Method of least squares

J. M. Powers
University of Notre Dame

February 28, 2003

One important application of data analysis is the method of least squares. This method is often used to fit data to a given functional form. The form is most often in terms of polynomials, but there is absolutely no restriction; trigonometric functions, logarithmic functions, Bessel functions can all serve as well. Here we will restrict ourselves to strictly scalar functions of the form

$$x = f(t; a_j), \qquad j = 1, \ldots, M,$$

where x is a dependent variable, t is an independent variable, f is an assumed functional form, and a_j is a set of M constant parameters in the functional form. The analysis can easily be extended for functions of many variables. Mathematically, the fundamental problem is given

- a set of N discrete data points, x_i, t_i, $i = 1, \ldots, N$,
- an assumed functional form for the curve fit $f(t, a_j)$ which has M parameters a_j, $j = 1, \ldots, M$,

find the *best set of parameter values* a_j so as to minimize the least squares error between the curve fit and the actual data points. That is, the problem is to find a_j, $j = 1, \ldots, M$, such that

$$\ell_2 = ||x_i - f(t_i, a_j)||_2 \equiv \sqrt{\sum_{i=1}^{N} \left(x_i - f(t_i, a_j)\right)^2},$$

is minimized. Here ℓ_2 represents a total error of the approximation. It is sometimes called a "norm" of the approximation or an "L-two norm." The notation $|| \cdot ||_2$ represents the L-two norm of a vector represented by "$\cdot$."

In the least squares method, one

- examines the data,
- makes a non-unique judgment of what the functional form might be,
- substitutes each data point into the assumed form so as to form an overconstrained system of equations,
- uses straightforward techniques from linear algebra to solve for the coefficients which best represent the given data *if* the problem is linear in the coefficients a_j,
- uses techniques from optimization theory to solve for the coefficients which best represent the given data *if* the problem is non-linear in a_j.

The most general problem, in which the dependency a_j is non-linear, is difficult, and sometimes impossible. For cases in which the functional form is linear in the coefficients a_j or can be rendered linear via simple transformation, it is possible to get a unique representation of the best set of parameters a_j. This is often the case for common curve fits such as straight line, polynomial, or logarithmic fits.

Let us first consider polynomial curve fits. Now if one has say, ten data points, one can in principle, find a ninth order polynomial which will pass through all the data points. Often times, especially when there is much experimental error in the data, such a function may be subject to wild oscillations, which are unwarranted by the underlying physics, and thus is not useful as a predictive tool. In such cases, it may be more useful to choose a lower order curve which does not exactly pass through all experimental points, but which does minimize the error.

Unweighted least squares

This is the most common method used when one has equal confidence in all the data.

Example 0.1

Find the best straight line to approximate the measured data relating x to t.

t	x
0	5
1	7
2	10
3	12
6	15

A straight line fit will have the form

$$x = a_1 + a_2 t,$$

where a_1 and a_2 are the terms to be determined. Substituting each data point to the assumed form, we get five equations in two unknowns:

$$\begin{aligned} 5 &= a_1 + 0a_2, \\ 7 &= a_1 + 1a_2, \\ 10 &= a_1 + 2a_2, \\ 12 &= a_1 + 3a_2, \\ 15 &= a_1 + 6a_2. \end{aligned}$$

This is an overconstrained problem, and there is no unique solution that satisfies all of the equations! If a unique solution existed, then the curve fit would be perfect. However, there does exist a solution which minimizes the error, as is often proved in linear algebra textbooks (and will not be proved here). The procedure is straightforward. Rearranging, we get

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 6 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 5 \\ 7 \\ 10 \\ 12 \\ 15 \end{pmatrix}.$$

This is of the form $\mathbf{A} \cdot \mathbf{a} = \mathbf{b}$. We then find

$$\begin{aligned} \mathbf{A}^T \cdot \mathbf{A} \cdot \mathbf{a} &= \mathbf{A}^T \cdot \mathbf{b}, \\ \mathbf{a} &= \left(\mathbf{A}^T \cdot \mathbf{A}\right)^{-1} \cdot \mathbf{A}^T \cdot \mathbf{b}. \end{aligned}$$

Substituting, we find that

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \left[\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 6 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 6 \end{pmatrix} \right]^{-1} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 6 \end{pmatrix} \begin{pmatrix} 5 \\ 7 \\ 10 \\ 12 \\ 15 \end{pmatrix} = \begin{pmatrix} 5.7925 \\ 1.6698 \end{pmatrix}.$$

So the best fit estimate is

$$x = 5.7925 + 1.6698\ t.$$

The least squares error is $||\mathbf{A} \cdot \mathbf{a} - \mathbf{b}||_2 = 1.9206$. This represents what is known as the ℓ_2 error norm of the prediction. In matlab, this is found by the command $norm(A * a - b)$ where A, a, and b are the coefficient matrix $\mathbf{A}$, the solution $\mathbf{a}$ and the input vector $\mathbf{b}$, respectively. If the curve fit were perfect the error norm would be zero.

A plot of the raw data and the best fit straight line is shown in Figure 1

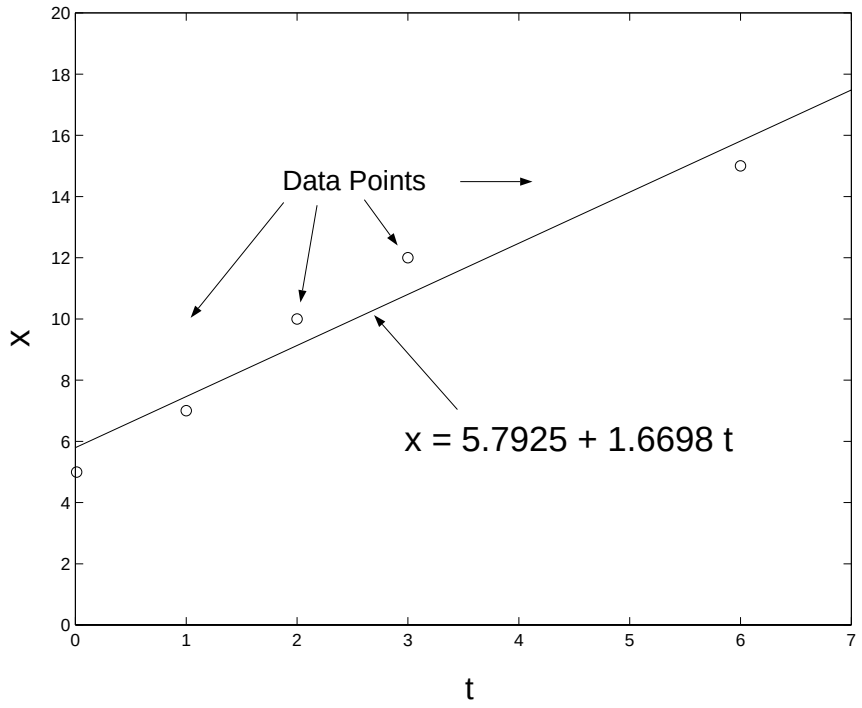

Figure 1: Plot of $x - t$ data and best least squares straight line fit.

Weighted least squares

If one has more confidence in some data points than others, one can define a weighting function to give more priority to those particular data points.

Example 0.2

Find the best straight line fit for the data in the previous example. Now however, assume that we have five times the confidence in the accuracy of the final two data points, relative to the other points. Define a square weighting matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 5 \end{pmatrix}.$$

Now we perform the following operations:

$$\begin{aligned} \mathbf{A}\cdot\mathbf{a} &= \mathbf{b}, \\ \mathbf{W}\cdot\mathbf{A}\cdot\mathbf{a} &= \mathbf{W}\cdot\mathbf{b}, \\ (\mathbf{W}\cdot\mathbf{A})^T\cdot\mathbf{W}\cdot\mathbf{A}\cdot\mathbf{a} &= (\mathbf{W}\cdot\mathbf{A})^T\cdot\mathbf{W}\cdot\mathbf{b}, \\ \mathbf{a} &= \left((\mathbf{W}\cdot\mathbf{A})^T\cdot\mathbf{W}\cdot\mathbf{A}\right)^{-1}(\mathbf{W}\cdot\mathbf{A})^T\cdot\mathbf{W}\cdot\mathbf{b}. \end{aligned}$$

With the above values of $\mathbf{W}$, direct substitution leads to

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 8.0008 \\ 1.1972 \end{pmatrix}.$$

So the best weighted least squares fit is

$$x = 8.0008 + 1.1972\ t.$$

A plot of the raw data and the best fit straight line is shown in Figure 2

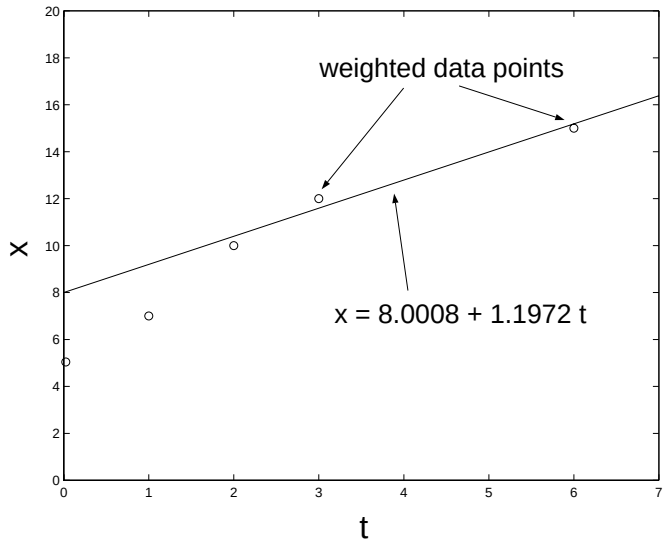

Figure 2: Plot of $x - t$ data and best weighted least squares straight line fit.

When the measurements are independent and equally reliable, $\mathbf{W}$ is the identity matrix. If the measurements are independent but not equally reliable, $\mathbf{W}$ is at most diagonal. If the measurements are not independent, then non-zero terms can appear off the diagonal in $\mathbf{W}$. It is often advantageous, for instance in problems in which one wants to control a process in real time, to give priority to recent data estimates over old data estimates and to continually employ a least squares technique to estimate future system behavior. The previous example does just that. A famous fast algorithm for such problems is known as a *Kalman Filter*.

Power law/logrithmic curve fits

It is extremely common and useful at times to fit data to either a power law form, especially when the data range over wide orders of magnitude. For clean units, it is highly advisable to scale both x and t by characteristic values. Sometimes this is obvious, and sometimes it is not. Whatever the case, the following form can usually be found

$$\frac{x(t)}{x_c} = a_1 \left(\frac{t}{t_c}\right)^{a_2}.$$

Here x is a dependent variable, t is an independent variable, x_c is a characteristic value of x (perhaps its maximum), and t_c is a characteristic value of t (perhaps its maximum), and a_1 and a_2 are curve fit parameters. This fit is not linear in the coefficients, but can be rendered so by taking the logarithm of both sides to get

$$\ln\left(\frac{x(t)}{x_c}\right) = \ln\left(a_1 \left(\frac{t}{t_c}\right)^{a_2}\right) = \ln(a_1) + a_2 \ln\left(\frac{t}{t_c}\right).$$

Often times one must not include values at $t = 0$ because of the logrithmic singularity there.

Example 0.3

An experiment yields the following data:

$t(s)$	$x(nm)$
0.0	0.0
1×10^{-3}	1×10^{0}
1×10^{-2}	5×10^{1}
1×10^{0}	3×10^{5}
1×10^{1}	7×10^{9}
1×10^{2}	8×10^{10}

A plot of the raw data is shown in Figure 3. Notice that the linear plot obscures the data at very small time, while the log-log plot makes the trends more clear. Now to get a curve fit for the log-log plot, we assume a power law form. We first eliminate the point at the origin, then scale the data, in this case by the maximum values of t and x, and take appropriate logarithms to get

$t(s)$	$x(nm)$	t/t_{max}	x/x_{max}	$\ln\left(\frac{t}{t_{max}}\right)$	$\ln\left(\frac{x}{x_{max}}\right)$
1×10^{-3}	1×10^{0}	1×10^{-5}	1.25×10^{-11}	-11.5129	-25.1053
1×10^{-2}	5×10^{1}	1×10^{-4}	6.25×10^{-10}	-9.2013	-21.1933
1×10^{0}	3×10^{5}	1×10^{-2}	3.75×10^{-6}	-4.6052	-12.4938
1×10^{1}	7×10^{9}	1×10^{-1}	8.75×10^{-2}	-2.3026	-2.4361
1×10^{2}	8×10^{10}	1×10^{0}	1×10^{0}	0.0000	0.0000

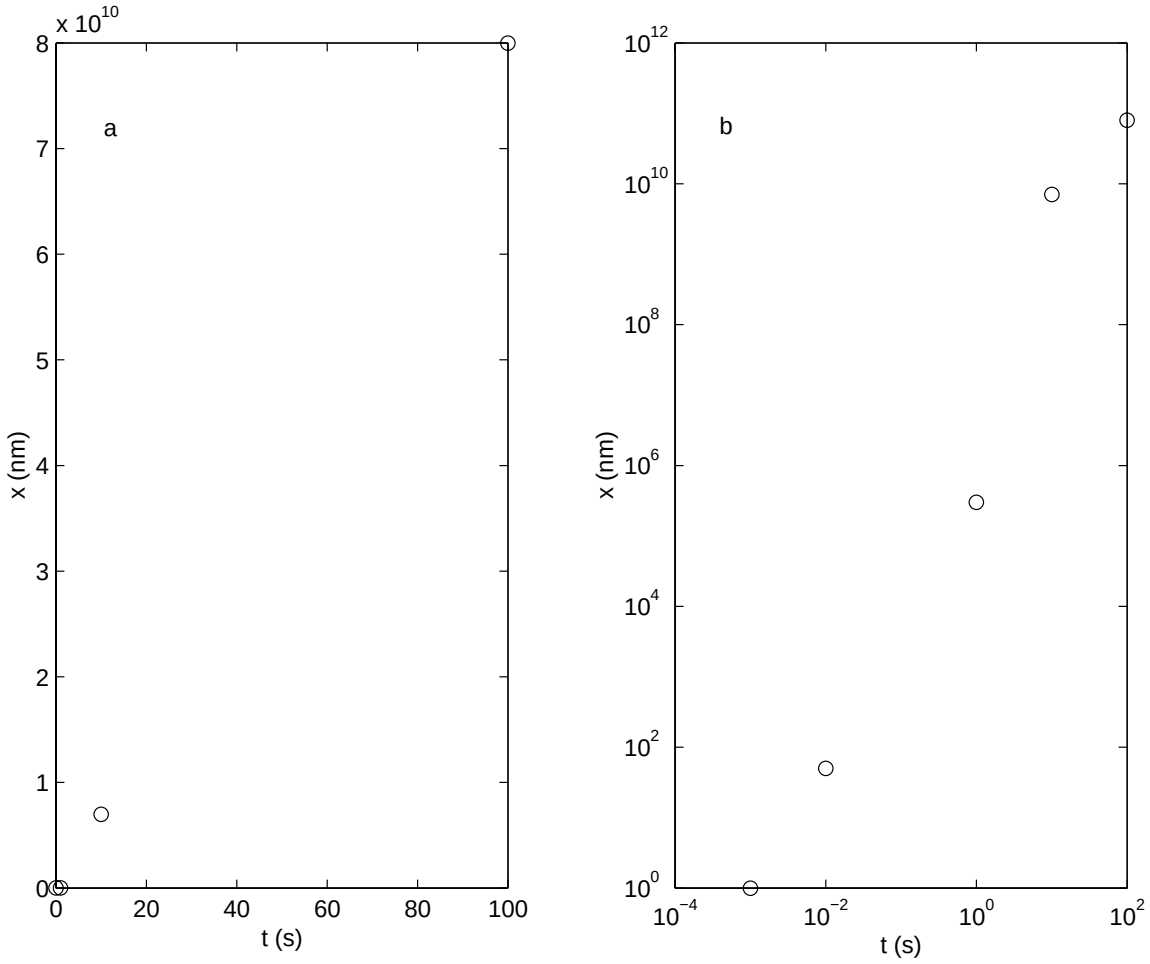

Figure 3: Plot of $x - t$ data in a) linear and b) log-log plots.

Now we prepare the system of linear equations to solve

$$\begin{aligned}
\ln\left(\frac{x}{x_{max}}\right) &= \ln a_1 + a_2 \ln\left(\frac{t}{t_{max}}\right), \\
-25.1053 &= \ln a_1 + a_2(-11.5129), \\
-21.1933 &= \ln a_1 + a_2(-9.2013), \\
-12.4938 &= \ln a_1 + a_2(-4.6052), \\
-2.4361 &= \ln a_1 + a_2(-2.3026), \\
0.0000 &= \ln a_1 + a_2(0.0000),
\end{aligned}$$

In matrix form, this becomes

$$\begin{pmatrix} 1 & -11.5129 \\ 1 & -9.2013 \\ 1 & -4.6052 \\ 1 & -2.3026 \\ 1 & -0.0000 \end{pmatrix} \begin{pmatrix} \ln a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} -25.1053 \\ -21.1933 \\ -12.4938 \\ -2.4361 \\ 0.0000 \end{pmatrix}.$$

This is of the form

$$\mathbf{A} \cdot \mathbf{a} = \mathbf{b}.$$

As before, we multiply both sides by $\mathbf{A}^T$ and then solve for $\mathbf{a}$, we get

$$\mathbf{a} = \left(\mathbf{A}^T \cdot \mathbf{A}\right)^{-1} \cdot \mathbf{A}^T \cdot \mathbf{b}.$$

Solving, we find

$$\mathbf{a} = \begin{pmatrix} 0.4206 \\ 2.2920 \end{pmatrix}.$$

So that

$$\ln a_1 = 0.4206, \qquad a_2 = 2.2920.$$

or

$$a_1 = 1.5228.$$

So the power law curve fit is

$$\frac{x(t)}{8.000 \times 10^{10}\ nm} = 1.5228 \left(\frac{t}{100\ s}\right)^{2.2920},$$

or

$$x(t) = \left(1.2183 \times 10^{11}\ nm\right) \left(\frac{t}{100\ s}\right)^{2.2920}.$$

A plot of the raw data and curve fit is shown in Figure 4.

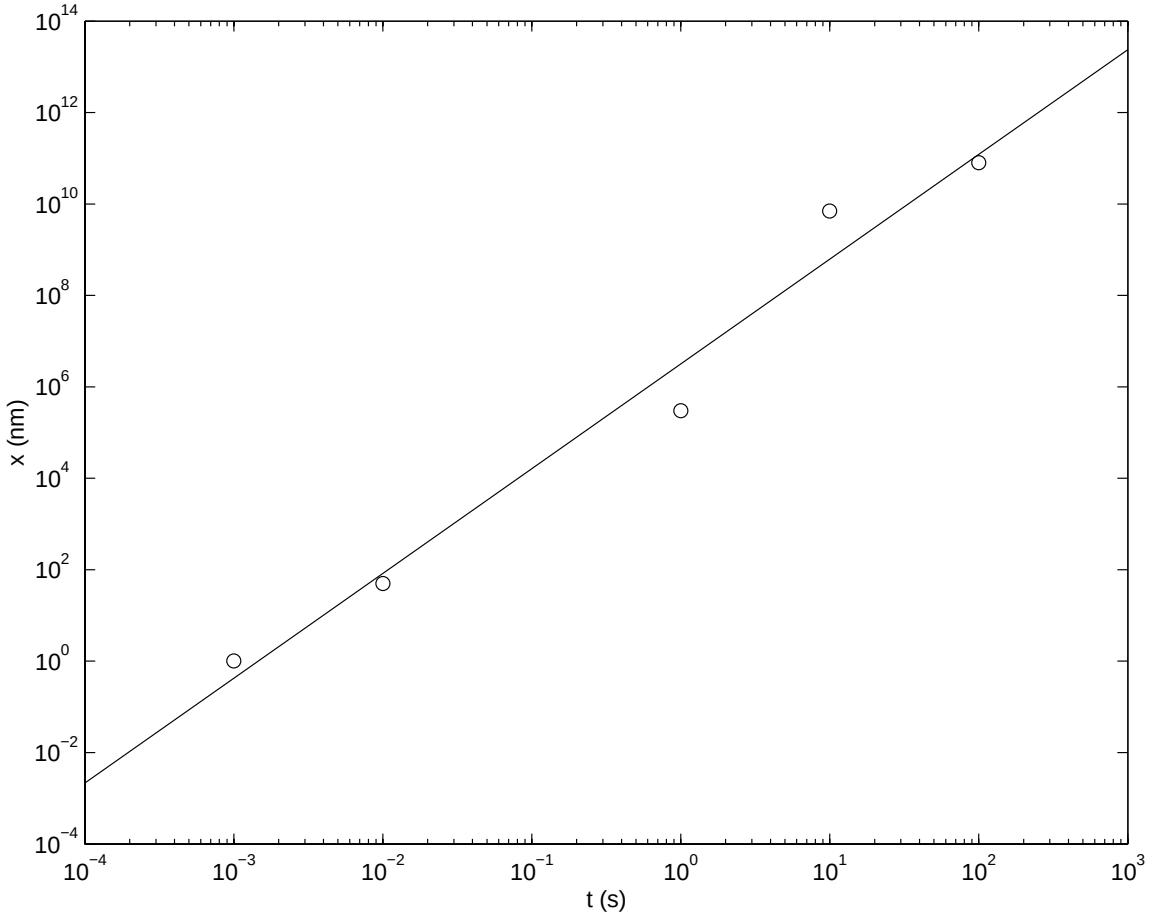

Figure 4: Plot of $x-t$ data and power law curve fit: $x(t) = \left(1.2183 \times 10^{11}\ nm\right) \left(\frac{t}{100\ s}\right)^{2.2920}$.

Higher order curve fits

As long as the assumed form for the curve fit is linear in the coefficients, it is straightforward to extend to high order curve fits as demonstrated in the following example.

Example 0.4

An experiment yields the following data:

t	x
0.0	1.0
0.7	1.6
0.9	1.8
1.5	2.0
2.6	1.5
3.0	1.1

Find the least squares best fit coefficients a_1, a_2, and a_3 if the assumed functional form is

1. $x = a_1 + a_2 t + a_3 t^2$
2. $x = a_1 + a_2 \sin\left(\frac{t}{6}\right) + a_3 \sin\left(\frac{t}{3}\right)$

Plot on a single graph the data points and the two best fit estimates. Which best fit estimate has the smallest least squares error?

- $x = a_1 + a_2 t + a_3 t^2$

We substitute each data point into the assumed form and get the following set of linear equations

$$\begin{aligned} 1.0 &= a_1 + a_2(0.0) + a_3(0.0)^2, \\ 1.6 &= a_1 + a_2(0.7) + a_3(0.7)^2, \\ 1.8 &= a_1 + a_2(0.9) + a_3(0.9)^2, \\ 2.0 &= a_1 + a_2(1.5) + a_3(1.5)^2, \\ 1.5 &= a_1 + a_2(2.6) + a_3(2.6)^2, \\ 1.1 &= a_1 + a_2(3.0.0) + a_3(3.0)^2, \end{aligned}$$

This can be rewritten as

$$\begin{pmatrix} 1 & 0.0 & 0.0 \\ 1 & 0.7 & 0.49 \\ 1 & 0.9 & 0.81 \\ 1 & 1.5 & 2.25 \\ 1 & 2.6 & 6.76 \\ 1 & 3.0 & 9.00 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 1.0 \\ 1.6 \\ 1.8 \\ 2.0 \\ 1.5 \\ 1.1 \end{pmatrix}$$

This is of the form

$$\mathbf{A} \cdot \mathbf{a} = \mathbf{b}.$$

As before, we multiply both sides by $\mathbf{A}^T$ and then solve for $\mathbf{a}$, we get

$$\mathbf{a} = \left(\mathbf{A}^T \cdot \mathbf{A}\right)^{-1} \cdot \mathbf{A}^T \cdot \mathbf{b}.$$

Solving, we find

$$\mathbf{a} = \begin{pmatrix} 0.9778 \\ 1.2679 \\ -0.4090 \end{pmatrix}.$$

So the best quadratic curve fit to the data is

$$x(t) \sim 0.9778 + 1.2679t - 0.4090t^2.$$

The least squares error norm is

$$||\mathbf{A} \cdot \mathbf{a} - \mathbf{x}||_2 = 0.0812.$$

- $x = a_1 + a_2 \sin\left(\frac{t}{6}\right) + a_3 \sin\left(\frac{t}{3}\right)$

This form has applied a bit of intuition. The curve looks like a sine wave of wavelength 6 which has been transposed. So we suppose it is of the form. The term a_1 is the transposition; the term on a_2 is the fundamental frequency which fits in the domain; the term on a_3 is the first harmonic, which we have thrown in for good measure.

We substitute each data point into the assumed form and get the following set of linear equations

$$\begin{aligned}
1.0 &= a_1 + a_2 \sin\left(\frac{0.0}{6}\right) + a_3 \sin\left(\frac{(0.0)}{3}\right), \\
1.6 &= a_1 + a_2 \sin\left(\frac{0.7}{6}\right) + a_3 \sin\left(\frac{(0.7)}{3}\right), \\
1.8 &= a_1 + a_2 \sin\left(\frac{0.9}{6}\right) + a_3 \sin\left(\frac{(0.9)}{3}\right), \\
2.0 &= a_1 + a_2 \sin\left(\frac{1.5}{6}\right) + a_3 \sin\left(\frac{(1.5)}{3}\right), \\
1.5 &= a_1 + a_2 \sin\left(\frac{2.6}{6}\right) + a_3 \sin\left(\frac{(2.6)}{3}\right), \\
1.1 &= a_1 + a_2 \sin\left(\frac{3.0}{6}\right) + a_3 \sin\left(\frac{(3.0)}{3}\right).
\end{aligned}$$

This can be rewritten as

$$\begin{pmatrix} 1 & 0.0 & 0.0 \\ 1 & 0.1164 & 0.2312 \\ 1 & 0.1494 & 0.2955 \\ 1 & 0.2474 & 0.4794 \\ 1 & 0.4199 & 0.7622 \\ 1 & 0.4794 & 0.8415 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 1.0 \\ 1.6 \\ 1.8 \\ 2.0 \\ 1.5 \\ 1.1 \end{pmatrix}$$

This is of the form

$$\mathbf{A} \cdot \mathbf{a} = \mathbf{b}.$$

As before, we multiply both sides by $\mathbf{A}^T$ and then solve for $\mathbf{a}$, we get

$$\mathbf{a} = \left(\mathbf{A}^T \cdot \mathbf{A}\right)^{-1} \cdot \mathbf{A}^T \cdot \mathbf{b}.$$

Solving, we find

$$\mathbf{a} = \begin{pmatrix} 1.0296 \\ -37.1423 \\ 21.1848 \end{pmatrix}.$$

So the best curve fit of the form is

$$x(t) \sim 1.0116 - 37.1423 \sin\left(\frac{t}{6}\right) + 21.1848 \sin\left(\frac{t}{3}\right)$$

The least squares error norm is

$$||\mathbf{A} \cdot \mathbf{a} - \mathbf{x}||_2 = 0.1165.$$

A plot of the raw data and the two best fit curves is shown in Figure 5

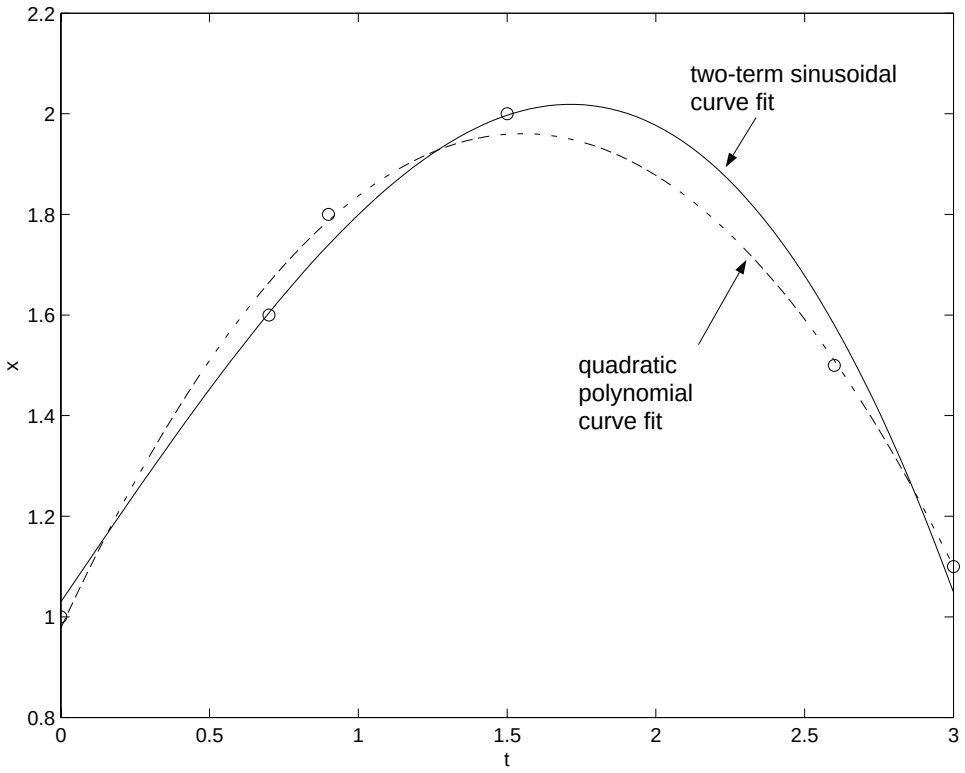

Figure 5: Plot of $x-t$ data and two least squares curve fits $x(t) \sim 0.9778+1.2679t^2-0.4090t^2$, and $x(t) \sim 1.0116 - 37.1423 \sin\left(\frac{t}{6}\right) + 21.1848 \sin\left(\frac{t}{3}\right)$.