

## Establishing Similarity of Modeled and Experimental PK/PD Hysteretic Loops using Pseudo Time Series Analysis and Dynamic Time Warping

Ron Smith, Florida Southwestern State College, Fort Myers, FL.

### Abstract

An iterative Bezier curve cubic spline algorithm is used to approximate a hysteretic loop for a pharmacodynamics(PD) dose/response curve. The accuracy of the resulting vector-valued function is tested using graphs produced through PROC SGPLOT. The centroid of each curve is found through PROC IML. Euclidean distances from each centroid to points on the respective polygonal curves (experimental and model) produce several distance vectors for each curve. A sequence of these distance vectors is plotted as a Pseudo Time Series for both the original experimental and the adapted model curves. The graphical similarity of these curves is examined using the Dynamic Time Warping (DTW) algorithm in PROC SIMILARITY. The information produced from the DTW process gives a numerical measure of how well the model follows the experimental curve. The parametric equations from this a *posteriori* model can form a composition with a time-based function to produce the final PK/PD effect versus time equations.

### Introduction

Hysteresis loops occur from the time lag of a response (effect) behind the instigator of the “effect”. In dose response curves the two typical reasons for the lag are (1) a dearth of drug receptor sites and (2) slowed drug-receptor site interaction. The aforementioned reasons produce counter-clockwise hysteresis loop, in which plasma drug concentration precedes then lags the observed effect with time. An alternative hysteretic curve process would include any situation in which the value of one variable depends on whether the other variable is increasing or decreasing. A clockwise hysteresis path occurs with the development of tolerance to a drug.

Hysteresis loops are graphical traces occasionally encountered in the analysis of pharmacokinetic and pharmacodynamic relationships. Hysteretic curves often indicate more complex drug-receptor interaction and/or the presence of confounding drug metabolites. They are typically relations and not functions, forcing a different modeling process that might involve systems of equations.

### Figure 1. Clockwise and Counter-clockwise Hysteresis loops.

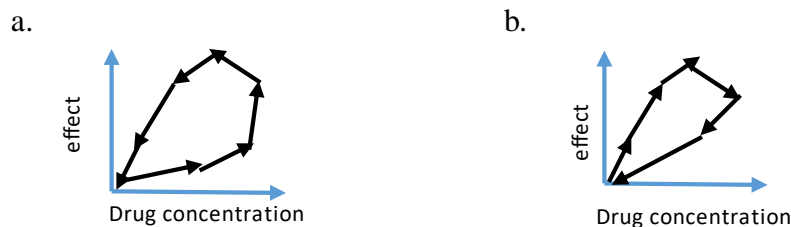


Figure 1. (a) counterclockwise movement of a dose response curve where the drug and receptor interactions are limited. (b) clockwise movement where drug tolerance has increased.

Pharmacokinetics (PK) measure effect versus time, while pharmacodynamics (PD) measure concentration versus effect. Ultimately a PK/PD curve will measure effect versus time (Fig. 2).

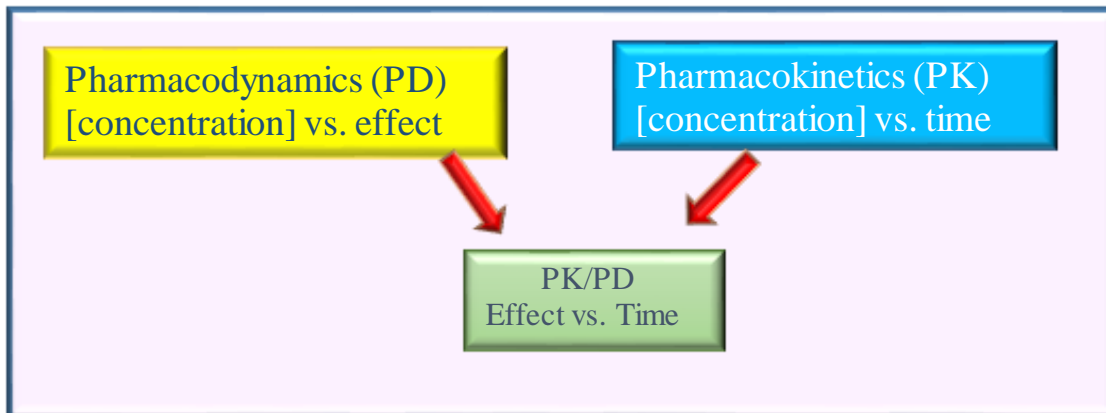


Figure 2. The combination of the dimensions of the pharmacokinetics and pharmacodynamics in pharmaceutical dose-response studies.

The present paper will describe an *a posteriori* model of the effect versus drug concentration hysteresis loop using a Bezier type curve adaptation that will produce two parametric equations of the form  $x(\tau)$  and  $y(\tau)$ , with  $\tau$  as a parametric placeholder for a function of time. That is, we have a composition where  $\tau = f(t)$ . Figure 3. shows how a time component may be displayed along with the planar hysteresis loop in an experimental setting.

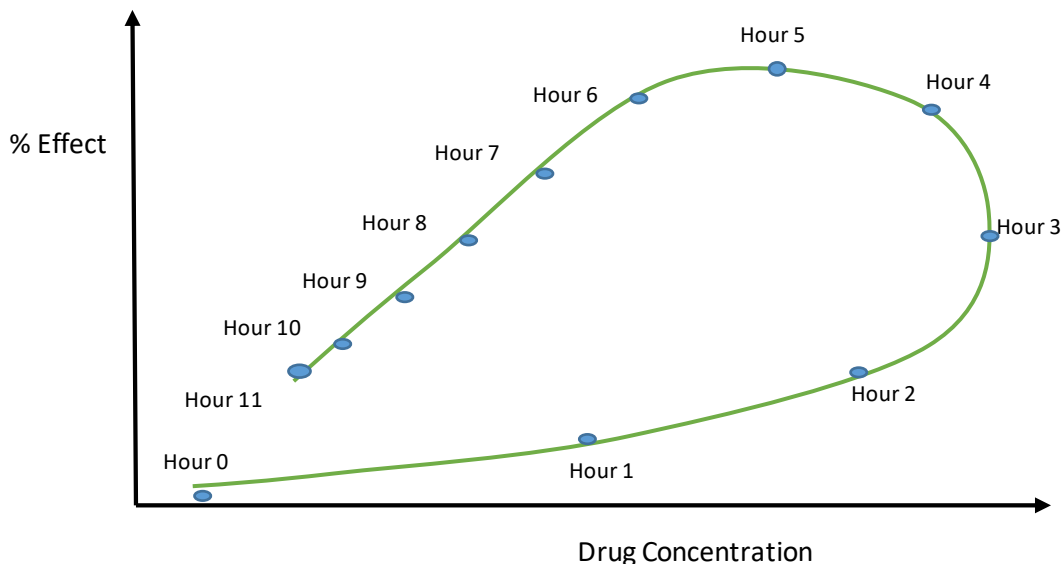


Figure 3. Hourly assay points superimposed on a typical experimental hysteresis plot. The dots represent time values where blood assays are taken along the concentration effect curve. Note: Time is often not in linear relation with the unit movement along the curve. Time and distance along the curve may be logarithmically or exponentially related.

There are many researchers who have developed *a priori* models for hysteretic systems. The equations for these systems are very briefly described in Appendix B.

The outline of the **a posteriori** process of the current paper follows:

1. A cubic Bezier spline curve is produced using an iterative process that determines the distances of 1/3 and 2/3 the measure along the complete curve path. Control points for the model curve are derived from points on the experimental curve that might be obtained through pharmaceutical research.
2. Once the centroid of each of the two curves is found, the Euclidean distance from the center of mass to each of the sequential points **on** the curve is found. The included Bezier Curve analysis insures that the starting and ending points (identified as anchor points) for the two curves are automatically the same.
3. A pseudo time series for each curve is determined using the Euclidean distance plotted against an integer n-value starting at 1 and ending with the nth point on each polygon. Here, the (n+1)<sup>st</sup> point is equivalent to the 1<sup>st</sup> point.
4. The pseudo time series are evaluated by PROC ARIMA and checked for ARIMA model and stationarity.
5. The pseudo time series values are examined in PROC SIMILARITY using Dynamic Time Warping to give a measure of the similarity of the two curves. A small difference found through DTW is taken as a confirmation that the curves are similar and that the adaptive curve might be used to model the experimental curve.

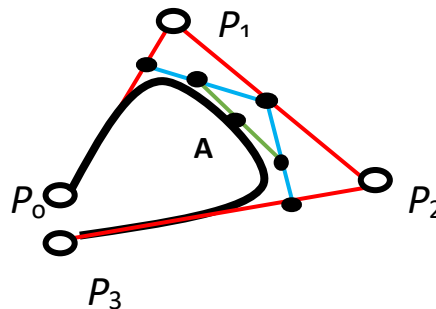
## Modeling the Dose-Response Hysteretic Loop

The shape of the hysteretic loop in the PK/PD dose-response curve is mindful of a cubic spline curve, with the cubic Bezier curve being a special form of the spline curve. A spline is a series of connected polygonal segments. The segments can be linear, quadratic, cubic, or even higher order polynomials. As we produce our model curve, we derive a set of parametric-form vector-valued functions using a set of four control points ( $P_0, P_1, P_2, P_3$ ) that produce the cubic Bezier polynomial. The cubic Bezier curve is given by is found from the binomial expansion of Equation 1.

**Equation 1:**

$$\text{Cubic Bezier } (3, \tau) = \sum_{k=0}^3 P_k (\tau)^k (1-\tau)^{3-k}$$

using the parameter,  $\tau$ , and its complement  $(1-\tau)$  since  $\tau \in [0,1]$ . The control points help form the curve, with  $P_0, P_3$  as anchor points which are on the Bezier curve (black) and  $P_1, P_2$  are control points not on the curve. The solid back dots are auxiliary control points with point A on the Bezier curve.



**Figure 4.**The four control points of the cubic Bezier spline with interior tangent at point A.

Since points  $P_0, P_3$  are already known, we need find points  $P_1, P_2$  to produce the equation for the model curve.  $P_0, P_3$  will be the same for the experimental and the adapted model curve. In order to get the best

fit for the model curve, we must obtain two points from the original curve for the dose-response data. Measuring (counter-clockwise) **along** the curve, it is common to select a point one-third of the distance from the anchor point,  $P_0$ , to the terminal anchor point,  $P_3$  and another point at two-thirds of the distance.

In terms of a matrix equation, Equation 1 becomes (now identified as Equation 2):

**Equation 2:**

$$\mathbf{B}(\tau) = \mathbf{T} \cdot \mathbf{C} \cdot \mathbf{P}$$

where  $\mathbf{B}$  contains the two parametric equations,  $x(\tau)$  and  $y(\tau)$ , that will trace the curve on the two-dimensional plane.  $\mathbf{T}$  is the  $\tau$  power matrix,  $\mathbf{C}$  is the coefficient matrix and  $\mathbf{P}$  is the matrix of control points.

In the expanded form, the matrices become:

**Equation 3:**

$$\begin{bmatrix} x(\tau) \\ y(\tau) \end{bmatrix} = \begin{bmatrix} 1 & \tau & \tau^2 & \tau^3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_0(x_0, y_0) \\ P_1(x_1, y_1) \\ P_2(x_2, y_2) \\ P_3(x_3, y_3) \end{bmatrix}$$

We need four data values from the experimental original curve and determine control point information that will interpolate our data for a model. We can state that we have four values  $\mathbf{Q}_0$ ,  $\mathbf{Q}_1$ ,  $\mathbf{Q}_2$  and  $\mathbf{Q}_3$ , from our experimental curve and that we want to determine control values so that our Bezier function will pass through these data points. To force interpolation, it is only necessary to set the definition of the control point values as follows:

**Equation 4**

$$\mathbf{Q} = \frac{1}{27} \begin{bmatrix} 27 & 0 & 0 & 0 \\ 8 & 12 & 6 & 1 \\ 1 & 6 & 12 & 8 \\ 0 & 0 & 0 & 27 \end{bmatrix} \cdot \mathbf{P}$$

The above formulas were derived by evaluating the formal representation of the cubic Bezier function at **path distance = 0, 1/3, 2/3 and 1**. The four equations produce the  $x$ -values and  $y$ -values of the control points we seek.

The Bezier function  $\mathbf{P}(n)$  determined by these control values  $\mathbf{Q}(\tau)$  will have the interpolation property we desired, namely, that:

**Equation 5**

$$\begin{aligned} \mathbf{Q}(0) &= \mathbf{P}_0 \\ \mathbf{Q}(1/3) &= \mathbf{P}_1 \\ \mathbf{Q}(2/3) &= \mathbf{P}_2 \\ \mathbf{Q}(1) &= \mathbf{P}_3 \end{aligned}$$

The relationship we derived above may be written as that found in Equation 6. This relationship is found through the inverse of the coefficient matrix:

**Equation 6**

$$\mathbf{P} = \frac{1}{6} \begin{bmatrix} 6 & 0 & 0 & 0 \\ -5 & 18 & -9 & 2 \\ 2 & -9 & 18 & -5 \\ 0 & 0 & 0 & 6 \end{bmatrix} \cdot \mathbf{Q}$$

The fact that each **row** contains positive entries that add up to 1.0 is an illustration of the fact that the data values are convex combinations of the control values.

The value from the experimentally derived curve are as follows for twenty-one points identified.

point	x	y	point	x	y
1	50	58	12	290	63
2	60	60	13	300	49
3	75	70	14	350	39
4	112	75	15	340	33
5	120	77	16	300	30
6	160	80	17	250	21
7	180	81	18	229	20
8	215	82	19	200	17
9	220	81	20	100	10
10	240	80	21	0	0
11	255	70			

**Table 1 Points Attained from Experimental Curve. Numbered points (in shaded orange columns) with x and y values.**

An in-depth examination of the curve's arc gives  $Q_0(50,58)$ ;  $Q_1$  (distance = 1/3) (220,81);  $Q_2$  (distance = 2/3) (350,39);  $Q_3(0,0)$ .

Multiplication of the four points by the coefficient matrix gives:

**Equation 7**

$$\mathbf{P} = \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 6 & 0 & 0 & 0 \\ -5 & 18 & -9 & 2 \\ 2 & -9 & 18 & -5 \\ 0 & 0 & 0 & 6 \end{bmatrix} \cdot \begin{bmatrix} 50 & 58 \\ 220 & 81 \\ 350 & 39 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 50 & 58 \\ 109.3 & 131.7 \\ 729.2 & 19.3 \\ 0 & 0 \end{bmatrix}$$

Which gives the two parametric equations:

**Equation 8**

$$x(\tau) = 50 + 174\tau + 1689\tau^2 - 1913\tau^3$$

$$y(\tau) = 58 + 219\tau - 555\tau^2 + 278\tau^3$$

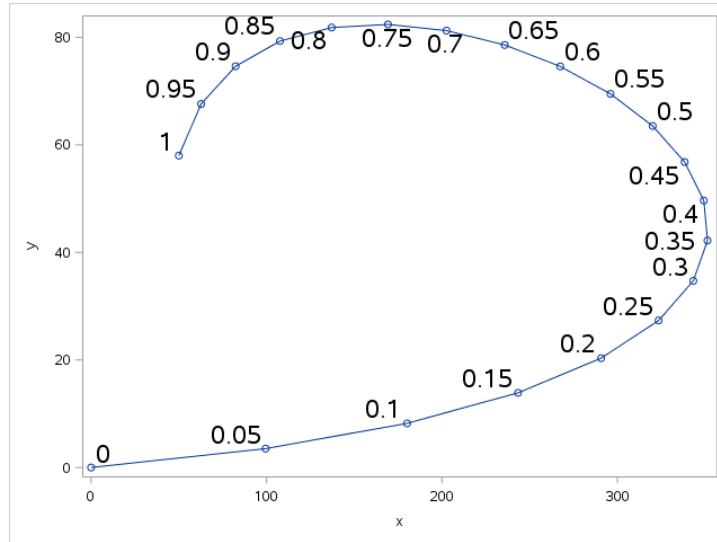
Rewriting these parametric equations with the previously mentioned  $f(t) = \tau$ , we can now plot the location on the curve with respect to the parameter time using

**Equation 9**

$$x(f(t)) = 50 + 174(f(t)) + 1689(f(t))^2 - 1913(f(t))^3$$

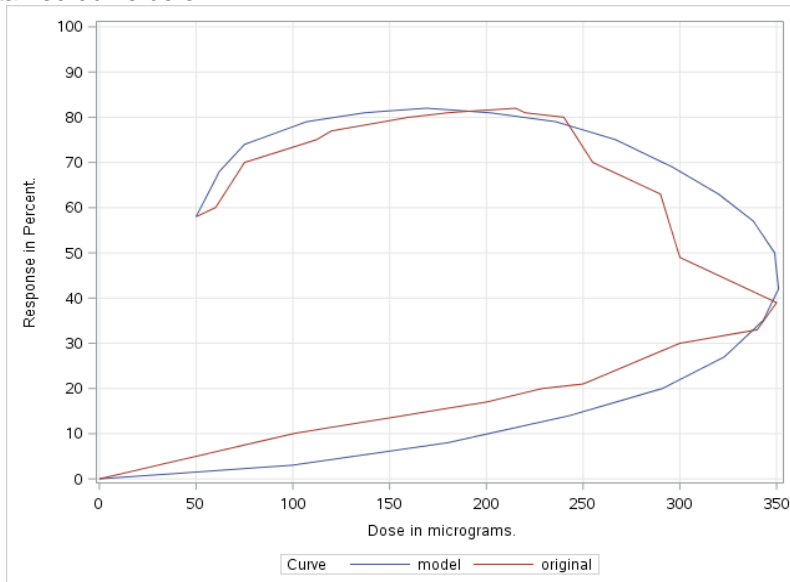
$$y(f(t)) = 58 + 219(f(t)) - 555(f(t))^2 + 278(f(t))^3$$

These equations produce the graph shown below for the adaptive model Bezier curve. The numerical values associated with the dots on the curve are  $\tau$  - values.



**Figure 5. The Adapted Model Curve. Showing parametric tau-values at 0.05 values along the curve. See Appendix C for SAS code.**

From these parametric equations, we can add the modeled-curve graph to that of the original experimentally obtained curve below:



**Figure 6. Graph of the Original Curve and the Adapted Model Curve. Common plot of the experimental curve (red) and the modeled curve (blue).**

Point	x	y	Point	x	y
1	50	58	12	338	57
2	62	68	13	349	50
3	75	74	14	351	42
4	107	79	15	343	35
5	137	81	16	323	27
6	169	82	17	291	20
7	202	81	18	243	14
8	236	79	19	180	8
9	267	75	20	99	3
10	296	69	21	0	0
11	320	63			

**Table 2. Numbered points (in shaded orange columns) with x and y values (the closed polygon actually starts at (0,0) and returns to (0,0)).**

### Produce a Centroid for Each Curve using PROC IML.

The area of a simple closed polygon with vertices  $(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})$ , is given by **Equation 10**

$$Area = A = \frac{1}{2} \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i)$$

In the formula, the vertex  $(x_n, y_n)$  is identified with the first vertex,  $(x_0, y_0)$ . The centroid of the polygon occurs at the point  $(cx, cy)$ , where

**Equation 11**

$$cx = \frac{1}{6A} \sum_{i=0}^{n-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$

$$cy = \frac{1}{6A} \sum_{i=0}^{n-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$

The above equations are employed in finding the “center of mass” (centroid) of closed polygons formed from the joining of the ends of the two curves under study (the experimental drug curve and the model cubic Bezier curve).

Rick Wicklin, of the SAS Institute, has produced (in his blog, *The Do Loop*) a very fine SAS procedure in PROC IML to find the centroid (barycenter) of a simple closed polygon. This procedure (see appendix) was employed to find a barycentric point for each of the experimental curve and the adaptive model curve. The results of the aforementioned procedure are presented in the following tables and graphs.

Experimental Drug Polygon	
Centroid x	Centroid y
161.583	42.8722

**Table 3. Calculated x and y values of the centroid based upon the program furnished by Dr. Rick Wicklin in *The DO Loop*.**

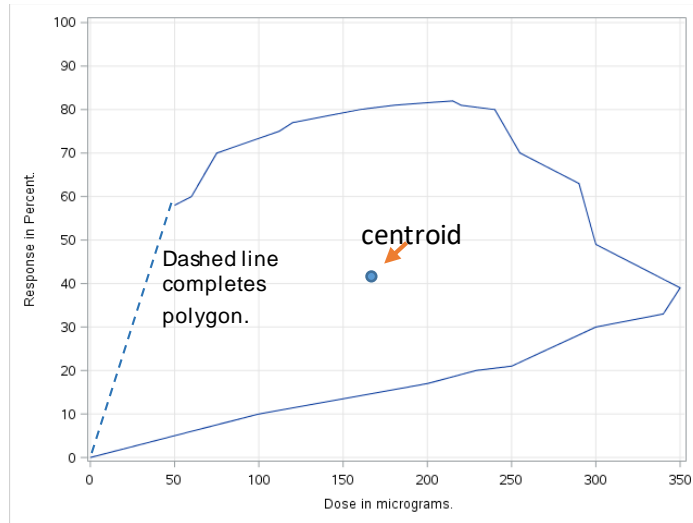


Figure 7. Plot of the experimental curve showing the calculated centroid position.

Adaptive Model Polygon	
Centroid x	Centroid y
172.895	41.5548

Table 4. Calculated x and y values of the centroid based upon the program furnished by Dr. Rick Wicklin in *The DO Loop*.

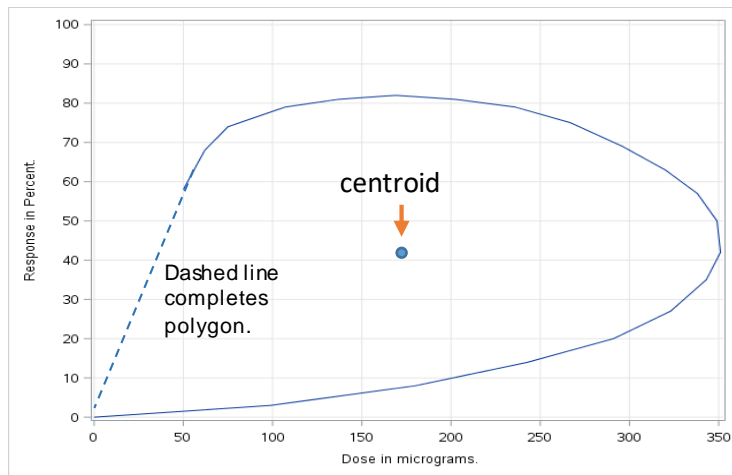


Figure 8. Plot of the modeled curve showing the calculated centroid position.

Compared to the absolute distances found in the experimental and adapted model curves, the centroids are quite close in location. While the goal of the procedure is to test the similarity of the two curves, it is of note that the Euclidean distance between the centroids (although not exactly coincident) is proximal (approximately 9.1 units apart).



Finding the Euclidean distance from the center of mass to each of the sequential points on the curve.

Using the distance formula,  $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ , the Euclidean distance from the centroid to each point in the data set that lies on the curve is found. The Euclidean distance is the distance metric commonly used in simpler shape analyses. The results are in the two tables that follow:

Point	X-value	Y-value	Euclid Distance from Centroid
1	50	58	112.603
2	60	60	103.016
3	75	70	90.733
4	112	75	59.081
5	120	77	53.794
6	160	80	37.162
7	180	81	42.343
8	215	82	66.215
9	220	81	69.759
10	240	80	86.763
11	255	70	97.277
12	290	63	129.985
13	300	49	138.553
14	350	39	188.457
15	340	33	178.690
16	300	30	139.015
17	250	21	91.083
18	229	20	71.192
19	200	17	46.317
20	100	10	69.807
21	0	0	167.173

**Table 5. Euclidean distances of points on the experimental curve is found using the distance formula.**

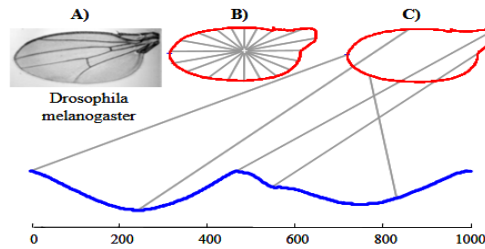
Point	X-value	Y-value	Euclid Distance from Centroid
1	50	58	123.991
2	62	68	114.005
3	75	74	103.132
4	107	79	75.792
5	137	81	53.333
6	169	82	40.632
7	202	81	49.020
8	236	79	73.378
9	267	75	99.871
10	296	69	126.127
11	320	63	148.660
12	338	57	165.825
13	349	50	176.307
14	351	42	178.105
15	343	35	170.231
16	323	27	150.809
17	291	20	120.055
18	243	14	75.325
19	180	8	34.299
20	99	3	83.349
21	0	0	177.819

**Table 6. Euclidean distances of points on the modeled curve is found using the distance formula.**

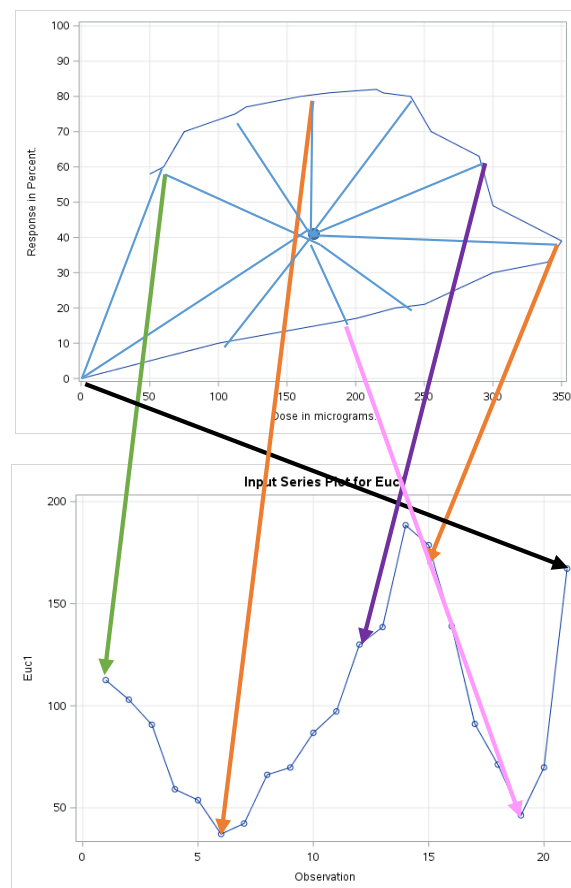
## A Pseudo Time Series is formed.

A pseudo time series is determined using the Euclidean distance plotted against number of the  $n^{\text{th}}$  sequential point on the curve. Here, the  $(n+1)^{\text{st}}$  point is equivalent to the  $1^{\text{st}}$  point.

Wei (et al.) demonstrates in the following figure how a plot of the Euclidean distance against the point designation produces a Pseudo Time Series.

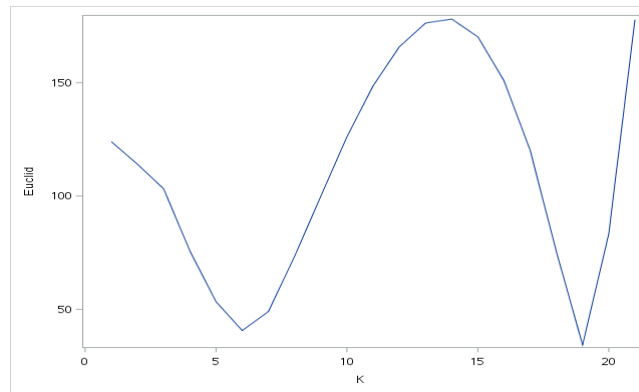


**Figure 9. Example used to demonstrate mapping of the centroid and Euclidean distance to convert a closed simple curve to a Pseudo Time Series. From Wei (2008).**



**Figure 10. The colored coded arrows in the figure above point to the place in the Pseudo Time Series that relates to the Euclidean distance from that example point to the centroid.**

The process used in producing the time series for the experimental drug curve is also used to produce a Pseudo Time Series for the adaptive model curve



**Figure 6. Plot of the Euclidean Distance versus the index number of the Pseudo Time Series.**

### The Pseudo Time Series Tested for Stationarity using PROC ARIMA

The Euclidean distances for the Pseudo Time Series were obtained from a closed polygon and that process would necessarily indicate a stationary time series. However, the stationarity of each Pseudo Time Series was further tested through an ARMA model using PROC ARIMA. The following tables display the mean and standard deviation of the respective pseudo time series.

Mean and Std. Dev. Model	
Mean of Working Series	111.4317
Standard Deviation	46.86071
Number of Observations	21

**Table 7. Mean and St. Dev. of the Model Pseudo Time Series Found through PROC ARIMA**

Mean and Std. Dev. Experimental	
Mean of Working Series	97.0961
Standard Deviation	43.9356
Number of Observations	21

**Table 8. Mean and St. Dev. of the Experimental Pseudo Time Series Found through PROC ARIMA**

Example code for the model follows:

```
proc arima data=adaptive;
  identify var=euc dist(1) stationarity=(ADF=(0,1,2,3)) outcov=adf;
  label euc_dist = "Adaptive Model Curve";
run;
```

The models for both times series shows that the MA (1) model is appropriate. This model assures us that the two Pseudo Time Series are stationary (since these times series are necessarily finite). If the respective time series are not stationary (i.e. with a constant mean), we would not be able to compare them properly.

SCAN (from PROC ARIMA)			
Experimental Pseudo Time Series		Modeled Pseudo Time Series	
p+d	q	p+d	q
0	1	0	1
2	0		

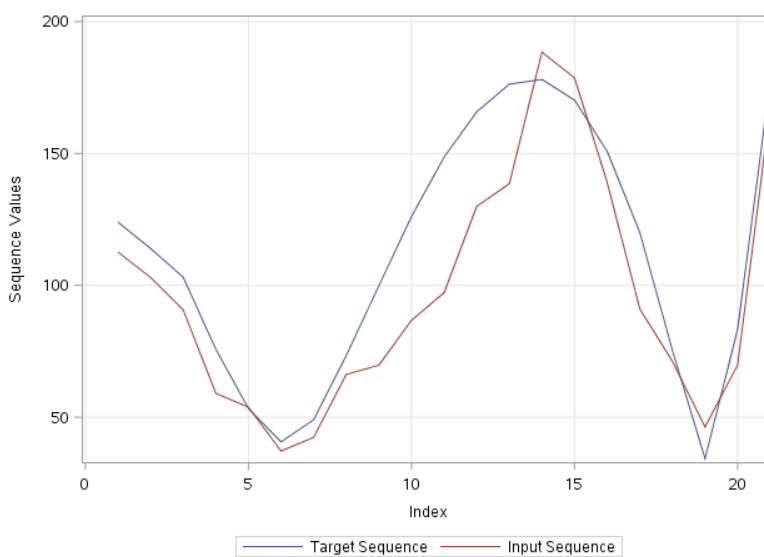
**Table 9. The models suggested by the SCAN option from PROC ARIMA. Models suggested by the ARIMA Procedure with ARMA (0,1) chosen as the applicable format. MA(1):  $(Y_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1})$  is stationary for all for all  $\theta$ ) (Hamilton, 1994).**

The following data set shows the Euclidean Distance values for the two curves and the index point.

Index	Euclidean Distance for Model Curve	Euclidean Distance for Experimental Curve	Index	Euclidean Distance for Model Curve	Euclidean Distance for Experimental Curve
1	123.991	112.603	12	165.825	129.985
2	114.005	103.016	13	176.307	138.553
3	103.132	90.733	14	178.105	188.457
4	75.792	59.081	15	170.231	178.690
5	53.333	53.794	16	150.809	139.015
6	40.632	37.162	17	120.055	91.083
7	49.020	42.343	18	75.325	71.192
8	73.378	66.215	19	34.299	46.317
9	99.871	69.759	20	83.349	69.807
10	126.127	86.763	21	177.819	167.173
11	148.660	97.277	13	176.307	138.553

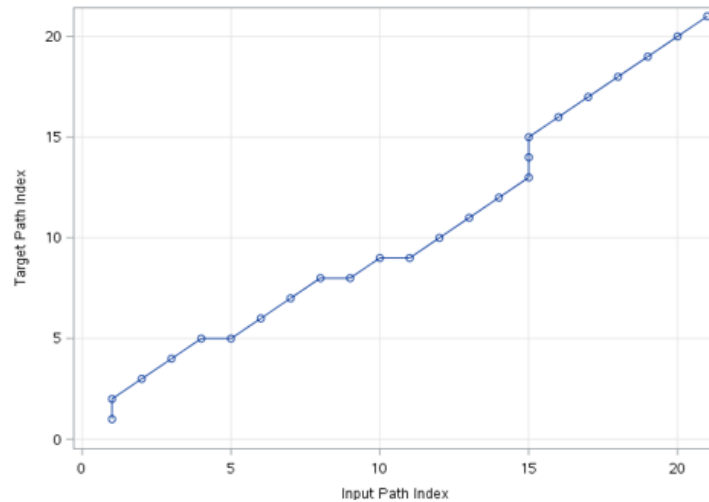
**Table 10. Dataset Summary of the Euclidean Distances for both curves. These were compared using PROC Similarity**

The two pseudo time series are shown in an overlay in the graph below. It appears that they are in good agreement.



**Figure 7. The two pseudo time series plotted on the same graph. The times series in red(input) is the experimental curve Pseudo Time Series. The time series in blue (target) is the modeled Pseudo Time Series.**

The table and warp map that follows show that the distances are kept relatively minimum. As can be seen from the dynamic time warp map, most of the deviation from similarity takes place from points 7 to 15. This is also noted in the original comparison of the experimental curve and the curve generated using control points for the cubic Bezier spline.

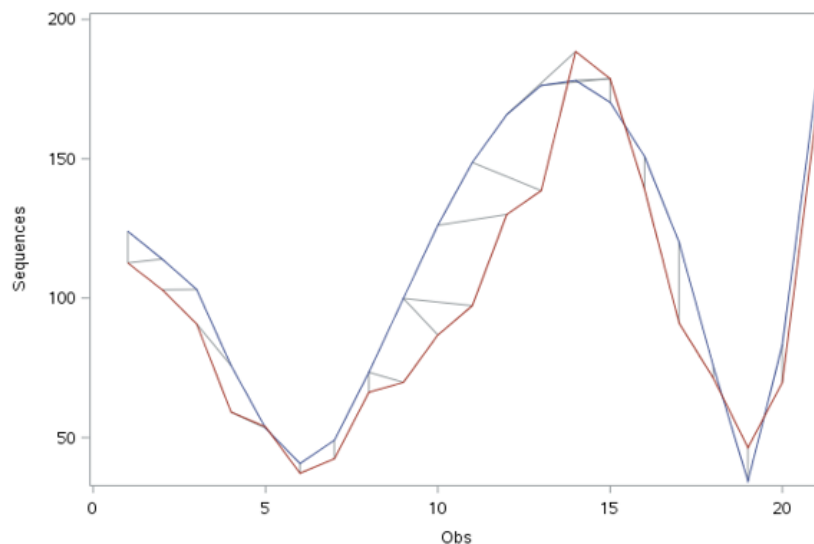


**Figure 8. Path diagram for the minimal paths for linking the nodes for both Pseudo Time Series.**

Path	Numb.	Path %	Input %	Target %	Max	Path Maximum %	Input Maximum %	Target Maximum %
Missing Map	0	0	0	0	0	0	0	0
Direct Maps	18	75	85.71	85.71	6	25.00	28.57	28.57
Compression	3	12.5	14.29	14.29	2	8.333	9.524	9.524
Expansion	3	12.5	14.29	14.29	1	4.167	4.762	4.762
Warps	6	25	28.57	28.57	2	8.333	9.524	9.524

**Table 11. This table shows that there were 6 warps (3 compressions and 3 expansion).**

The warp diagram displays good agreement between the respective time series.



**Figure 14. The trace in red is the experimental curve and the trace in blue is the modeled curve. This map shows the minimum distance map for connecting nearest nodes. Note that the majority of the difference in distances comes from indices 5 to 15.**

The minimum relative measure of connecting the nearest nodes of the respective time series is approximately 8.33.

Minimum Measure Summary	
Input Variable	Model Pseudo Time Series
Experimental Pseudo Time Series	8.32567

**Table 12.** The minimum measure of connecting sequential elements of the time series.

## Conclusion

The processes involved in retro-fitting a Bezier-type curve to the experimentally plotted PK/PD curve produced a very similar model. The experimental curve in the present study is fairly easily modeled by a cubic Bezier spline. Non-simple curves and more complex experimental curves might require splines of higher order. PROC TRANSREG is very effective modeling splines, but, as found in this case, parametric equations are needed to model the curve as a relation and not as a function.

The intent of Dynamic Time Warping is to find a minimally weighted sum of path lengths between the two compared series. In the present example, the sum of the Dynamic Time Warp distances is 8.32, a much smaller number than the distances plotted for the vertical axis of each of the two series.

It is well understood that Dynamic Time Warping does not produce a distance metric in the usual sense in that it does not obey the triangle inequality. However, two series that are identical in magnitude and phase should, theoretically, produce a weighted Dynamic Time Warping of 0. Thus, a relatively small value compared to the magnitude of the series numerical values, would indicate a stronger similarity for the two series that are compared.

A production of a model PK/PD curve that is very similar to the experimentally derived curve provides a chance to form a set of parametric equations that define the system under study. These parametrically defined equations can be composed with a time-parameter function to allow the prediction of affect response versus time. One could then calculate the percent response to a drug based upon the time elapsed since initial treatment. It is interesting that the solutions to the *a priori* modeling process (briefly described in Appendix 2) would produce equations similar to the composition functions of the Bezier equations with exponential time curves.

## Appendices

### Appendix A

Program adapted from Wicklin, R., Compute the centroid of a polygon in SAS, *The DO Loop*.

```
proc iml;
  /* _PolyCentroid: Return the centroid of a simple polygon.
     P is an N x 2 matrix of (x,y) values for consecutive vertices. N > 2. */
  start _PolyCentroid(P);
    lagIdx = 2:nrow(P) || 1;          /* vertices 2, 3, ..., N, 1 */
    xi = P[,1];      yi = P[,2];     /* x_i and y_i for i=1..N */
    xip1 = xi[lagIdx]; yip1 = yi[lagIdx]; /* x_{i+1} and y_{i+1}, x_{N+1}=x_1 */
    d = xi#yip1 - xip1#yi;           /* vector of difference of products */
    A = 0.5 * sum(d);                /* signed area of polygon */
    cx = sum((xi + xip1) # d) / (6*A); /* X coord of centroid */
    cy = sum((yi + yip1) # d) / (6*A); /* Y coord of centroid */
    return( cx || cy );
  finish;

  /* test the function by using an L-shaped polygon */
  L = { 0 0, 110 3 , 189 7.7 , 218 15 , 235 30, 227 38 , 211 47 , 189 51 , 162 53, 132 54 , 101 51 , 70 45 , 40 30};
  centroid = _PolyCentroid(L);
  print centroid;
```

### Appendix B

Use of the Bouc-Wen model for hysteresis loop systems and the Hill function from Biochemistry are becoming more common in dose response curves, gene regulation networks, and studies in anesthesiology. Drugs and enzymes often synergistically interact in biological systems. Therefore, any drugs, their metabolites and activated enzymes must be accounted for in a system of equations and/or differential equations. The Hill function is:

#### Equation 14

$$x_1 = \frac{ax_0^n}{b^n + x_0^n}$$

Where  $x_0$  may denote the drug concentration in the experiment.  $x_1$  represents molecular structural changes in a hybrid form that allows the expression of subsequently expressed enzymes. The values for  $a$  and  $b$  are constants relevant to the system, while  $n$  is the Hill coefficient. The overall model for the system is designed from an adapted Bouc-Wen non-linear differential equation and its sub-functions.

### Equation 15

$$\frac{dw}{dt} = \rho \left( \frac{dx}{dt} - \sigma \frac{dx}{dt} w + \kappa x \left| \frac{dw}{dt} \right| - \gamma w \left| \frac{dw}{dt} \right| \right) \quad \text{and}$$

### Equation 16

$$x = -1 + 2 \frac{x_1 - x_0}{c_m - c_0}, c_0 = \frac{ax_{0\min}^n}{b^n + x_{0\min}^n}$$
$$c_m = \frac{ax_{0\max}^n}{b^n + x_{0\max}^n}, w = -1 + 2 \frac{x_2 - x_{2\min}}{x_{2\max} - x_{2\min}}$$

Here  $x_2$  is the expressed enzyme concentration,  $w$  is the normalization of  $x_2$  and  $\rho, \sigma, \gamma, \kappa$  are model parameters determined through experiment.

**Note: The above equations can be modeled by PROC MODEL.**

### Appendix C

The following program shows the code for the Bezier Curve in Figure 5. Code suggested by Dr. Rick Wicklin (personal communication).

```
data adapt;
do tau = 0 to 1 by 0.05;
  x = 50 + 174*tau + 1689*tau**2 - 1913*tau**3;
  y = 58 + 219*tau - 555*tau**2 + 278*tau**3;
  output;
end;
run;

proc sgplot data=adapt;
series x=x y=y /markers datalabel=tau datalabelattrs=(size=16);
run;
```



## References

- Belal, S., Shashank, K. T., Shashank K., Md. Nadeem A., Satish, K.D., Rahul S. (2013) Inverse Point Solution of Bezier Curve, *International Journal of Scientific & Engineering Research*, 4(6)
- DeMedeiros-Martins, A., Doria-Neto, A.D., and DeNelo, J.D. (2004) Comparison between Mahalanobis distance and Kullback-Leibler divergence in clustering analysis. Available at <http://www.wseas.us/e-library/conferences/udine2004/papers/483-202.pdf>
- Hamilton, J.D. (1994), *Time Series Analysis*, Princeton, NJ: Princeton University Press.
- Hu J., Qin, K. R., Xiang, C., Lee T. H. (2012), Modeling of Hysteresis in Gene Regulatory Networks, *Bulletin of Mathematical Biology*. 74(8) 1727-1753
- Leonard, M., Sloan, J. Lee, T. and Alzheimer, B. (2008) "An Introduction to Similarity Analysis Using SAS", Proceedings of International Symposium of Forecasting. SAS Institute Inc., Cary, NC. Available at: <https://support.sas.com/rnd/app/ets/papers/similarityanalysis.pdf>
- Louizos, C., Yanez, J., Forrest, L., Davies, N. (2014) Understanding the hysteresis Loop conundrum in pharmacokinetic / pharmacodynamic relationships. *J. Pharm. Sci.* 17(1) 34-91.
- Nygaard, N., Time Series (2006) Found at: [www.finmath.uchicago.edu/docs/Time%20Series.pdf](http://www.finmath.uchicago.edu/docs/Time%20Series.pdf)
- Pleuvry, Barabara J. (2005) Hysteresis in Drug Response, *Anaesthesia & Intensive Care Medicine*, 6(8), 286-287
- Ramsay, J.O. and Silverman, B.W. (2005). Functional data analysis, 2<sup>nd</sup> Ed. New York: Springer
- Rumpler, M.J., Kandala, B., Vickroy, T. W., Hochhaus, G., Samms, R. A., (2013) Pharmacokinetics and pharmacodynamics of glycopyrrolate following continuous-infusion in the horse. *J Vet. Pharmacol. Therap.* 37, 133-144
- Taghnia, J. (2011) *Forcing Bezier Interpolation*. Available at: <http://www.javadtagnia.com/control-system-math/forcing-bezier-interpolation>
- Tran, V., (2007) Reverse Engineering Bezier Curves. Found at: [Polymath Programmer polymathprogrammer.com/2007/06/27/reverse-engineering-bezier-curves/](http://polymathprogrammer.com/2007/06/27/reverse-engineering-bezier-curves/)
- Wei L., E Keogh, X Xi, M Yoder, (2008) Efficiently finding unusual shapes in large image databases. *Data Mining and Knowledge Discovery*, New York: Springer
- Wicklin, R., (2012) Testing data for multivariate normality, *SAS Blogs Home > The DO Loop* Available at: <http://blogs.sas.com/content/iml/2012/03/02/testing-data-for-multivariate-normality.html>
- Wicklin, R., (2012) Compute the multivariate normal density in SAS, *SAS Blogs Home > The DO Loop* Available at: <http://blogs.sas.com/content/iml/2012/07/05/compute-multivariate-normal-density.html>
- Wicklin, R., (2016) Compute the centroid of a polygon in SAS, *SAS Blogs Home > The DO Loop* Available at: <http://blogs.sas.com/content/iml/2016/01/13/compute-centroid-polygon-sas.html>
- Zhang, D. and Lu, G. (2004). Review of shape representation and description techniques. *Pattern Recognition*, 37(1): 1-19.

## Acknowledgements

The author would like to thank Rick Wicklin, PhD, Distinguished Researcher at the SAS Institute in Cary, North Carolina and Arthur Li, Professor and Researcher at the City of Hope, Duarte, California. These gentlemen provided much needed and much appreciated structural and editorial support for this paper.

## Contact Information

Your comments and questions are valued and encouraged. Contact the author at:

Ronald Smith  
Florida Southwestern State College / Professor and Statistical Consultant  
(239) 292-1810  
ronsmith.math@gmail.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.