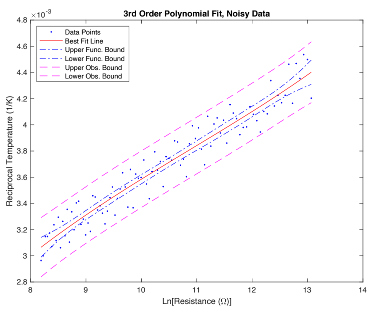
E80

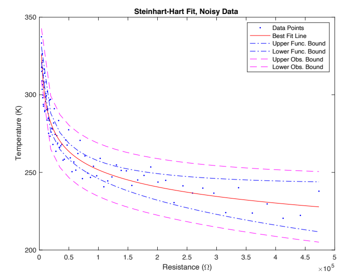
Experimental
Engineering



$$F = F\left(a_0, \cdots, a_n, x\right) = a_0 f_0(x) + a_1 f_1(x) + \cdots + a_n f_n(x)$$

```
f3 =

     Linear model Poly3:
     f3(x) = p1*x^3 + p2*x^2 + p3*x + p4
     Coefficients (with 95% confidence bounds):
       p1 =   3.053e-06  (-6.602e-06, 1.271e-05)
       p2 =  -9.963e-05  (-0.0004061, 0.0002069)
       p3 =    0.001339  (-0.001874, 0.004552)
       p4 =   -0.002894  (-0.01401, 0.008224)
```



Non-Linear Regression
Part of: Error Analysis

1

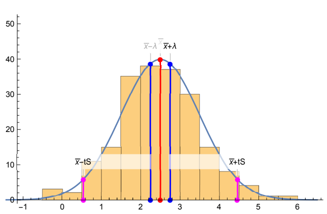
E80

Experimental
Engineering

Review

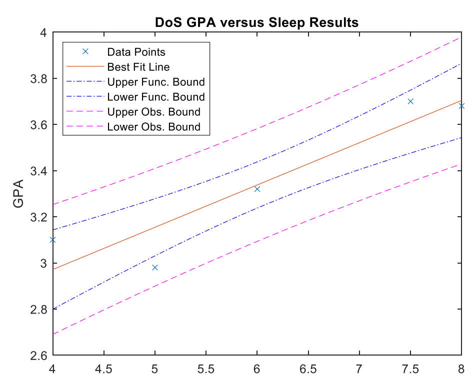
$$\overline{x} = \frac{\sum_{i=1}^{N} x_i}{N} \approx \mu$$

$$\lambda = tS_{\overline{x}} = \frac{tS}{\sqrt{N}}$$



$$\hat{\beta}_1 = \frac{\sum_{i=1}^{N}\left(x_i - \overline{x}\right)\left(y_i - \overline{y}\right)}{\sum_{i=1}^{N}\left(x_i - \overline{x}\right)^2}$$

$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}$$



2

E80
Experimental
Engineering

General Linear Fit

Linear in parameters, a_0, ..., a_n

$$F = F\left(a_0, \cdots, a_n, x\right) = a_0 f_0(x) + a_1 f_1(x) + \cdots + a_n f_n(x)$$

Example: Polynomial

$$P_n(x) = a_0 \cdot 1 + a_1 \cdot x + a_1 \cdot x^2 + \cdots + a_n \cdot x^n, f_i = x^i$$

Example: Fourier series

$$F_n(t) = a_{-n} \cdot e^{-jn\omega_0 t} + \cdots + a_0 \cdot 1 + a_1 \cdot e^{j\omega_0 t} + \cdots + a_n \cdot e^{jn\omega_0 t}, f_i = e^{ji\omega_0 t}$$

Example: Composite

$$F_n(x) = a_0 \cdot 1 + a_1 \cdot x + a_2 \cos(x) + a_3 \ln(x)$$

3

E80
Experimental
Engineering

General Nonlinear Fit

In general, not linear in parameters, a_0, ..., a_n

$$F = F\left(a_0, \cdots, a_n, x\right)$$

Example: linear in a, not linear in b.

$$y = ax^b$$

Example: linear in a, not linear in t_0 or τ.

$$y = a e^{-\frac{t-t_0}{\tau}}$$

4

E80 Experimental Engineering

Least Squares

Data set: $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$

Fitting function: $F = F(a_0, \cdots, a_n, x)$

Residuals: $e_i = y_i - F(a_0, \cdots, a_n, x_i)$

Minimize $SSE = \sum e_i^2$ by varying $(a_0, \cdots, a_n)$

5

E80 Experimental Engineering

MATLAB ‘`fit`’

Syntax

`[fitobject,gof] = fit(x,y,fitType)`

`fitType` is the function to fit.

Built in functions include:

- Polynomials: ‘`poly1`’ through ‘`poly9`’
- Exponentials: ‘`exp1`’
- Fourier Series: ‘`fourier1`’ through ‘`fourier9`’
- Power expressions: ‘`power1`’

User Defined Function ‘`fittype`’

6

E80
Experimental Engineering

Syntax of ‘fittype’

Linear Function

```
    aFittype = fittype(linearModelTerms)
```
e.g.,
```
    ft = fittype({'x','sin(x)','1’})
```
generates
```
    ft(a,b,c,x) = a*x + b*sin(x) + c
```

7

E80
Experimental Engineering

Syntax of ‘fittype’

Nonlinear Function

```
    aFittype = fittype(expression,Name,Value)
```
e.g.,
```
    g = fittype(‘a*u+b*exp(c*u)’,…
              'independent','u’)
```
generates
```
    g(a,b,n,u) = a*u+b*exp(c*u)
```
where u is the independent variable.

8

E80
Experimental Engineering

‘predint’

```
ci = predint(fitresult,x,level,intopt,simopt)

e.g.,

p11 = predint(fitresult,x,0.95,'observation','off');
```

9

E80
Experimental Engineering

Steinhart-Hart Fitting of TCS10K5 Data

August, 2013

WAVELENGTH ELECTRONICS

TCS10K5

10 kΩ NTC Cylindrical Head Thermistor

Pb RoHS Compliant

Graphic enlarged for detail

TCS10K5 NTC THERMISTOR

GENERAL DESCRIPTION:
This ±1% thermistor is encapsulated in a polyimide tube, for assemblies where surface mounting or embedding the thermistor is required. Ideal for tight mounting spaces with 38 AWG nickel bifilar leads and a diameter of 0.5 mm by 3 mm.

Thermal Resistance or Dissipation Constant is 0.2 mW / °C.

Thermal Time Constant is 200 mSec.

FEATURES:
- Low Cost
- Ideal for Optical or Thin Surfaces & Small Laser Packages
- 1% Tolerance
- 3” Long Nickel Bifilar Leads
- Isolated Leads Provide Isolation from Metal Housing
- RoHS Compliant

10

E80
Experimental Engineering

Steinhart-Hart Equation

Standard form:

$$T = \frac{1}{A + B\ln(R) + C\left[\ln(R)\right]^2 + D\left[\ln(R)\right]^3}$$

As power series in $\ln(R)$:

$$1/T = A_0 + A_1\ln(R) + A_2\left[\ln(R)\right]^2 + A_3\left[\ln(R)\right]^3 + \cdots$$

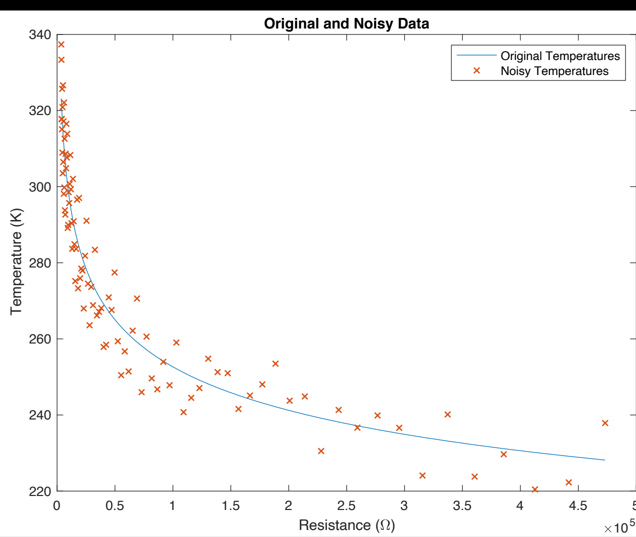
Quadratic term, $C(\ln R)^2$, usually left out:

$$T = \frac{1}{A + B\ln(R) + D\left[\ln(R)\right]^3}$$

11

E80
Experimental Engineering

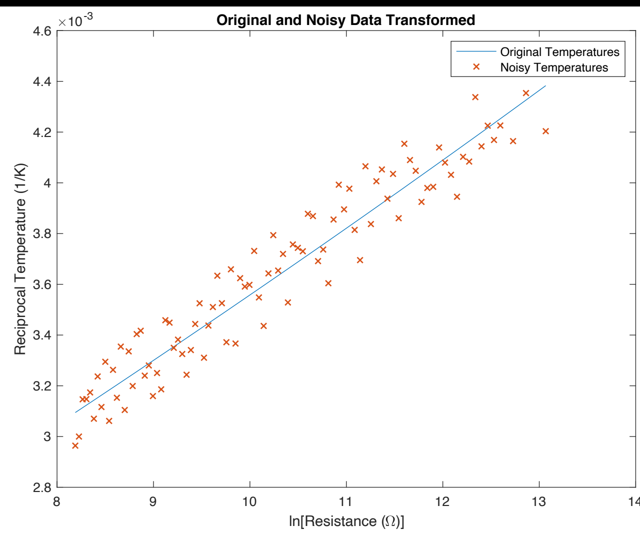
Original and Noisy Data



12

E80
Experimental Engineering

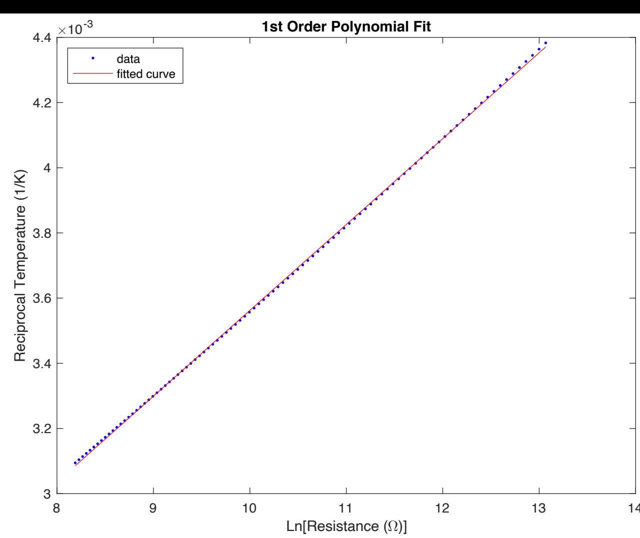
Transformed Data



13

E80
Experimental Engineering

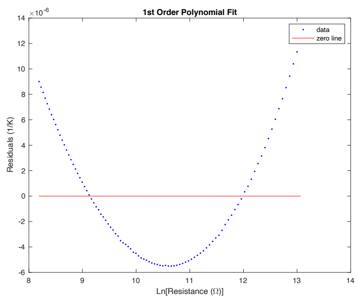
Linear (1^{st}-Order) Fit

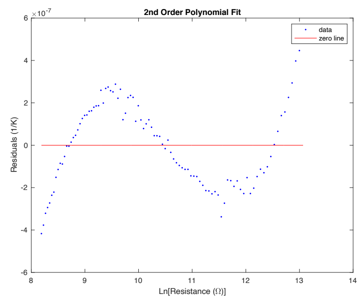


14

E80
Experimental
Engineering

Residuals

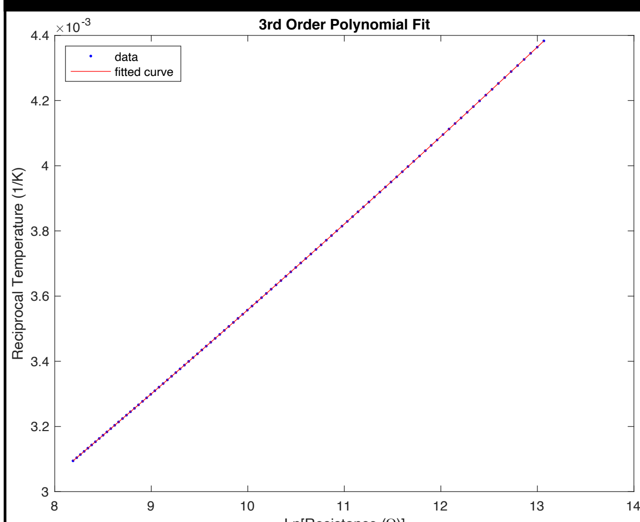




15

E80
Experimental
Engineering

3rd-Order Fit



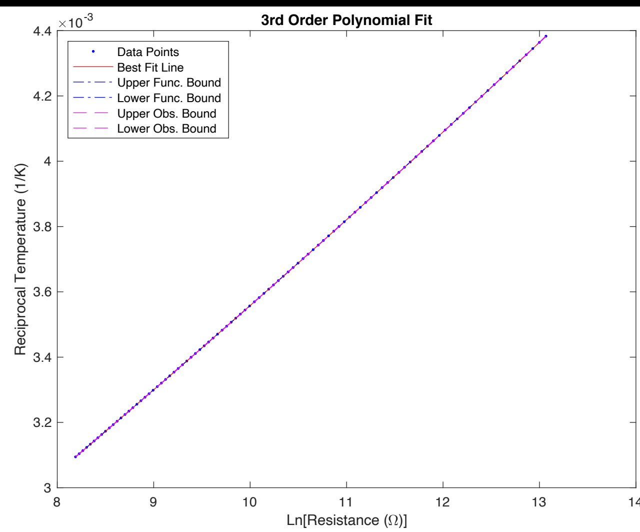
f3 =

Linear model Poly3:
f3(x) = p1*x^3 + p2*x^2 + p3*x + p4
Coefficients (with 95% confidence bounds):
p1 = 8.634e-08 (8.341e-08, 8.927e-08)
p2 = -3.558e-08 (-1.285e-07, 5.735e-08)
p3 = 0.0002351 (0.0002342, 0.0002361)
p4 = 0.001124 (0.001121, 0.001127)

16

E80
Experimental
Engineering

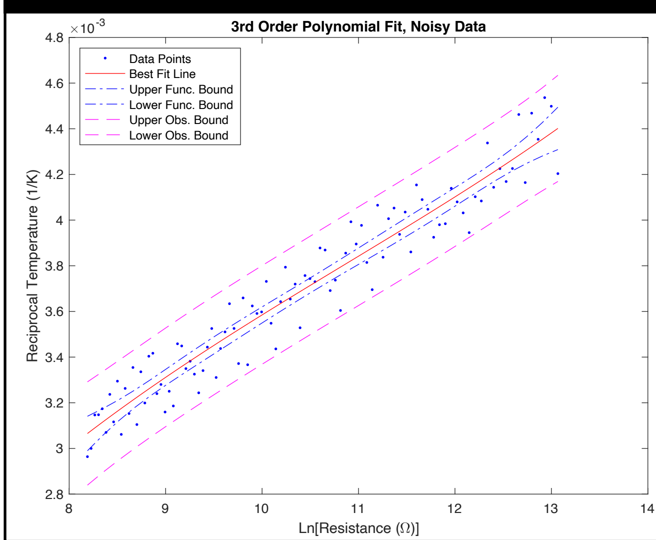
Functional & Observational Bounds



17

E80
Experimental
Engineering

3rd-Order With Noise in Data



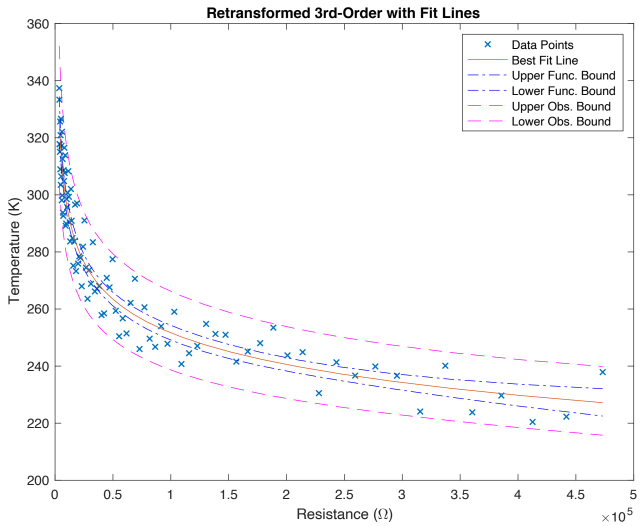

```
f3 =

     Linear model Poly3:
     f3(x) = p1*x^3 + p2*x^2 + p3*x + p4
     Coefficients (with 95% confidence bounds):
       p1 =   3.053e-06  (-6.602e-06, 1.271e-05)
       p2 =  -9.963e-05  (-0.0004061, 0.0002069)
       p3 =    0.001339  (-0.001874, 0.004552)
       p4 =   -0.002894  (-0.01401, 0.008224)
```

18

E80
Experimental
Engineering

Transformed Back



19

E80
Experimental
Engineering

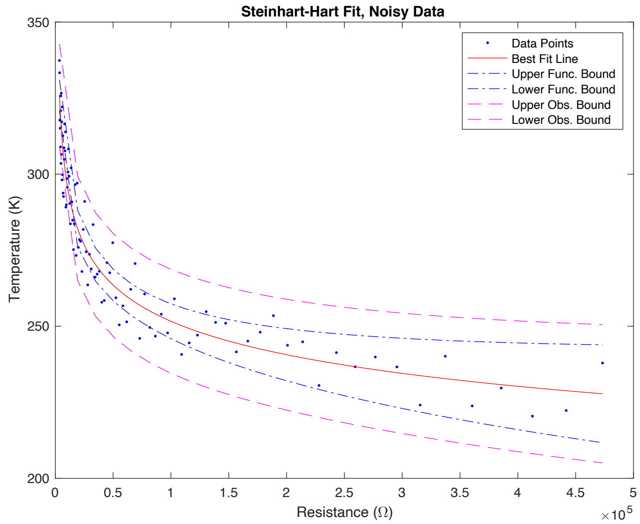
Nonlinear Fit

```
% First we have to define the function we will fit.
% Things work better if we have starting points for a, b, c, and d. We'll
% use our values from above and 'fitoptions'
fo = fitoptions('Method','NonlinearLeastSquares',...
    'StartPoint',[-0.002894 0.001339 -9.963e-05 3.053e-06]);
ft = fittype('1/(a+b*log(R)+c*(log(R)^2)+d*(log(R)^3))','independent',...
    'R','options',fo);
% Next, we have to get our data into the correct format for 'fit'.
[Xout,Yout] = prepareCurveData(R, TN);
% Now we'll do our fit.
[f4,stat4] = fit(Xout,Yout,ft)
```

20

E80
Experimental Engineering

Steinhart-Hart Nonlinear Fit



f4 =

General model:
f4(R) =
1/(a+b*log(R)+c*(log(R)^2)+d*(log(R)^3))
Coefficients (with 95% confidence bounds):
a = -0.001534 (-0.01282, 0.009757)
b = 0.0009416 (-0.002388, 0.004271)
c = -6.146e-05 (-0.0003868, 0.0002639)
d = 1.844e-06 (-8.756e-06, 1.244e-05)

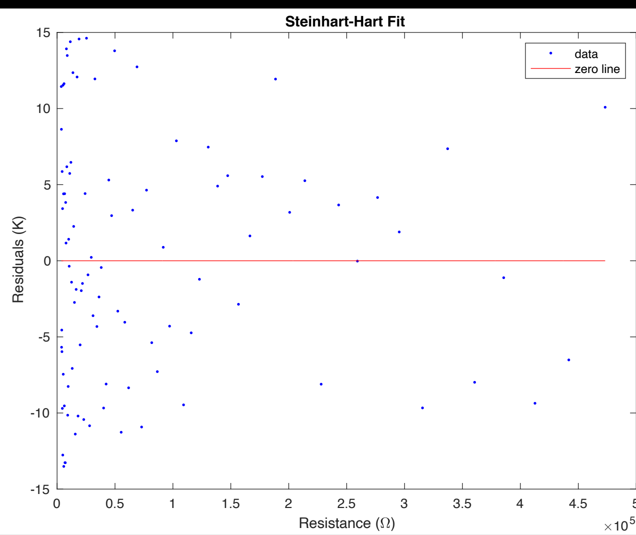
f3 =

Linear model Poly3:
f3(x) = p1*x^3 + p2*x^2 + p3*x + p4
Coefficients (with 95% confidence bounds):
p1 = 3.053e-06 (-6.602e-06, 1.271e-05)
p2 = -9.963e-05 (-0.0004061, 0.0002069)
p3 = 0.001339 (-0.001874, 0.004552)
p4 = -0.002894 (-0.01401, 0.008224)

21

E80
Experimental Engineering

Steinhart-Hart Residuals



22

E80
Experimental
Engineering

Takeaways

1. Nonlinear data fitting uses the same terms with the same meanings as fitting a line does.
2. It’s much easier to have the fitting routines do the work for you.
3. Choose the simplest model that has random-looking residuals.
4. When reporting the results of a nonlinear fit, plot the data, the fit, the functional bounds, and possibly the observational bounds, as well as the confidence intervals on the fitted parameters.