the degree of freedom analysis, gives more Theorems, and proposes 3 algorithms, the fourth section gives experiment results, and the final section concludes the result.

II. DEFINITION AND NOTATION

We follow the definitions and Theorems in the reference [1], and list below:

Definition 1[1]:
The hodograph $r'(t)=(x'(t),y'(t))$ of a planar PH curve satisfy $x'(t)+y'(t)=\sigma(t)$, where $\sigma(t)$ is a polynomial.

Theorem 1(Kubota[7,1])
Three real polynomials $a(t)$, $b(t)$, and $c(t)$, where $\max[\deg(a), \deg(b)]=\deg(c)>0$, satisfy the Pythagorean condition $a^2(t)+b^2(t)=c^2(t)$ if and only if they can be expressed in terms of real polynomials $u(t)$, $v(t)$, and $w(t)$ in the form:

$a(t)=w(t)[u^2(t)-v^2(t)]$
$b(t)=2w(t)u(t)v(t)$
$c(t)=w(t)[u^2(t)+v^2(t)]$

Theorem 2[1]: The polynomial curve corresponding to the hodograph $r'(t)=(x'(t),y'(t))=(w(t)[u^2(t)-v^2(t)],2w(t)u(t)v(t))$, is of degree $n=2\mu+\lambda+1$, where $\lambda=\deg(w)$ and $\mu=\max(\deg(u),\deg(v))$. Taking $w(t)=1$ and $\gcd(u(t),v(t))=1$ gives a primitive polynomial PH curve.

We are concerned primitive polynomial PH curves in this paper and denote the primitive polynomial PH curve as PPH curves.

Let $r(t)=(x(t),y(t))$ be a PPH curve whose control points are $p_k=(x_k,y_k)$, $k=0,1,\dots,2\mu+1$, and whose hodograph has control vectors $\Delta p_k=p_{k+1}-p_k$, $k=0,1,\dots,2\mu$. This PPH curve is constructed by two polynomials $u(t)$ and $v(t)$ given in Bernstein-Bézier form as: $u(t)=\sum_{i=0}^{\mu}u_iB_i^{\mu}(t)$ and $v(t)=\sum_{i=0}^{\mu}v_iB_i^{\mu}(t)$, where $B_i^{\mu}(t)=C_i^{\mu}t^i(1-t)^{\mu-i}$, $C_i^{\mu}=\dfrac{\mu!}{i!(\mu-i)!}$.

We denote $r_i=[u_i,v_i]$, $N(i,j)=(I_{i,j},J_{i,j})=\left(u_iu_j-v_iv_j,u_iv_j+u_jv_i\right)$, $\varphi_\mu(k,i)=\dfrac{C_i^\mu C_{k-i}^\mu}{C_k^{2\mu}}$, then we have the following Theorem:

Theorem 3: The PPH curve constructed by two polynomials $u(t)$ and $v(t)$ given in Bernstein-Bézier form as: $u(t)=\sum_{i=0}^{\mu}u_iB_i^\mu(t)$ and $v(t)=\sum_{i=0}^{\mu}v_iB_i^\mu(t)$, then the relationship between r_i and control vectors of the hodograph is:

$$(2\mu+1)\Delta p_k=\sum_{i=0}^{k}\varphi_\mu(k,i)\,(I_{i,k-i},J_{i,k-i}),\ 0\le k\le 2\mu \qquad (1)$$

Proof:

From the equation $B_i^m(t)B_j^n(t)=\dfrac{C_i^m C_j^n}{C_{i+j}^{m+n}}B_{i+j}^{m+n}(t)$, we derive:

$$u^2(t)=\left(\sum_{i=0}^{\mu}u_iB_i^n(t)\right)\left(\sum_{j=0}^{\mu}u_jB_j^n(t)\right)$$

$$=\sum_{i=0}^{\mu}\sum_{j=0}^{\mu}u_iu_j\frac{C_i^\mu C_j^\mu}{C_{i+j}^{2\mu}}B_{i+j}^{2\mu}(t)$$

$$=\sum_{k=0}^{2\mu}\left(\sum_{i=0}^{k}u_iu_{k-i}\frac{C_i^\mu C_{k-i}^\mu}{C_k^{2\mu}}\right)B_k^{2\mu}(t)$$

From the definition of the hodograph of PPH curve, $r'(t)=(x'(t),y'(t))=(u^2(t)-v^2(t),2u(t)v(t))$, we have

$$x'(t)=u^2(t)-v^2(t)=\sum_{k=0}^{2\mu}\left(\sum_{i=0}^{k}\left(u_iu_{k-i}-v_iv_{k-i}\right)\frac{C_i^\mu C_{k-i}^\mu}{C_k^{2\mu}}\right)B_k^{2\mu}(t)$$

With similar approach, we derive:

$$y'(t)=2u(t)v(t)=\sum_{k=0}^{2\mu}\left(\sum_{i=0}^{k}\left(u_iv_{k-i}+u_{k-i}v_i\right)\frac{C_i^\mu C_{k-i}^\mu}{C_k^{2\mu}}\right)B_k^{2\mu}(t).$$

Compare with the property that $r'(t)$ has the control vectors at $(2\mu+1)\Delta p_k$, $k=0,1,\ldots,2\mu$, we prove this Theorem.

♦

Notice that $\varphi_\mu(k,i)=\varphi_\mu(k,k-i)$, $N(i,j)=N(j,i)$, so we can simplify the Equation (1) into:

$$(2\mu+1)\Delta p_k$$
$$=\begin{cases} 2\sum_{i=0}^{k/2}\varphi_\mu(k,i)\,N(i,k-i)+\varphi_\mu(k,k/2)N(k/2,k/2) & \text{if } k \text{ is even}\\[2mm] 2\sum_{i=0}^{(k-1)/2}\varphi_\mu(k,i)\,N(i,k-i) & \text{if } k \text{ is odd}. \end{cases}$$
$$(2)$$

From Theorem 1, the relation between $r_i=[u_i,v_i]$ and Δp_k can be shown. For example, when $\mu=3$, we have

$$C(t)=7\begin{bmatrix}\Delta p_0\\ \Delta p_1\\ \Delta p_2\\ \Delta p_3\\ \Delta p_4\\ \Delta p_5\\ \Delta p_6\end{bmatrix}\begin{bmatrix}(u_0^2-v_0^2,2u_0v_0)\\ (u_0u_1-v_0v_1,u_0v_1+u_1v_0)\\ (\tfrac{2}{5}(u_0u_2-v_0v_2)+\tfrac{3}{5}(u_1^2-v_1^2),\tfrac{2}{5}(u_0v_2+u_2v_0)+\tfrac{3}{5}(u_1v_1))\\ (\tfrac{1}{10}(u_0u_3-v_0v_3)+\tfrac{9}{10}(u_1u_2-v_1v_2),\tfrac{1}{10}(u_0v_3+u_3v_0)+\tfrac{9}{10}(u_1v_2+u_2v_1)))\\ (\tfrac{2}{5}(u_1u_3-v_1v_3)+\tfrac{3}{5}(u_2^2-v_2^2),\tfrac{2}{5}(u_1v_3+u_3v_1)+\tfrac{3}{5}(u_2v_2))\\ u_2u_3-v_2v_3,u_2v_3+u_3v_2)\\ (u_3^2-v_3^2,2u_3v_3)\end{bmatrix}\ (3)$$

The control vectors can be found directly by $u(t)$ and $v(t)$. It is not easy to predict the geometric variation of the PH curve through $u(t)$ and $v(t)$. If we can construct the PH curve from the control vectors or the hodograph of the curve, we can establish a smart graphic user interface, request for control points or control vector for the PPH curve, and find one or more sets of $r_i=[u_i,v_i]$, $i=0,1,..,\mu$, it is helpful for us to design a PH curve. The problems are: (1) How many control points (of the hodograph) we need to construct a PH curve? (2) How to find at least one set of $r_i=[u_i,v_i]$, $i=0,1,..,\mu$, satisfied the requirement of PH curve.

III. CONVERSION FROM PPH CURVE TO BÉZIER CURVE

We analyze the degree of the freedom of the PPH curve first, and then give an algorithm to construct the associated PH curve in this section.

A Analysis of the Degree of Freedom

We would like to know how many control vectors we need to construct the PPH curve. Consider equation (1), there are $6\mu+4$ variables in the system of equations, including $r_i=[u_i,v_i]$, $0\le i\le\mu$ and $\Delta p_k=(\Delta x_k,\Delta y_k)$, $0\le k\le 2\mu$. So the degree of freedom for the system of equations is $6\mu+4$. There are $2\mu+1$ equations for Δx_k, $0\le k\le 2\mu$, and the other $2\mu+1$ equations for Δy_k, $0\le k\le 2\mu$, so we have $4\mu+2$ constraints for this system of equations. Before we want find all solutions, we need to give values to $(6\mu+4)-(4\mu+2)=2\mu+2$ variables. The natural selection for these $2\mu+2$ variables is $r_i=[u_i,v_i]$, $0\le i\le\mu$, it convert the PPH curve into Bézier curve. Or, we can select $\mu+1$ control vectors among $\Delta p_k=(\Delta x_k,\Delta y_k)$, $0\le k\le 2\mu$, and find other control vectors with $r_i=[u_i,v_i]$, $0\le i\le\mu$.

B. Theorems for the Node Value

Theorem 4: The solutions for $u^2-v^2=a$, $2uv=c$, $(a,c)\ne(0,0)$ is:
$$(u,v)=\pm(0,\sqrt{-a})\ \text{if } a<0, c=0$$
$$=\pm(\sqrt{a},0)\ \text{if } a>0, c=0$$
$$=\pm\frac{1}{\sqrt{2}}\left(\frac{c}{\sqrt{\sqrt{a^2+c^2}-a}},\sqrt{\sqrt{a^2+c^2}-a}\right)\ \text{if } c\ne 0$$

Proof: Trivial.

Notice that we are working on the real number, so that there are two solutions for each case. These solutions are two couples not equal to $(0,0)$ with different sign. We denote this two solutions as (u,v) and $-(u,v)$. Notice that $(u,v)\ne(0,0)$. Notice further that when $a=0$, $u=v=c=0$. Both (u,v) and $-(u,v)$ always produce the same (a,c).

Theorem 5: Let $N(i,i)=(I_{i,i}, J_{i,i})=(a,c) \neq(0,0)$, $N(j,j)=(I_{j,j},J_{j,j})=(b,d)\neq(0,0)$, then $(I_{i,j}^2-J_{i,j}^2, 2I_{i,j}J_{i,j})=(ab-cd, ad+bc)\neq(0,0)$.

Proof: It is easy to prove $((I_{i,j})^2-(J_{i,j})^2, 2I_{i,j}J_{i,j})=(ab-cd, ad+bc)$ by carefully algebraic computation. Now, we need to prove $(ab-cd, ad+bc)\neq(0,0)$. Assume $(ab-cd, ad+bc)=(0,0)$. Let's treat b,d as variables, a,c as constants, the system of equation

has solution $(b,d)=(\begin{vmatrix} 0 & -c \\ 0 & a \end{vmatrix} \Big/ \begin{vmatrix} a & -c \\ c & a \end{vmatrix}, \begin{vmatrix} a & 0 \\ c & 0 \end{vmatrix} \Big/ \begin{vmatrix} a & -c \\ c & a \end{vmatrix}) =(0,0)$ if

$a^2+c^2\neq0$, which contradicts with $(b,d)\neq(0,0)$. If $a^2+c^2=0$, it contradicts $(a,c) \neq(0,0)$. So we proved $(ab-cd, ad+bc)\neq(0,0)$.
♦

Combine Theorem 4 and 5, we have the following corollary:

Corollary 1: Given $N(i,i)$ and $N(j,j)$, the vectors associated to the solutions of $N(i,j)$ are two couples with opposite sign.

Theorem 6: Let $N(i,i) =(a,c)\neq(0,0)$, $N(i,j)=(e,f) \neq(0,0)$, then $N(j,j)= \frac{1}{a^2+c^2} ((a,c)\cdot(e^2-f^2,2ef), (-c,a)\cdot(e^2-f^2,2ef))\neq(0,0)$. The notation $\cdot$ is the inner product of two vectors.

Proof: assume $N(j,j)=(b,d)$, from Theorem 5, we have $ab-cd=e^2-f^2$ and $cb+ad=2ef$, from these two equations with b,d unknown, we can easily find one solution for (b,d).

Now we need to prove $(b,d)\neq(0,0)$. That is, $((a,c)\cdot(x,y), (a,-c)\cdot(y,x))\neq(0,0)$ where $x= e^2-f^2$, $y=2ef$. Assume $(b,d)=(0,0)$, we need to solve $ax+cy=0$ and $-cx+ay=0$. Because $a^2+c^2\neq0$, the system of equations has solution $(x,y)=(e^2-f^2, 2ef)=(0,0)$. It implies $(e,f)=(0,0)$ and contradicts with $(e,f)\neq(0,0)$. So we proved $N(j,j)=(b,d)\neq(0,0)$.
♦

Because $N(i,j)=N(j,i)$, the Theorem can be easily modify as:

Corollary 2: Let $N(i,i) =(a,c)\neq(0,0)$, $N(j,i)=(e,f) \neq(0,0)$, then $N(j,j)=\frac{1}{a^2+c^2} ((a,c)\cdot(e^2-f^2,2ef), (-c,a)\cdot(e^2-f^2,2ef))\neq(0,0)$.

Let's give some graphs to denote the input/output node for our Theorems and corollaries. In corollary 1, we draw correspond graph in Fig. 1(a), using arrow to indicate input node $N(i,i)$ and $N(j,j)$ points to the output node $N(i,j)$. The node $N(i,j)$ in Fig. 1 is the output node for correspond Theorems and corollaries, and other nodes which has arrow point out are input nodes.

The graphs for Theorem 6 and corollary 2 are shown in Fig. 1(b) and (c) respectively.

Currently, we use two input nodes to find one output node. The following Theorems and corollaries use three input nodes to find one output nodes.

Theorem 7: Let $N(k,k)=(e,f) \neq(0,0)$, $N(k,i)=(a,c) \neq(0,0)$ and $N(k,j)=(b,d) \neq(0,0)$, then

$N(i,j)= \frac{1}{e^2+f^2}((e,f)\cdot(ab-cd,ad+bc), (-f,e)\cdot(ab-cd,ad+bc)\neq(0,0)$.

Proof: The proof is similar to Theorem 6.
♦

From the fact that $N(k,i)=N(i,k)$, $N(k,j)=N(j,k)$, more combination can be generated. The value for $N(i,j)$ can be generated from (1) $N(k,k)$, $N(i,k)$ and $N(j,k)$, (2) $N(k,k)$, $N(k,i)$ and $N(j,k)$, (3) $N(k,k)$, $N(i,k)$ and $N(k,j)$.

The graphs for the input/output nodes and number of solutions for Theorem 7 are shown in Fig. 1(d),(e),(f) respectively.
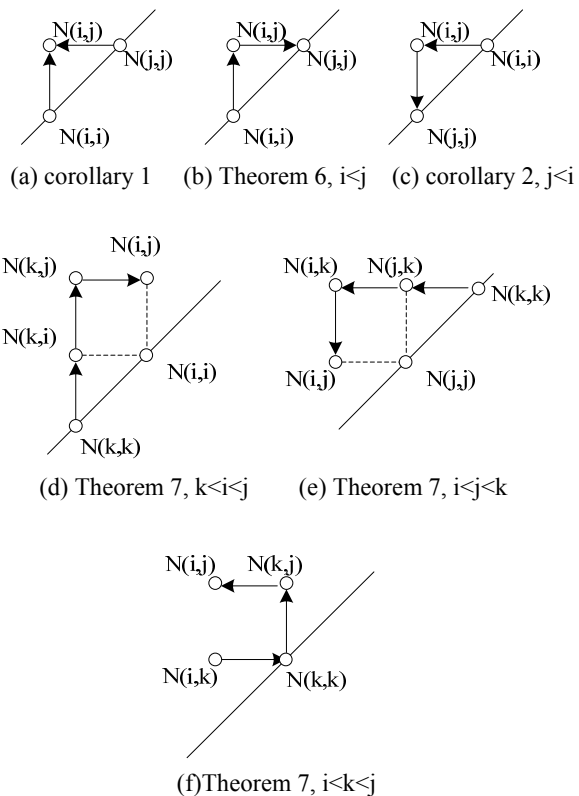

(a) corollary 1   (b) Theorem 6, i<j   (c) corollary 2, j<i

(d) Theorem 7, k<i<j   (e) Theorem 7, i<j<k

(f)Theorem 7, i<k<j
Fig. 1 I/O nodes for Theorems and corollaries

Notice that when k=i<j, Fig. 1(d) degenerate into Fig. 1(b), when i<j=k, Fig. 1(e) degenerate into Fig. 1(c).

Theorem 7 can be extended by finding one node from other 3 nodes. For example, we find $N(i,j)$ from $N(k,i)$, $N(k,j)$ and $N(k,k)$, as shown in Fig. 1(d). The other 3 cases are (1) finding $N(k,i)$ from $N(i,j)$, $N(k,j)$ and $N(k,k)$, (2) finding $N(k,j)$ from other 3 nodes, (3) finding $N(k,k)$ from other 3 nodes. So there are 4 cases associated to Fig. 1(d). The extension for Fig. 1(e) and Fig. 1(f) are similar. Under these extensions, there are 12 cases for finding one node from other 3 nodes assuming these nodes are on the left upper side of the diagonal. That is, these nodes $N(i,j)$ has the properties i≤j. we emphasis these 3 cases shown in Fig. 1(d)(e)(f) because they are the cases we used in our algorithms.

Now we start the algorithms for finding PPH curve from the control vectors of its hodographs.

C. Algorithms

Given $r_i=[u_i, v_i]$, where $0 \le i \le \mu$, we want to find the control vectors for associated Bézier Curve. The associated Bézier curve can be constructed from Equation (2) directly. We can find all $(I_{i,j}, J_{i,j})$, $0 \le i \le \mu$, $0 \le j \le \mu$ from $r_i=[u_i, v_i]$, and find the control vectors Δp_k, $0 \le k \le 2\mu$, from Equation (2). With the reference point $p_0=(x_0, y_0)$, we can draw the Bézier curve. Now we would like to construct the PH curve in the other way. It is not so easy to preview the curve from r_i, $0 \le i \le \mu$, so we would like to construct the PPH curve by indicating part of the control vectors, and calculate the others. As mentioned in section *A*, we need to know $\mu+1$ control vectors of the hodograph in order to find r_i, $0 \le i \le \mu$, so that other control vectors can be found through r_i and the control vectors we found.

Consider Equation (3) as an example. Given any four control vectors from Δp_k, $0 \le k \le 6$, we want to find other control vectors so that the curve can be drawn. Assume we know Δp_k, $0 \le k \le 3$, and want to find Δp_k, $4 \le k \le 6$, we can generate 8 degree 2 equations by the first 4 row of Equation 3, and there are 8 variables, $[u_k, v_k]$, $0 \le k \le 3$, among them. There are at most $2^8=256$ solutions in the system of equation. The algorithms we proposed here try to reduce the number of solution with more efficient way.

We illustrate three algorithms in this section. The first algorithm and second algorithm select the first $\mu+1$ control vectors Δp_k, $0 \le k \le \mu$, and find others. The first algorithm finds r_i, $0 \le i \le \mu$, first, and then finds $N(i,j)=(I_{i,j}, J_{i,j})= (u_i u_j - v_i v_j, u_i v_j + u_j v_i)$, the control vectors Δp_k, $0 \le k \le 2\mu$, is produced by Equation (2) finally. The second algorithm finds $N(i,j)$ directly, and then derives the other control vectors. The third algorithm selects both side of the control vectors, that is, Δp_k, $0 \le k \le \lfloor \frac{\mu}{2} \rfloor$, $\lfloor \frac{3\mu}{2} \rfloor+1 \le k \le 2\mu$, and find others. All of these three algorithms use the Equation (2).

We use Equation (3) to illustrate the idea of the first algorithm. In Equation (3), we can find two solutions for $r_0=[u_0, v_0]$ from Theorem 4 and the first row of Equation (3): $7(\Delta x_0, \Delta x_0)=(u_0^2-v_0^2, 2u_0v_0)$. We further solve $r_1=[u_1, v_1]$ by solving system of linear equations in the second row of Equation (3): $7(\Delta x_1, \Delta x_1)=(u_0u_1-v_0 v_1, u_0v_1+u_1v_0)$. Repeat the process on the third row and fourth row of Equation (3), we can find single solution for $[u_2, v_2]$ and $[u_3, v_3]$, associated with each solution of r_0. With all of the r_i, $i=0,1,2,3$, we can find Δp_k, $4 \le k \le 6$ directly from Equation (2). We totally find two sets of control vectors and produce at most two curves. However, these two sets of r_i produce the same curve. We list the idea in the following algorithm: (Let $P=\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, Q=\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$)

Algorithm 1 // Given Δp_k, $0 \le k \le \mu$, find Δp_k, $\mu+1 \le k \le 2\mu$
1. Find $r_0=[u_0, v_0]$ from the equation:

$$(2\mu+1)\Delta p_k=\left(r_0 \Pr_0^T, r_0 Q r_0^T\right)=(u_0^2-v_0^2, 2u_0v_0)$$

2. Select any solution for $r_0=[u_0, v_0]$.
3. For $k=1$ to μ
Find $r_k=[u_k, v_k]$ from the equation:

$$[u_k, v_k]=\frac{2\mu+1}{2\varphi_\mu(k,0)(u_0^2+v_0^2)}[\Delta x_k, \Delta y_k]\begin{bmatrix} u_0 & -v_0 \\ v_0 & u_0 \end{bmatrix}$$

$$-\frac{1}{2\varphi_\mu(k,0)(u_0^2+v_0^2)}\sum_{i=1}^{k-1}\varphi_\mu(k,i)\left[r_i \Pr_{k-i}^T, r_i Q r_{k-i}^T\right]\begin{bmatrix} u_0 & -v_0 \\ v_0 & u_0 \end{bmatrix}$$
(4)

4. For $k=\mu+1$ to 2μ
Find $\Delta p_k=(\Delta x_k, \Delta y_k)$ from the equation:

$$(\Delta x_k, \Delta y_k)=\frac{1}{2\mu+1}\sum_{i=0}^{k}\varphi_\mu(k,i)\left(r_i \Pr_{k-i}^T, r_i Q r_{k-i}^T\right)$$

The equation (4) can be derived from Equation (1) with $(I_{i,k-i}, J_{i,k-i})=\left(r_i \Pr_{k-i}^T, r_i Q r_{k-i}^T\right)$. We can simplify this equation into:

$$2\varphi_\mu(k,0)\left(r_0 \Pr_k^T, r_0 Q r_k^T\right)$$

$$=(2\mu+1)(\Delta x_k, \Delta y_k)-\sum_{i=1}^{k-1}\varphi_\mu(k,i)\left(r_i \Pr_{k-i}^T, r_i Q r_{k-i}^T\right)$$

So, we can solve $[u_k, v_k]$ via $[u_0, v_0]$, $[u_1, v_1]$,…, $[u_{k-1}, v_{k-1}]$.

How many different solutions for this algorithm? From Theorem 2, there are two solutions for $[u_0, v_0]$ in the first step of the algorithm, call them $[u_0, v_0]$ and $-[u_0, v_0]$. The steps (2) gives single solution on each $[u_0, v_0]$ and $-[u_0, v_0]$ for $[u_k, v_k]$, $1 \le k \le \mu$, and in the third step, we find unique control points $(\Delta x_k, \Delta y_k)$, $\mu+1 \le k \le 2\mu$ on each $[u_0, v_0]$ and $-[u_0, v_0]$. Observe Equation (4), we find two sets of solutions, each $[u_k, v_k]$ associated with the other solution $-[u_k, v_k]$.

We have two set of solutions, they are $[u_k, v_k]$, $0 \le k \le \mu$, and $-[u_k, v_k]$, $0 \le k \le \mu$. In step 3 of Algorithm 1, we found these two sets of solutions produce the same control vectors Δp_k, $\mu+1 \le k \le 2\mu$, because:

$$\left(r_i \Pr_{k-i}^T, r_i Q r_{k-i}^T\right)=\left((-r_i)P(-r_{k-i})^T, (-r_i)Q(-r_{k-i})^T\right)$$

So we conclude that the first algorithm produce single curve.

In Algorithm 1, we find all $r_i=[u_i, v_i]$, $0 \le i \le \mu$, so that all of the control vectors Δp_k, $\mu+1 \le k \le 2\mu$ can be found. In fact, we can find all $N(i,j)$ from Δp_k, $0 \le k \le \mu$ and then find Δp_k, $\mu+1 \le k \le 2\mu$ directly from $N(i,j)$ by the following algorithm:

Algorithm 2: // Given Δp_k, $0 \le k \le \mu$, find Δp_k, $\mu+1 \le k \le 2\mu$
1. Find $N(0,0)$ from Δp_0. // Equation (2)
2. For $j=1$ to μ do {
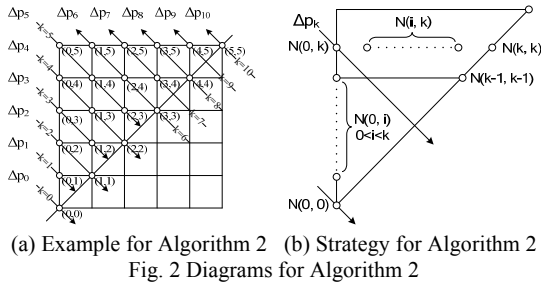 21 Find $N(0,j)$ from Δp_j // Equation (2)
  22 For $i=1$ to $j-1$
    Find $N(i,j)$ from $N(0,0)$, $N(0,i)$, $N(0,j)$ //Thm 7
 23 Find $N(j,j)$ from $N(0,0)$ and $N(0,j)$ } // Thm 6
3 For $k=\mu+1$ to 2μ do

$$\Delta p_k=\frac{1}{2\mu+1}\sum_{i=0}^{k}\varphi_\mu(k,i)N(i,k-i)$$

(a) Example for Algorithm 2   (b) Strategy for Algorithm 2
Fig. 2 Diagrams for Algorithm 2

Let's use an example to explain Algorithm 2. We draw Fig. 2(a) to illustrate the idea of Algorithm 2. In Fig. 2(b), the arrow from upper left to the lower right indicates the vector at the beginning of the arrow is an input vectors the arrow from lower right to the upper left indicates the vector at the end of the arrow is an input vectors. We select μ=5 as the example. In Algorithm 2, the step 1 find the initial node N(0,0) from Δp_0. In the first iteration of step 2 (j=1), we find N(0,1) from Δp_1 first (step 21), and skip the step 22 because the for loop has initial value i=1 to the value j-1=0, Now we know N(0,0) and N(0,1), we can find N(1,1) via Theorem 6. In the second iteration of step 2(j=2), we find N(0,2) from Equation (2) (step 21), and find N(1,2)(step 22) from N(0,0), N(0,1), N(0,2) via Theorem 7, and Find N(2,2)(step 23) from N(0,0) and N(0,2) via Theorem 6. Now we know all of the node N(i,j), where j≤2, i≤j. Repeat step 2, we can find all N(i,j).

Generally, in beginning iterations in step 21, we assume that the value of the node j≤k-1 is known, as marked the area (1) in the Fig. 2(b). With the new input value Δp_k, we can find the single value of new node N(0,k) from Equation (2). Notice that all of the nodes used in this step is known except N(0,k). We further find the single value of N(i,k), 1≤i≤k-1 from N(0,0), N(0,i) and N(0,k) by Theorem 7 in step 22, and find the single value for N(k,k) from N(0,0) and N(0,k) by Theorem 6 in step 23. Repeat this process, we can find all value of N(i,j), 1≤i≤μ,1≤j≤μ. After we find all node value, 0≤i≤μ,0≤j≤μ, other control vectors can be easily found in step 3 by Equation (2).

Notice that we find single value for all control vectors. That is, we construct single curve in this algorithm.
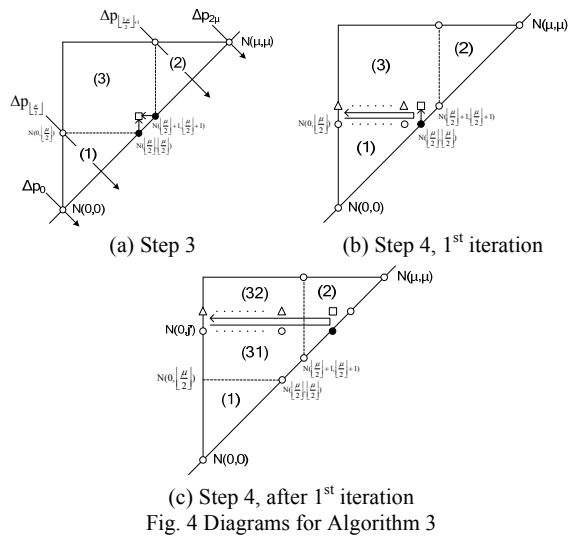

(a) algorithm 2   (b) algorithm 3
Fig. 3 I/O control vectors for algorithm 2 and 3
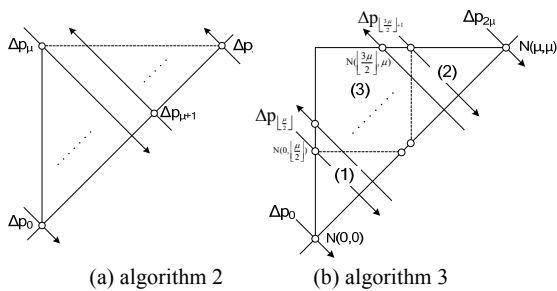
In the second algorithm, we try to find the last μ control vectors $(\Delta p_k, \mu+1 \le k \le 2\mu)$ from the first μ+1 control vectors$(\Delta p_k, 0 \le k \le \mu)$. We use Fig. 3(a) to represent the input and output of the control vectors for this algorithm. The disadvantage of the algorithm is that the final part of the curve

is not easy to predict. So, we change our approach to find the control vectors in the middle by giving two sides of the control vectors, which produce the third algorithm. Fig. 3(b) represents the input and output of the control polygon for the following algorithm:

Algorithm 3: Given the control victors Δp_k, $0 \le k \le \lfloor \frac{\mu}{2} \rfloor$ and $\lfloor \frac{3\mu}{2} \rfloor + 1 \le k \le 2\mu$, find the control vectors Δp_k, $\lfloor \frac{\mu}{2} \rfloor + 1 \le k \le \lfloor \frac{3\mu}{2} \rfloor$.

1. Find N(i,j), $0 \le i \le \lfloor \frac{\mu}{2} \rfloor$, $0 \le j \le \lfloor \frac{\mu}{2} \rfloor$, i≤j by Algorithm 1 using Δp_k, $k=0 \le k \le \lfloor \frac{\mu}{2} \rfloor$ as input vectors.

2. Find N(i,j), $\lfloor \frac{\mu}{2} \rfloor + 1 \le i \le \mu$, $\lfloor \frac{\mu}{2} \rfloor + 1 \le j \le \mu$, i≤j by modifying Algorithm 1 using Δp_k, $\lfloor \frac{3\mu}{2} \rfloor + 1 \le k \le 2\mu$ as input vectors.

3. Find N($\lfloor \frac{\mu}{2} \rfloor$,$\lfloor \frac{\mu}{2} \rfloor$+1) by Theorem 5.

4. For j=$\lfloor \frac{\mu}{2} \rfloor$+1 to μ

    For i=0 to $\lfloor \frac{\mu}{2} \rfloor$

        Find N(i,j) by Theorem 7 using N(i,j-1), N(j-1,j-1), and N(j-1,j) as input.   // Theorem 7

5. For k=$\lfloor \frac{\mu}{2} \rfloor$+1 to $\lfloor \frac{3\mu}{2} \rfloor$, find Δp_k.   // Equation (2).

(a) Step 3   (b) Step 4, 1st iteration

(c) Step 4, after 1st iteration
Fig. 4 Diagrams for Algorithm 3

In algorithm 3, the first step find N(i,j), $0 \le i \le \lfloor \frac{\mu}{2} \rfloor$, $0 \le j \le \lfloor \frac{\mu}{2} \rfloor$, i≤j, marked in area (1) in Fig. 4(a), using the idea in algorithm 1. With simple modification, we can easily find N(i,j), $\lfloor \frac{\mu}{2} \rfloor + 1 \le i \le \mu$, $\lfloor \frac{\mu}{2} \rfloor + 1 \le j \le \mu$, i≤j, marked in area (2) in Fig. 3(a). These two nodes N($\lfloor \frac{\mu}{2} \rfloor$,$\lfloor \frac{\mu}{2} \rfloor$), and N($\lfloor \frac{\mu}{2} \rfloor$+1,$\lfloor \frac{\mu}{2} \rfloor$+1), as mark solid circle in Fig. 4(a), are find at the first and second steps respectively. The value of nodes we did not find yet is in area (3). Notice that all N(i,i), 0≤i≤μ is found already. In this situation, all N(i,j) can be derived by corollary 1. However, we find 2 solutions in corollary 1 for all nodes in area (3),

which makes the final Δp_k, $\left\lfloor\frac{\mu}{2}\right\rfloor+1\leq k\leq\left\lfloor\frac{3\mu}{2}\right\rfloor$, are hard to select, because the selection of correct value for each node is hard. So, we try to use corollary 1 as few as possible.

In step 3, we find the node $N(\left\lfloor\frac{\mu}{2}\right\rfloor,\left\lfloor\frac{\mu}{2}\right\rfloor+1)$ in the lower right corner of area (3) by using corollary 1, as marked square in Fig. 4(a), which produce two couples with different sign. Let's select one value for $N(\left\lfloor\frac{\mu}{2}\right\rfloor,\left\lfloor\frac{\mu}{2}\right\rfloor+1)$, and continue the algorithm, consider the first iteration of step 4($j=\left\lfloor\frac{\mu}{2}\right\rfloor+1$), the node $N(i,\left\lfloor\frac{\mu}{2}\right\rfloor+1)$, $0\leq i\leq\left\lfloor\frac{\mu}{2}\right\rfloor$, as marked triangle in Fig. 4(b), can be found by Theorem 7, using $N(i,\left\lfloor\frac{\mu}{2}\right\rfloor)$, $0\leq i\leq\left\lfloor\frac{\mu}{2}\right\rfloor$, $N(\left\lfloor\frac{\mu}{2}\right\rfloor,\left\lfloor\frac{\mu}{2}\right\rfloor)$(solid circle), and $N(\left\lfloor\frac{\mu}{2}\right\rfloor,\left\lfloor\frac{\mu}{2}\right\rfloor+1)$(square) as input. Notice that each node produce single value for each value of $N(\left\lfloor\frac{\mu}{2}\right\rfloor,\left\lfloor\frac{\mu}{2}\right\rfloor+1)$ we found in step 3. There are two sets of values for nodes in area (3).

Start from the second iteration ($j>\left\lfloor\frac{\mu}{2}\right\rfloor+1$) of step 4, assume all nodes $N(i,j)$, $0\leq i\leq\left\lfloor\frac{\mu}{2}\right\rfloor, 0\leq j<j'$, shown in the area (31) in Fig. 4(c), produce single value if we select one value for $N(\left\lfloor\frac{\mu}{2}\right\rfloor,\left\lfloor\frac{\mu}{2}\right\rfloor+1)$, we want to produce the value $N(i,j')$, $0\leq i\leq\left\lfloor\frac{\mu}{2}\right\rfloor$ by Theorem 7, using $N(i,j'-1)$, $N(j'-1,j'-1)$, $N(j'-1,j')$ as input. Notice that $N(i,j'-1)$ in area (31) has single value, $N(j'-1,j'-1)$ on the diagonal has single value, and $N(j'-1,j')$ in area (2) also has single value. From the observation, we know $N(i,j)$, $0\leq i\leq\left\lfloor\frac{\mu}{2}\right\rfloor, 0\leq j\leq j'$ has single value. At the end of iteration, we find single value for all nodes, which produce single control vectors, which produce single curve.

With the second value in $N(\left\lfloor\frac{\mu}{2}\right\rfloor,\left\lfloor\frac{\mu}{2}\right\rfloor+1)$, we can find the second set of values for all node in area (3) in Fig. 4(a), which produce the second curve. So, this algorithm produces at most 2 curves.

All 3 algorithm3 takes $O(\mu)$ times, where μ is the maximum degree of $u(t)$ and $v(t)$.

IV. EXPERIMENTAL RESULT

We implement all three algorithms using Maple 10. These programs works on PC with Pentium D 2.8GHz CPU and 1GB RAM. These examples in 3 algorithms takes very shout time to compute output vectors (about 0.015 seconds for each algorithm).

Example 1:
Given the control points $p_0=(0,0)$, $p_1=(1,2)$, $p_2=(2,3.6)$, $p_3=(3,4.7)$, $p_4=(4,5.3)$, as shown in Fig. 5(a), both the algorithm 1 and algorithm 2 produce the same curve, as shown in Fig. 5(b). In algorithm 1, we find the value $(u_0,v_0)=(3.365,2.080),(u_1,v_1)=(2.993,1.478),(u_2,v_2)=(2.372,0.3 10)and(u_3,v_3)=(2.478,-0.911)$. Both algorithms find the value for $\Delta p_4=[0.975,0.180]$, $\Delta p_5=[0.881,-0.199]$ and $\Delta p_6=[0.759,-$

0.645]. This result produces $p_5=(4.975,5.480)$, $p_6=(5.855,5.281)$, $p_7=(6.614,4.636)$. The output curve is shown in Fig. 5(b).
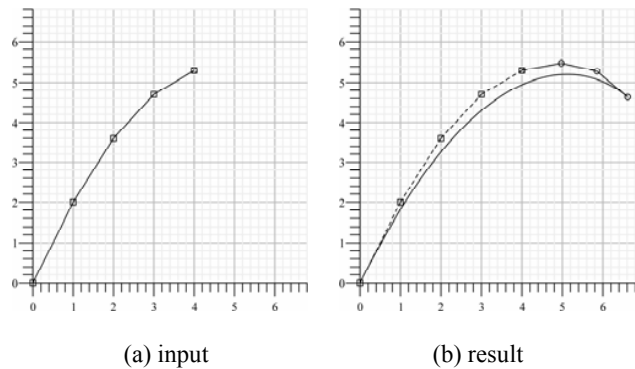

(a) input   (b) result

Fig. 5 I/O vectors and result curves for example 1

Example 2:
Given the control vectors $\Delta p_0=[1,2]$, $\Delta p_1=[1,1.2]$ with reference point at the origin, so that the control points $p_0=(0,0)$, $p_1=(1,2)$, $p_2=(2,3.2)$, the front part of the control polygon is shown in Fig. 6(a). Given the control vectors $\Delta p_5=[-1.2,-1]$, $\Delta p_6=[-1.2,-0.5]$, with a reference point p_5 at the origin, the end part of the control polygon is shown in Fig. 6(b). With algorithm 3, we find two solutions. The first solution is $\Delta p_2=[-0.188,0.801]$, $\Delta p_3=[-1.058,1.004]$, and $\Delta p_4=[-0.850,-0.531]$, and its associated curve is shown in Fig. 6(c). The second solution is $\Delta p_2=[1.234,-0.013]$, $\Delta p_3=[1.058,-1.004]$, and $\Delta p_4=[-0.377,-1.358]$, and its associated curve is shown in Fig. 6(d).
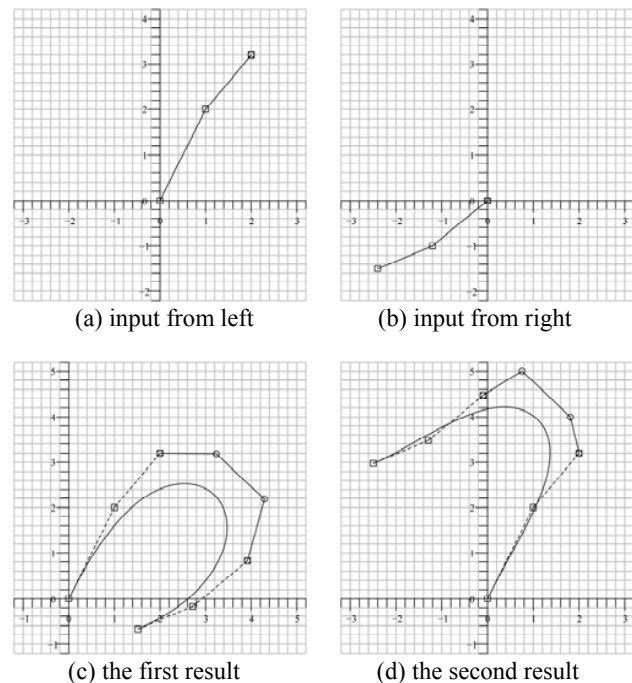

(a) input from left   (b) input from right

(c) the first result   (d) the second result

Fig. 6 I/O vectors and result curves for example 2

V. CONCLUSION

The PPH curve is a Bézier curve but not vice versa. When we want design PPH curve from control points of Bézier curve, only part of control vectors we need, and produce the other parts of control vectors. Various ways we can select the input control vectors. If we select one side of the control vectors, we can find single curve which has PPH properties. This selection makes the algorithm simple and straight forward. However, the end part of the curve and the output control vector is sensitive and unpredictable. If we select both side of the control vectors, the associated algorithm find two curves which have PPH properties. Using these two selections, all of the algorithms proposed in this paper take very short time to construct the PPH curves.

REFERENCES

[1] R.T. Farouki, T. Sakkalis, "Pythagrorean Hodographs" IBM J. RES. DEVELOP. VOL 34, No. 5, September 1990.
[2] Farouki,R.T.,1992.Pythagorean-hodograph Curves in Practical Use. In:Barnhill,R.E. (Ed.), Geometry Processing For Design and Manufacturing. SIAM, Philadelphia, p.3-33.
[3] Farouki,R.T.,Neff,1990.Analytic properties of plane offset curves.Computer Aided Geometric Design,7(1-4):83-99.
[4] Moon, H.P., Farouki, R.T., "Construction and shape analysis of PH quintic Hermite interpolants", Computer Aided Geometric Design, 2001, 18, 93-115.
[5] W. Lü "Rationality of the offsets to algebraic curves and surfaces", Applied Mathematics, 9 (Ser. B), 265-278.
[6] W. Lü, "Offset-rational parametric plane curves", Computer-Aided Geometric Design, 12, 601-616, 1995.
[7] K.K.Kubota, "Pythagorean triples in unique factorization domains", American Mathematical Monthly, 79, 503-506, 1972.