Robot Path Application

Peter: "Dr. Kaw, I am taking a course in manufacturing. We are solving the following problem. A robot arm with a rapid laser is used to do a quick quality check, such as the radius of hole, on six holes on a rectangular plate $15'' \times 10''$ at several points as shown in Table 1 and Figure 1.

Table 1 The coordinate values of six holes on a rectangle plate.

x	y
2.00	7.2
4.5	7.1
5.25	6.0
7.81	5.0
9.20	3.5
10.60	5.0

I am using Excel to fit a fifth order polynomial through the 6 points. But, when I plot the polynomial, it is taking a long path! (Figure 2)"

Kaw: "Why do you not just join the consecutive points by a straight line; just like the kids do at Pizza Hut™ with those 'Connect the dots' activities?"

Peter: "You are making me hungry and I wish it were that easy. The path of the robot going from one point to another point needs to be smooth so as to avoid sharp jerks in the arm that can otherwise create premature wear and tear of the robot arm."

Kaw: "As I recall, you took my course in Numerical Methods. What was that …… one year ago?"

Peter: "Yes, your memory is sharp, but my retention from that course – can we not talk about that?!?"

Kaw: "Come into my office. I wrote this program using Maple. See this function, $f(x) = 1/(1+25x^2)$. I am choosing 7 points equidistantly (Table 2) between –1 and 1.

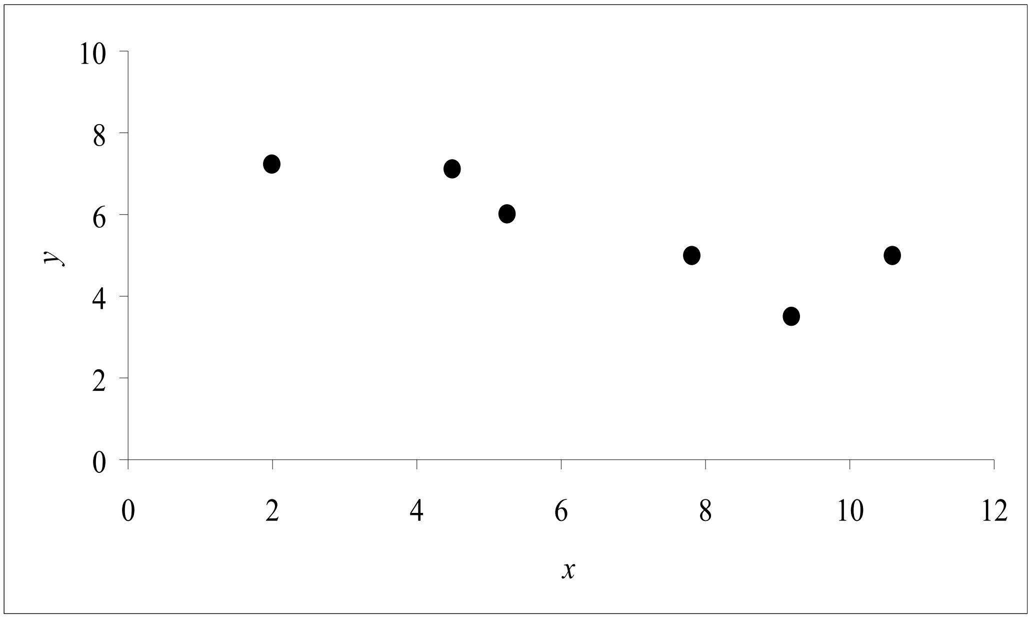

Figure 1 Locations of holes on the rectangular plate.

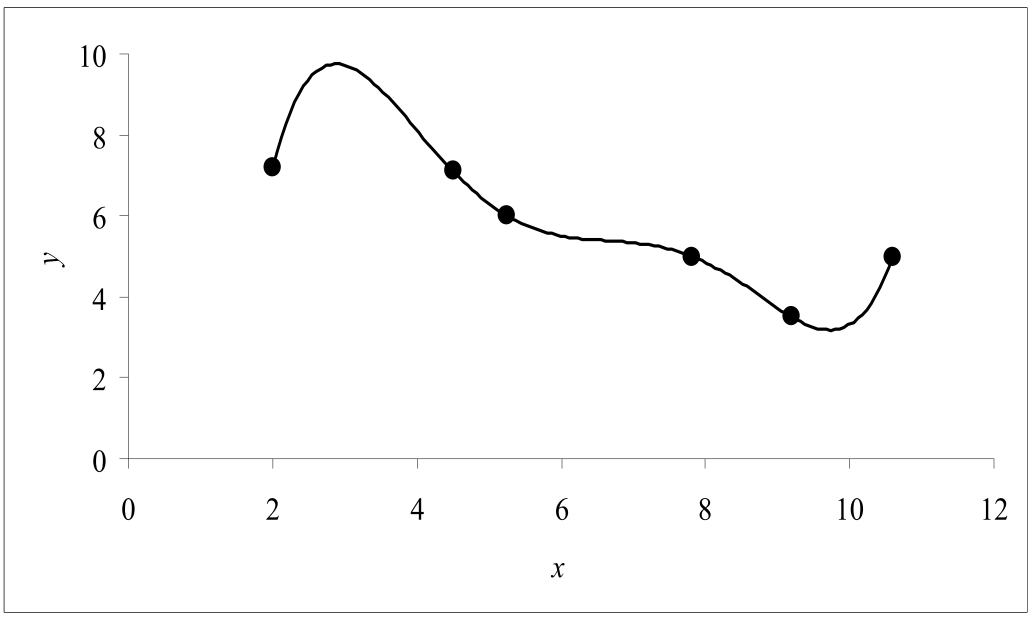

Figure 2 Approximating the path of the robot using 5th order polynomial.

Now look at the sixth order interpolating polynomial and the original function (Figure 3). See the oscillations in the interpolating polynomial. In 1901, Runge used this example function to show that higher order interpolation is a bad idea. One of the solutions to your robot path problem is to use quadratic or cubic spline interpolation. That will give you a smooth curve with fewer oscillations, and a smoother and shorter path."

Table 2 The coordinate values of 7 equidistantly spaced points.

x	$y = \dfrac{1}{1+25x^2}$
-1	0.038462
-0.66667	0.0826
-0.33333	0.264706
0	1
0.333333	0.264706
0.666667	0.082569
1	0.0385

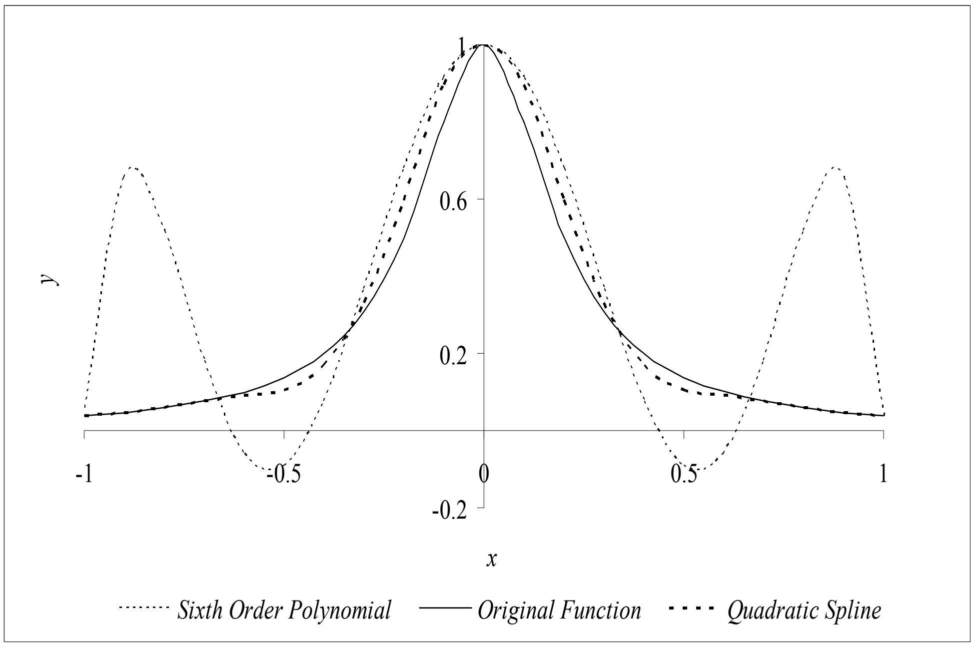

Figure 3 Runge's function interpolated.

Peter: "Okay. Let's give that a try."
Kaw: "Now, let's try generating a set of cubic splines to go through the data:"

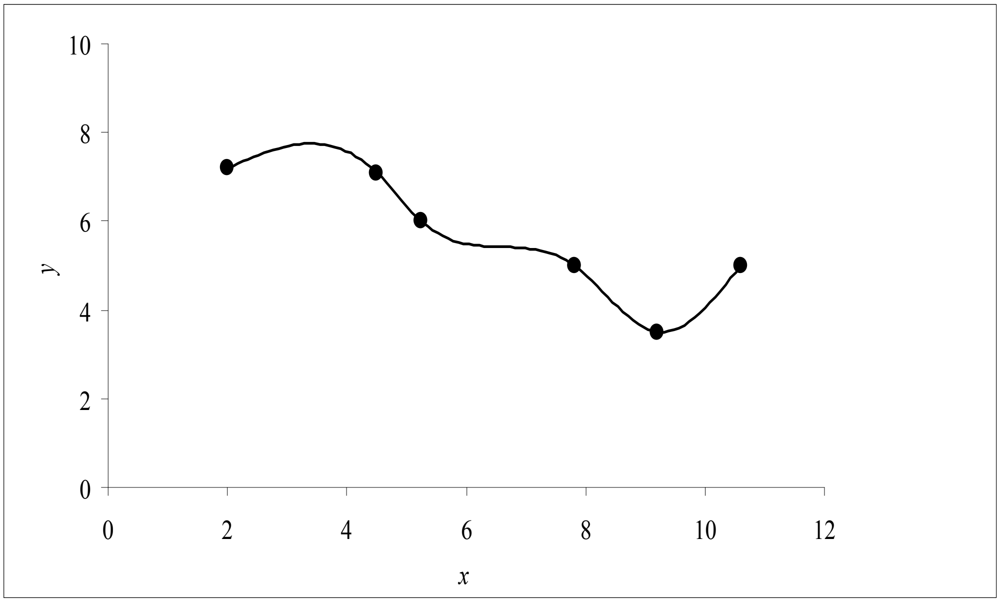

Figure 4 Path of the robot arm using cubic spline interpolation.

Peter: "Wow! That (Figure 4) looks much better!"
Kaw: "It may look better, but let's find out for sure. See if you can combine the two plots (Figure 5) and compare the lengths of each path."
Peter: "The length of a path S if $y = f(x)$ from a to b is given by

$$S = \int_a^b \sqrt{1 + \left(\frac{df}{dx}\right)^2}\, dx$$

Right?"

Table 3 Comparison of the length of curves.

Type of interpolation	**5th order polynomial**	**Cubic Spline**
Length of Curve	14.919"	11.248"

Kaw: "Yes! You solved the problem. See Table 3 for answers."
Peter: "I guess your class was good for something after all, Dr. Kaw."
Kaw: "Are you sure? You could have always fallen back on the connecting-the-dots method. Besides, you don't want to grow up … you're a Pizza Hut™ kid, right?"
Peter: "That's a Toys Я' Us™ kid. You'll do anything to be reminded of songs, won't you?"

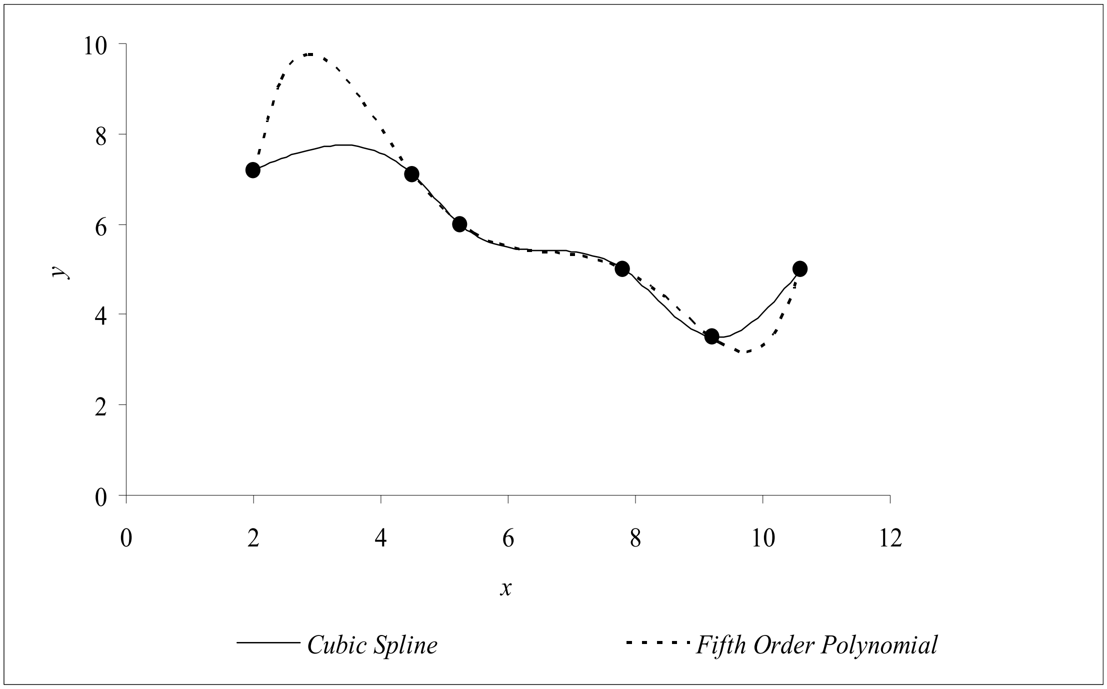

Figure 5 Path of robot arm compared using polynomial interpolation and cubic spline interpolation.