Designing Modern Linkages to Trace Bézier Curves

James D. Factor*

Abstract

A design of a class of linkages is presented which are less complex than those suggested by Kapovich and Millson in the configuration of conventional planar linkages. Conventional linkage constraints are relaxed allowing sliding joints and telescoping links. The precise number of fixed links, telescoping links, and sliding contacts is determined for this modern linkage to trace a Bézier curve of any degree in 2D or 3D space. A suggested realization of the linkage tracing a quadratic Bézier curve is provided.

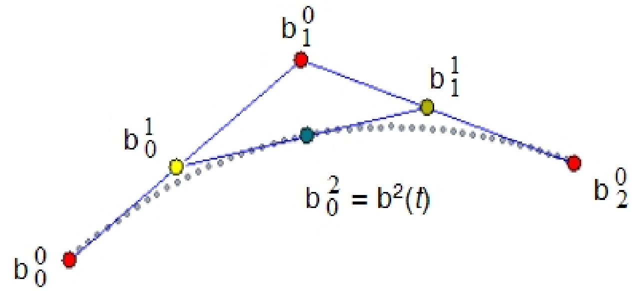

Figure 1: A Quadratic Bézier Curve Traced by an m-Linkage

1 Introduction

Kapovich and Millson [KM02] proved A. B. Kempe's Universality Theorem: *if C is a bounded portion of an algebraic curve in the plane, then there exists a planar linkage such that the orbit of one joint is precisely C.* Their theoretical result that there is a planar linkage that traces out any given algebraic curve is very elegant. Unfortunately, there is no efficient method known to realize this theorem. No method to systematically design linkages that will trace any class of free-form algebraic curves has been devised. This paper presents such a method for the class of Bézier curves. Any mechanical application requiring an exact trace or cut of a Bézier curve instead of an approximation can make use of this method. The approach slightly strengthens the notion of linkages, by allowing sliding joints and telescoping links. It weakens the class of algebraic curves to those known as Bézier curves, allowing the creation of less complicated linkages. In addition, a natural result of the design is that the curve traced by a joint is not limited to 2D space, but can be extended to 3D space.

The type of linkage used to trace the Bézier curve will be referred to as a *modern linkage (m-linkage).* In order to realize such a linkage, the approach will assume that the m-linkage is powered by multiple actuators coordinated electronically. In effect the electronic signal would be an "electronic crank" and would be the single "prime mover," which is often implemented as a physical crank driving the mechanism in a conventional linkage.

To build the m-linkage, certain structures are used. A *frame* must be built that is fixed. It is a collection of links held together by *fixed joints.* A *link* is a rigid object such as a bar or rod and is modeled using a line segment. In Figure 1, $\gamma_0^0 = [b_0^0, b_1^0]$ and $\gamma_1^0 = [b_1^0, b_2^0]$ are the links forming the frame with fixed joints b_0^0, b_1^0, and b_2^0.

In order to create a *curve*, i.e., a path of one dimension, movement must occur. A *telescoping link* is a link that is allowed to vary in length within a fixed well-defined range. For instance $\gamma_0^1 = [b_0^1, b_1^1]$ is a telescoping link in Figure 1. In addition to this, two types of movable joints are used. The first is a *sliding joint* that can move from one end of the link to the other. Second is a *sliding contact* that connects one link to itself and moves similarly to the sliding joint. Joints b_0^1 and b_1^1 are sliding joints and b_2^0 is a sliding contact in Figure 1. The entire framework of links and joints is an *m-linkage.* A curve can be traced using the m-linkage in the following manner. As b_0^1 slides on link γ_0^0 and b_1^1 slides on γ_1^0, the sliding contact b_2^0 traces a quadratic Bézier curve. The way these joints slide determines the linkage movement and the nature of the curve that is traced.

In the following sections, the linkage and the movement of its joints will be defined and shown to trace a Bézier curve, capturing its parametric direction.

2 Framework of the m-linkage

Let (V, E) be a *graph*, where V is a set of vertices and E a set of edges. From [G01], a d-dimensional framework $F_p(V, E, p)$ consists of a graph (V, E) and a function p from the vertex space V into Euclidean d-space, p: $V \longrightarrow E^d$. Consider $p(v)$, $v \in V$, as the joints and the line segments $\gamma_1 = [p(v_1), p(v_2)]$, $\{v_1, v_2\} \in E$, as

*Department of Mathematics, Statistics, and Computer Science, Marquette University, james.factor@marquette.edu

the links of the framework in Euclidean space. (V, E) is the *structure graph* of the framework $F_p(V, E, p)$ and the function p is an *embedding* of V in d-space. The structure graph captures the *combinatorial properties* of the framework describing how the links and joints are connected. These properties do not depend on the embedding function, which determines the *geometric properties* such as the position of the joints, their range of movement, and the lengths of the links. In this discussion Euclidean space will refer to Euclidean d-space for any $d = 1, 2$, or 3.

2.1 Structure Graph $(V_s(n), E_s(n))$ and Structure Digraphs $(V_s(n), A_s(n))$

The recursive subdivision algorithm defining the structure graph $(V_s(n), E_s(n))$ in the framework $F_c(n)$ is defined by Factor [F05]. The structure graph is constructed from $n+1$ connected vertices $v_0^0, v_1^0, ..., v_n^0$ forming a path ρ^0 of n edges $e_0^0, e_1^0, ..., e_{n-1}^0$. Path ρ^0 has *length n*. These vertices and edges are the only ones initially in $V_s(n)$ and $E_s(n)$, respectively. For example, $v_0^0, v_1^0, v_2^0, v_3^0$ are the initial vertices in $V_s(3)$ in Figure 2. The initial path is a 0-*level* path, where the superscript represents the level of each vertex and edge. Recursively subdivide the edges of the path ρ^0. The *subdivision of an edge* is performed by removing each edge $\{v_i^0, v_{i+1}^0\}$ and replacing it with a new vertex v_i^1 and the edges $\{v_i^0, v_i^1\}$ and $\{v_i^1, v_{i+1}^0\}$. In Figure 2, vertices v_0^1, v_1^1, and v_2^1 "subdivide" edges $\{v_0^0, v_1^0\}, \{v_1^0, v_2^0\}$, and $\{v_2^0, v_3^0\}$, respectively. Each new vertex v_i^1 is added to $V_s(n)$. Connect the newly obtained vertices with new edges e_i^1 creating a smaller path ρ^1 of length $n-1$ and add only these edges $e_i^1 = \{v_i^1, v_{i+1}^1\}$ to $E_s(n)$. Continue this process at each step until a path of length zero is created and the last vertex v_0^n (the *distinguished vertex)* is added to $V_s(n)$. The vertex v_0^3 is the distinguished vertex in Figure 2 where $n = 3$. The final structure graph therefore has $V_s(n) = \{v_0^0, v_1^0, ..., v_n^0, v_0^1, v_1^1, ..., v_{n-1}^1, ..., v_0^n\}$ and $E_s(n) = \{e_i^r | e_i^r = \{v_i^r, v_{i+1}^r\}$, for $r = 0, ..., n-1$ and $i = 0, ..., n-r-1\}$. By Factor [F05], the number of vertices in $V_s(n)$ is $(n+1)(n+2)/2$ and the number of edges in $E_s(n)$ is $n(n+1)/2$.

In order to capture the direction of the movement of a framework, the structure graph will be replaced with a structure digraph. In a *structure digraph*, the edges are oriented either from left to right, $A_s^{\Rightarrow}(n)$, or, right to left $A_s^{\Leftarrow}(n)$. This reflects the curve being traced in one direction or the other. $A_s(n)$ is the set of arcs $A_s^{\Rightarrow}(n)$ or $A_s^{\Leftarrow}(n)$. See Figure 3, where the structure digraph $(V_s(3), A_s^{\Rightarrow}(3))$ is shown. Thus, $(V_s(n), A_s(n))$ will be the structure digraph that embedding function c maps into the *directed framework* $F_c(n) = (V_s(n), A_s(n), c)$ in Euclidean space, where $F_c(n)$ traces an n^{th} degree Bézier curve, $n > 0$.

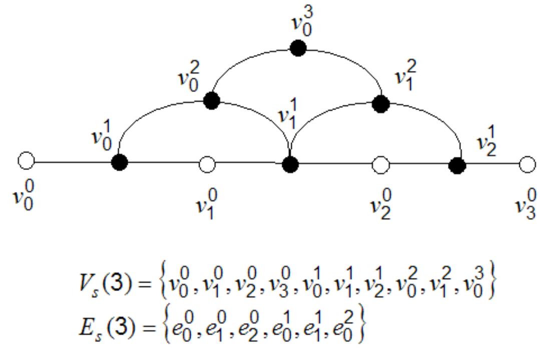

Figure 2: Structure Graph $n = 3$, $(V_s(3), E_s(3))$

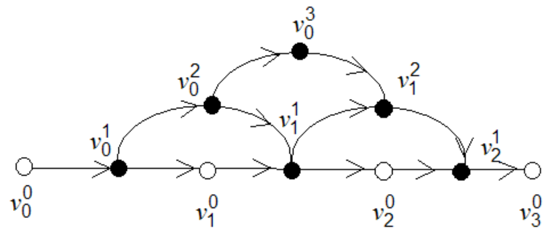

Figure 3: Structure Digraph $(V_s(3), A_s^{\Rightarrow}(3))$

2.2 Embedding Function c depending on $A_s(n)$.

The definition of the embedding function c depends on the direction of $A_s(n)$ in $F_c(n)$. The function c is an isomorphic map of the sets $V_s(n)$ and $A_s(n)$ into Euclidean space. It maps each of the initial $n+1$ vertices in indexed set $V_s(n)$ to a distinct fixed position joint $b_i^0 (i = 0, ..., n)$ in Euclidean space. As a result the n arcs between these vertices are mapped to fixed length links γ_i^0, for $i = 0, ..., n-1$. Note the *length* of each link $|\gamma_i^0|$ is the *distance,* $|b_i^0, b_{i+1}^0|$, between successive fixed joints. This path of links forms the frame for $F_c(n)$ in Euclidean space. If $A_s(n)$ is $A_s^{\Rightarrow}(n)$, then the vertices $v_0^1, v_1^1, ..., v_{n-1}^1, v_0^{n-1}, v_1^{n-1}, ..., v_{n-1}^{n-1}, v_0^n$ in $V_s(n)$ are each mapped by c to the initial position b_i^0 of each link γ_i^0 as a sliding joint or contact b_0^n, which is mapped by c from the distinguished vertex v_0^n to b_0^0. As a result, the remaining $n(n-1)/2$ arcs in $A_s^{\Rightarrow}(n)$ are mapped as bounded telescoping links γ_i^r by c on top of the fixed-length links γ_i^0, $i = 0, ..., n-2$, such that $\gamma_i^r = [b_i^r, b_{i+1}^r]$, $r = 1, \ldots, n-(i+1)$, are mapped to γ_i^0. Consequently, each initial telescoping link has an initial length of the fixed-length link to which it is mapped. If $A_s(n)$ is $A_s^{\Leftarrow}(n)$, then a similar construction exists when the vertices $v_0^1, ..., v_{n-1}^{n-1}, v_0^n$ are mapped by c to the end position b_{i+1}^0 of each link γ_i^0.

From [F05], there is a tight bound $\Theta(n^2)$ on the number of joints and joints in $F_c(n) = (V_s(n), A_s(n), c)$ following from the corollary:

Corollary 1 *For integer $n > 0$, the framework $\mathrm{F}_c(n) = (V_s(n), A_s(n), c)$ has $(n+1)(n+2)/2$ joints and $n(n+1)/2$ links.*

3 Motion and Geometry of the m-linkage

Let $[0, 1]$ be the interval of time t, where t begins at 0 and ends at 1. Consider for each integer $n > 0$, the directed framework $F_c(n) = (V_s(n), A_s(n), c)$ with indexed vertex set $V_s(n) = \{v_i^r | r = 0, ..., n \text{ and } i = 0, ..., n-r\}$. The definition of a motion for $F_c(n)$ comprises the indexed family of functions ${}_cT_i^r(t) : [0, 1] \longrightarrow E^d$, so that:

1. ${}_cT_i^r(t) = c(v_i^0) = b_i^0$, for $i = 0, ..., n$ for all $t \in [0, 1]$; (Frame of linkage.)

2. ${}_cT_i^r(t) = c(v_i^r) = b_i^r = \Phi b_i^{r-1} + \Psi b_{i+1}^{r-1}$, for $r = 0, ..., n$ and $i = 0, ..., n-r$, where $\Phi(t) + \Psi(t) = 1$; (The location of the sliding joint b_i^r at $t \in [0, 1]$)

3. ${}_cT_0^n(t) = c(v_0^n) = b_0^n = \Phi b_0^{n-1} + \Psi b_1^{n-1} = b(t)$ where $\Phi(t) + \Psi(t) = 1$; (The location of the sliding contact b_0^n at $t \in [0, 1]$)

4. Note that ${}_cT_i^r(t)$ is differentiable on the interval $[0, 1]$, for all r, i; (Guarantees that the change of position of the joints is smooth.)

5. $|{}_cT_i^r(t), {}_cT_{i+1}^r(t)| = |b_i^r, b_{i+1}^r|$ for all $t \in [0, 1]$. (The length of the links γ_i^r at $t \in [0, 1]$)

Note, by condition 1, the frame is *rigid* for the entire time of the linkage's motion. In conditions 2 and 3, the functions $\Phi(t)$ and $\Psi(t)$ control the shape of the curve and the direction in which it is traced. Condition 4, guarantees smooth motion. Finally, condition 5 addresses the length of each telescoping link at each moment of time. Figure 4 illustrates the curve traced by the motion. Again the direction in which the curve is traced is determined by $A_s(n)$. If $A_s(n)$ is $A_s^{\Rightarrow}(n)$, then $\Phi = (1-t)$ and $\Psi = t$ and the curve is traced from left to right. If $A_s(n)$ is $A_s^{\Leftarrow}(n)$, then $\Phi = t$ and $\Psi = (1-t)$ and the curve is traced from right to left.

In order to actually realize a physical linkage to trace the curve $b^n(t)$ based on the framework $F_c(n) = (V_c(n), A_c(n), c)$ and the trajectory functions ${}_cT_i^r(t)$, it is necessary to determine the range of each link's length $|\gamma_i^r|$. This information is completely determined from the location of the initial $n+1$ non-collinear joints $b_0^0, b_1^0, ..., b_n^0$ chosen for the frame. Figure 4 illustrates an m-linkage for $n = 3$ at a specific $t \in [0, 1]$. In the frame, the length of each link is fixed since it is determined by the fixed position of the initials joints. The angle between each pair of links in the frame can be determined by the direction cosines of the links. Using the Law of Cosines and elementary calculus, the minimum and maximum length of each telescoping link can be found. This approach will bound the range of movement of all telescoping links with joints sliding on the frame. From here the range of the remaining telescoping links is fully determined from the range of movement between any attached pair of telescoping links.

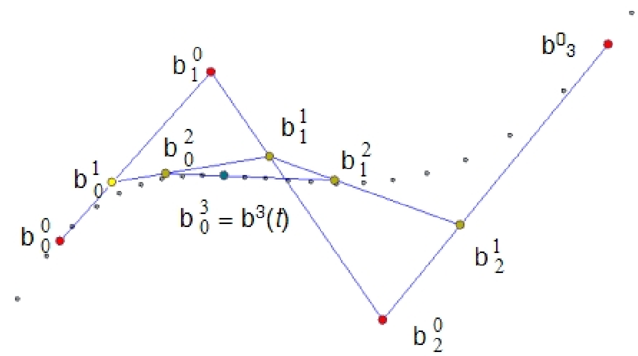

Figure 4: A Cubic Bézier Curve Traced by an m-linkage

4 The Algorithm of de Casteljau and Bezier Curves

If $\Phi = (1-t)$ and $\Psi = t$ in the motion of the linkage, then the linkage behaves the same as the algorithm of de Casteljau Algorithm [D63] for each $t \in [0, 1]$.

Algorithm:(de Casteljau) Let the points $b_0, b_1, ..., b_n \in E^d$ and $t \in R$, where R is the set of all real numbers. The point $b(t)$ is constructed on a curve by the recursive process:

1. $b_0^0 = b_0, b_1^0 = b_1, ..., b_n^0 = b_n$.
2. $b_i^r = (1-t)b_i^{r-1} + tb_i^{r-1}$ for $r = 1, ..., n$ and $i = 0, ..., n-r$.
3. $b(t) = b_0^n$.

The following lemma, from [PW01], equates the de Casteljau algorithm to Bézier curves and paves the way to proving that $F_c(n) = (V_s(n), A_s(n), c)$ traces a Bézier curve.

Lemma 2 *The algorithm of de Casteljau applied to points $b_0, b_1, ..., b_n$ and the real number t evaluates the polynomial Bézier curve with these control points at the parameter value t.*

Theorem 3 *Given a Bézier curve $b^n(t)$ of any degree integer $n > 0$ in Euclidean d-space, where $d = 1, 2,$ or 3, then an m-linkage tracing this curve can be constructed by using the framework $\mathrm{F}_c(n) = (V_s(n), A_s(n), c)$ with trajectory functions ${}_cT_i^r(t)$, $t \in [0, 1]$.*

Proof. Given a Bézier curve $b^n(t)$ of any degree integer $n > 0$ in Euclidean d-space, where $d = 1, 2$, or 3, $b_n(t)$ will have $n+1$ control vertices $b_0, b_1, ..., b_n$. Using Lemma 2, the algorithm of de Casteljau can generate this Bézier curve. For the given value n, the structure graph $(V_s(n), A_s(n))$ in the framework $F_c(n) = (V_s(n), A_s(n), c)$, where c maps into d-space is determined for the linkage. The embedding function c will define the joints of the frame of the linkage, based on the location of the $b_0, b_1, ..., b_n$, and the remaining joints. Applying the trajectory functions ${}_cT_i^r(t)$, $t \in [0, 1]$ to $F_c(n)$ results in a linkage that traces a Bézier curve by joint b_0^n, since ${}_cT_i^r(t)$ continuously applies the algorithm of de Casteljau to the movement of the linkage as t goes from 0 to 1. As a result, every point on the Bézier curve $b^n(t)$ is generated for $t \in [0, 1]$. □

See Figure 4, which illustrates an m-linkage tracing a cubic Bézier curve.

As a consequence, the m-linkage inherits many of the properties of a Bézier curve. Consider the configuration space of the curve and the m-linkage itself. A *configuration space* is defined as the set of all configurations or states of the object permitted by the motion constraints, with paths in space corresponding to motions of the object [DO04]. In particular, the movement configuration space of the sliding contact b_0^n corresponds to the traced Bézier curve $b^n(t)$ and the configuration space of the linkage is totally within the convex-hull formed by the extreme points of $b_0, b_1, ..., b_n$ which form the frame.

5 Example Realization: The Quadratic Bezier Curve

Since before the time of James Watt, linkages were designed to be powered by a single "prime mover" with all functions mechanically coordinated. This "prime mover" would often be implemented as a single crank driving the mechanism. A more modern approach for linkages has been made possible with recent developments in computer technology, coupled with improvements in electric motors and actuators [WK04]. This approach allows machines that are powered by multiple actuators coordinated electronically. In effect, the electronic signal is an "electronic crank" and acts as the single "prime mover." Such modern machines are simpler, less expensive, more easily maintained, and more reliable. This modern linkage is assumed in the realization of a linkage tracing a Bézier curve.

The linkage that traces the quadratic Bézier curve in Figure 1 could be realized as follows:

At $t = 0$, an actuator A_0^0 can be placed at b_0^0 (the start of link γ_0^0) and an actuator A_1^0 would be placed at b_1^0 (the start of link γ_1^0). Each actuator is controlled by the same electronic signal for a duration of t, as t varies from 0 to 1, and is designed to move the distance along link γ_0^0 and link γ_1^0, respectively, maintaining the ratio $t/(1-t)$ of the lengths $L1 = |\gamma_0^0|$ and $L2 = |\gamma_1^0|$. Each actuator can be driven by a pinion on a rack which is the length of each respective link γ_0^0 and link γ_1^0. The pinion of A_0^0 can be a cylinder containing, either a line or steel/wire cable with some elasticity. This line (link γ_0^1) from A_0^0 can be attached to A_1^0 having the length of $L1$ at $t = 0$ and the length of $L2$ at $t = 1$. As t changes, A_1^0 takes up the line that A_0^0 spools out by its pinion, always maintaining a taut line (a link). As the pinion of A_0^0 moves along the length $L1$ of its rack, link γ_0^1 spools out as line and is taken up by the pinion of A_1^0 as A_1^0 moves the length $L2$ of its rack. The movement of the tracing-joint $b_0^2 = b^2(t)$, in Figure 1, is a natural consequence of the above behavior of the two driving actuators. Note the tracing-joint $b_0^2 = b^2(t)$ is always at the same fixed place on the line, and its movement is only simulated by the rolling up of the line from A_0^0 by A_1^0. At time $t = 0$, a "pencil" can be placed at the point of the line located at b_0^0. As the pinions move along their racks, the "pencil" moves during the spool-out and rolling-up process. At time $t = 1$, the "pencil" is at point b_2^0. Consequently the "pencil" has traced the quadratic Bézier Curve.

6 Conclusion

A method has been presented allowing linkages to be systematically designed to trace any curve in the class of algebraic free-form curves known as Bézier curves. This method has been shown to be effective in designing an m-linkage for tracing a given Bézier curve of any degree n in 2D or 3D space. Further research will address extending this method to other classes of free-form curves, e.g., rational Bézier curves, B-Splines, NURBS, and surfaces.

References

[D63] Paul de Casteljau. *Courbes et surfaces á pôles.* Technical report, A. Citroën, Paris,1963.

[DO04] E.D. Demaine and J. O'Rourke. *Folding and Unfolding in Computational Geometry.* Unpublished monograph, 2004.

[F05] James D. Factor. *Defining a Class of Computational Curves based on a Recursive Structure Graph.* Preprint submitted for publication, 2005.

[G01] J. Graver. *Counting on Frameworks: Mathematics to Aid the Design of Rigid Structures.* Mathematical Association of America, DC, 2001.

[KM02] Michael Kapovich and John J. Millson. *Universality theorems for configuration spaces of planar linkages.* Topology, 41(6):1051-1107, 2002.

[PW01] H. Pottmann and J. Wallner. *Computational Line Geometry.* Springer Verlag, Heidelberg, Berlin u.a., 2001.

[WK04] K.J. Waldron and G.L. Kinzel. *Kinematics, Dynamics, and Design of Machinery.* 2nd Ed. John Wiley & Sons, Inc., New York, 2004.