

Ribbon Formation for Electrical Interconnection

Mentors:

J. Michael Gray and Robert Shimpf, Medtronic Inc.

Kapil Ahuja, Virginia Polytechnic Institute and State University

Yuan Dong, Northern Illinois University

Huaiying Gu, University of Michigan

Shiyuan Gu, Louisiana State University

Jon Van Laarhoven, University of Iowa

Jia Wei, Texas A & M University

Aug 15, 2008

Abstract

Connecting components is ubiquitous to all implantable medical devices. The Laser Ribbon Bonding (LRB) process used at Medtronic has four major steps: first weld, loop, second weld, and termination. Currently, no model exists for the looping process that could provide insight to the design engineer and reliability engineer on how particular input parameters affect the loop. This project developed two empirical models and a solver for finding the optimal loop shape. The empirical models consist of a geometric model focussing on the highest point of the loop and the loop length, and a shape model that is based on curvature. The solver for optimal loop shape combines the geometric model and the curvature model to find the ideal input parameters corresponding to minimum curvature and maximum length (with some weighting factors for both).

Keywords: wire bonding, interconnect ribbon, curvature, spline interpolation, regression analysis, optimization

1 Introduction

Some electrical interconnections in medical devices with powered circuitry are made by forming and welding piece of thin flat ribbon (or wire) between two electrical terminals(2-30 per device) [5]. Medtronic has developed a patented process called Laser Ribbon (LRB) to create these interconnections. The LRB process has four major steps: 1st Weld (ribbon is pressed against 1st terminal and laser energy applied to create weld between ribbon and 1st terminal); looping (machine head moves to form ribbon into loop); 2nd Weld (similar to 1stweld) and termination (machine head moves to tear off ribbon).

An implanted device is exposed to external loading that cause relative motion between interconnected components. Without a properly formed loop, the interconnection can fatigue over time due to these motions. Looping requirements for each device model are unique due to differences in features and geometry. Manufacturing variability means that even within the same model, loops vary slightly from device to device. Due to large patient base, millions of interconnections are placed into the field each year. Failure of even a single interconnection in the field can be catastrophic to a device, patient, and the company. For these reasons, looping is very important,so in this project we focus on the looping.

Current equipment for forming the ribbon allows for a virtually an infinite set of motions between the two terminals to be programmed. Currently the only method for determining what the resulting shape of the ribbon will be from a set of machine motions is to program the machine, form a ribbon, visually observe the resultant shape, and iterate until the “desired shape” is obtained. The problems proposed are:

1. Given some data regarding the ribbon shapes that result from a very limited set of tool motions, can a more general model be developed that can predict the shape of the loop based on the machine motions.
2. Can this model be improved by incorporating the material response behavior of the ribbon or other physical relationships that govern ribbon formation.
3. Can this model be inverted so that if a particular ribbon shape is desired, a corresponding set of machine parameters can be identified.
4. If only the spacing, positioning, and clearance around two terminals are known, can an optimal shape be identified that minimizes the stress induced in the ribbon from relative motion between the terminals while avoiding interference with any of the surrounding geometric constraints.

This project targeted 1 and 4 above with some literature review for 2. The report is organized as follows. Section 2 covers the fixes done with the provided data. In Section 3, empirical models developed are described. Empirical models were developed in two steps i.e. first was identification of output parameters and second was finding a mapping between the input and output parameters. Section 4 discusses how to use the empirical model to find the input parameters that describe a optimum loop shape. Finally, Section 5 gives a brief review of how different industries model the wire bonding process keeping the material response behavior in perspective.

2 Data Fix

It was nice to receive some experimental data, however the data received had couple of issues. Essentially it was inconsistent in the reference coordinates and view angles. Following is a detailed split-up of the problems that were fixed in the data:

- Some of the data starts from the first weld, and some from the second weld with no tracking.
- Three experiments with different numbering and missing data.
- Unsynchronized step and span between the master file and the data files.

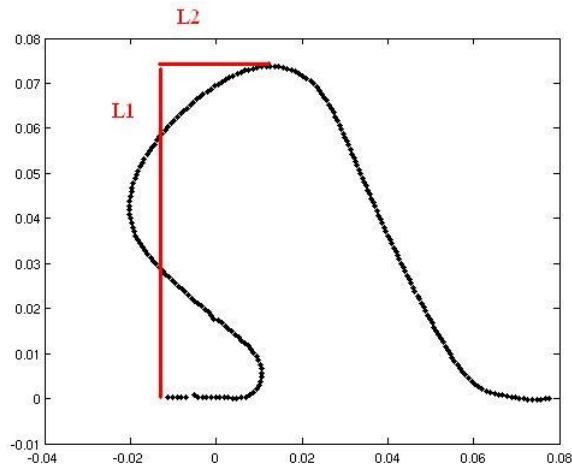


Figure 1: A geometric model.

- Noise at start and end.

The important aspect to highlight here is that in real world engineering these things do happen.

3 Empirical Models

The experimental data provided consisted of machine settings (input parameters), and the resultant loop as discrete data points (output parameters). Thus, to build the empirical model the first step was to identify what are the most important output parameters that the engineers are concerned about. The next step was to find the a mapping between the input parameters (provided) and the output parameters (identified).

3.1 Output Parameters

From Medtronic's perspective, most important output parameters are the ones related to characteristic of following three models of the loop: a geometric model, a shape or curvature model, and a complete model. The next three subsections touch upon how these models were computed.

3.1.1 Geometric Model

For a design engineer the most important aspect is the highest point of the loop (as shown in the Figure 1). X and Y coordinates corresponding to highest point on the loop give the value of $L1$ and $L2$ (as marked in the above mentioned figure). Length L is computed as the sum of Euclidean distance between all the data points of the loop.

3.1.2 Shape Model

Engineering experience shows that the reliability has strong correlation with curvature of the loop. Ideally, one would want to compute how much does the loop curve at points of high curvature. That

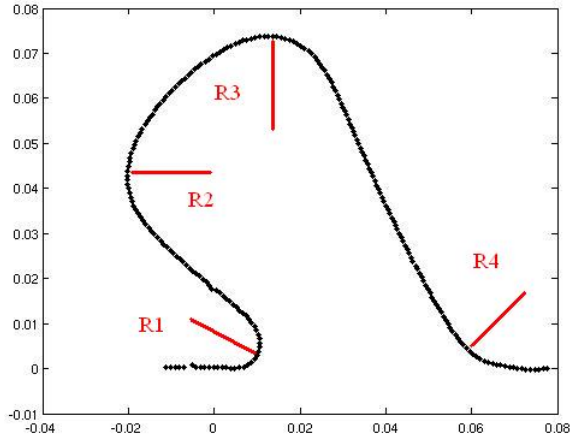


Figure 2: A shape model.

is, $R1$, $R2$, $R3$, and $R4$ as depicted in Figure 2, would form the most important output parameters for a reliability engineer. The next subsection discusses how to compute curvature, and the subsequent subsection talks of a alternative technique to compute $R1$, $R2$, $R3$, and $R4$ explicitly.

Computing the Curvature

To compute the curvature, following steps were followed:

- Preprocessing step: The original data consisted of 92 loops, each described by a set of discrete points sampled along the loop. Each loop was re-sampled to 50 points as this helped to smooth out some of the local fluctuations.
- Step 1: Splines were used to interpolate the parameter equation $r(s) = (x(s), y(s))$. Parameter s was chosen to be the arc-length which is approximated by the line segments of the original data points.
- Step 2: Curvature was computed as

$$\frac{|r'(s) \times r''(s)|}{\|r'(s)\|^3} \quad (1)$$

After curvature is obtained, all local maximum (peaks) of $\kappa(s)$ are identified. Note that computing the local maximum involves the third derivative of the curve which may not be a continuous function. Boeing package *Geoduck* provides us friendly routines to do the spline-interpolation and find the peaks in $\kappa(s)$. However, the curvature obtained by the above approach does not matched the engineering experience. Some good loops have greater maximum curvature than the bad loops. The logical reason being that curvature is too local to capture some useful global information. Thus, we take an average of the curvature of a region of arc-length l as follows:

$$\tilde{\kappa}(s) = \frac{1}{2l} \int_{-l}^l \kappa(s+t) dt \quad (2)$$

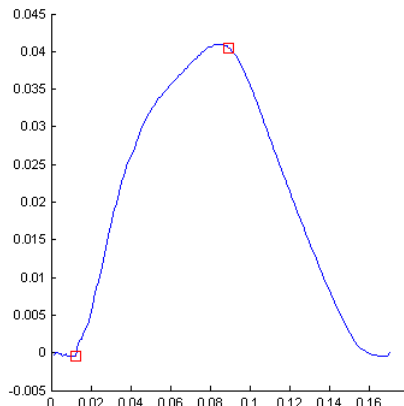


Figure 3: An Alternative Way of Computing all R.

Experiments were done on all 90 loops, taking $l = 10\%$ of the total length. All loops were manufacturing acceptable, but 5 of them are labeled "good" because they are more desirable than the others and 27 of them are labeled "bad" because they are less desirable than the others. The results are promising. All $\max_s \tilde{\kappa}(s)$ of good loops are strictly less than that of the bad loops. This is a quite strong evidence that $\max_s \tilde{\kappa}(s)$ (for a careful choice of l) is a useful measure for the reliability of a loop. In fact, we can think $\frac{1}{\max_s \tilde{\kappa}(s)}$ as a mathematical translation of the engineers' notion of $R1$. The experiment above shows this number is good enough for measuring the reliability of the 90 loops. However, we may prefer a more delicate measure for future need. A idea to do that is to combine the first peak in $\tilde{\kappa}(s)$ (that is, $\max_s \tilde{\kappa}(s)$) with the second, third and fourth peaks in $\tilde{\kappa}(s)$. Note that the reciprocal of the second, third and fourth peaks in $\tilde{\kappa}(s)$ may be the mathematical translations of $R2$, $R3$ and $R4$. However, this idea of combining several peaks has not been tested by experiment.

An Alternative Way of Computing all the R

To get $R1$, $R2$, $R3$, and $R4$ explicitly following approach was used:

- Find all the inflection points in the curve by computing the first derivative and checking where it changes sign.
- Break the loop along at these inflection points into three parts (as there are two inflection points per loop) .
- In each of the part, find the point where the second derivative is maximum (because we are looking for place where the curve curves the most i.e. there is large rate of change in the first derivative).

Figure 3 shows how the algorithm finds the points of highest curvature. This approach was not pursued further because it was very sensitive to noise

3.1.3 Complete Model

We seek output parameters to quantify the shape of a wire ribbon loop. One possibility is to divide the loop into several pieces and fit a polynomial to each of these pieces. The coefficients of the

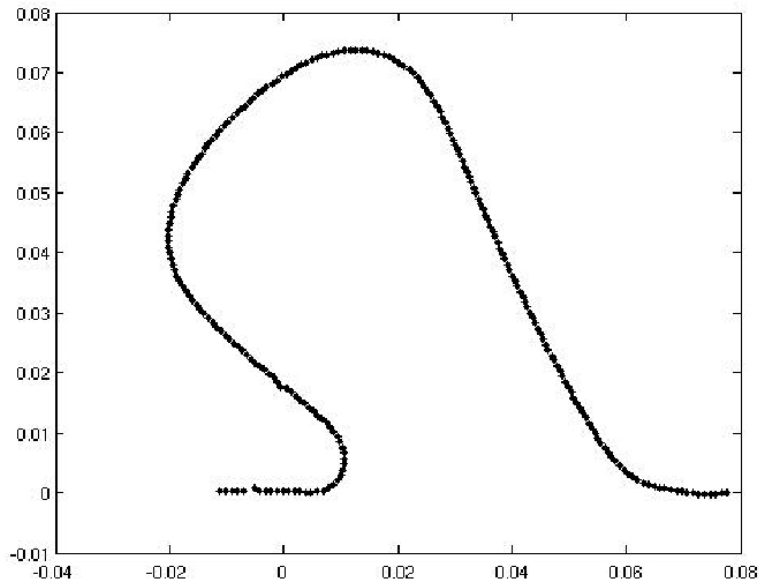


Figure 4: A typical loop configuration

polynomial are then used as the output parameters for the loop. We present two methods which parse the loop into two pieces and fit polynomials to each of these pieces. These methods differ in their application of boundary conditions.

Two Cubics

Let $X = \{x_k, y_k\}_{k=1}^n$ denote the set of data points comprising a loop. We break the loop into two parts as follows. Let i denote the argument which maximizes $\{y(i)\}$, this is the apex of the loop. Let $X_1 = \{x_k, y_k\}_{k=1}^i$, and $X_2 = \{x_k, y_k\}_{k=i}^n$ denote the left and right sides of the loop, respectively.

The right side of the wire loop is easily fitted with a polynomial. The set of points in X_2 defines a function for all data sets encountered. We use the “polyfit” function in Matlab to find the degree 3 polynomial which best fits the data in the least squares sense. In order to improve the fit of the polynomial to the shape of the curve, we remove all points from the right of the loop which are lower than 5% of the total loop height. Removing this “flat spot” corresponding with the second weld point improves the fit of the polynomial to the rest of the loop. Denote the degree three polynomial fitting the right hand data by $p_1x^3 + p_2x^2 + p_3x + p_4$. The four coefficients p_1, p_2, p_3 , and p_4 comprise our first output parameters.

The left side of the wire loop proved more challenging, as a number of the wire loops fold back on themselves (see Figure 4). Since the left portion of the loop folds back on itself, the points in X_1 do not typically define a function. We address this as follows. First remove the points corresponding to the first bond which are less than 5% of the total loop height. The idea here is again to improve fit to the majority of the curve while giving up fit to the first bond. (Figure 5 demonstrates how the fit degrades when all points are included.) Apply a transformation to the points in X_1 by switching the x and y coordinates. We fit a cubic polynomial to the data using the “polyfit” function in Matlab. We now have a second cubic polynomial denoted $q_1x^3 + q_2x^2 + q_3x + q_4$. This cubic polynomial provides a reasonable fit to the data as shown in Figure 6. There was one class of loop profiles with a

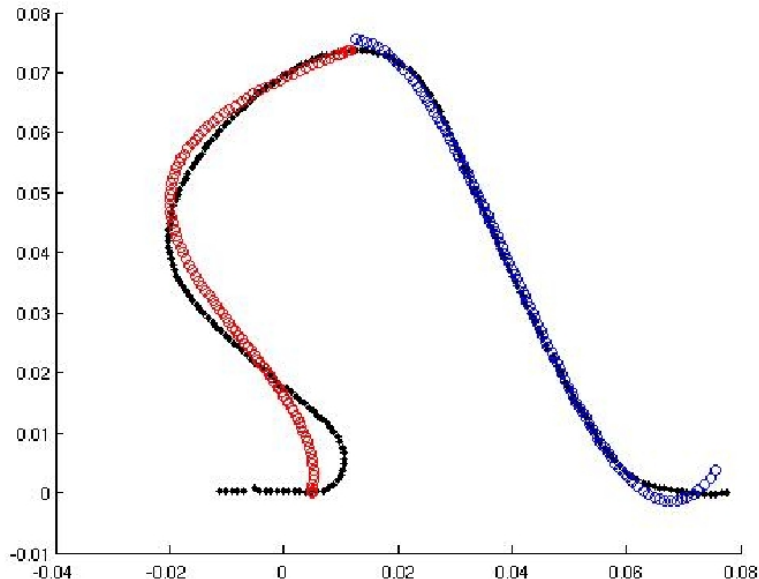


Figure 5: A poor fit when including the bond points.

wide “hump” at the top which did not admit a good fit, see Figure 7. We exclude this class of loops from the model. We now have eight output parameters $Y = \{p_1, \dots, p_4, q_1, \dots, q_4\}$ for the majority of loops.

Two Quintics with Boundary Conditions

As Figure 5 demonstrates, the two cubics fail to mimic the behavior of the loop at its apex and two weld points where the derivatives should all be 0. Here we parse the loop into two pieces as before and fit each piece with a degree 5 polynomial with constraints on the boundary in order to better fit the wire loop. This increase in fit unfortunately increases the number of output parameters.

Let (x_i, y_i) denote the apex of the loop, let X_2 denote the right hand portion of the wire loop as before. Let (x_j, y_j) denote the lowest point in the right portion of the wire loop, X_2 . That is, j is the argument which minimizes $\{y_j\}$ where $j \in \{i \dots n\}$. We seek a degree five polynomial $f(x)$ which fits the right hand portion of the loop and satisfies the additional constraints

$$\begin{aligned}
 f(x_i) &= y_i \\
 f'(x_i) &= 0 \\
 f(x_j) &= y_j \\
 f'(x_j) &= 0.
 \end{aligned}$$

The authors are aware of no least squares fitting algorithm which incorporates boundary conditions, so we formulate an optimization problem as follows. Let $f(x)$ denote the degree five polynomial which best fits the right hand data in the least squares sense, $f(x) = p_1x^5 + p_2x^4 + p_3x^3 + p_4x^2 + p_5x + p_6$. We seek a degree 5 polynomial, $g(x)$ so as to satisfy the conditions and still be “close” to $f(x)$. Let

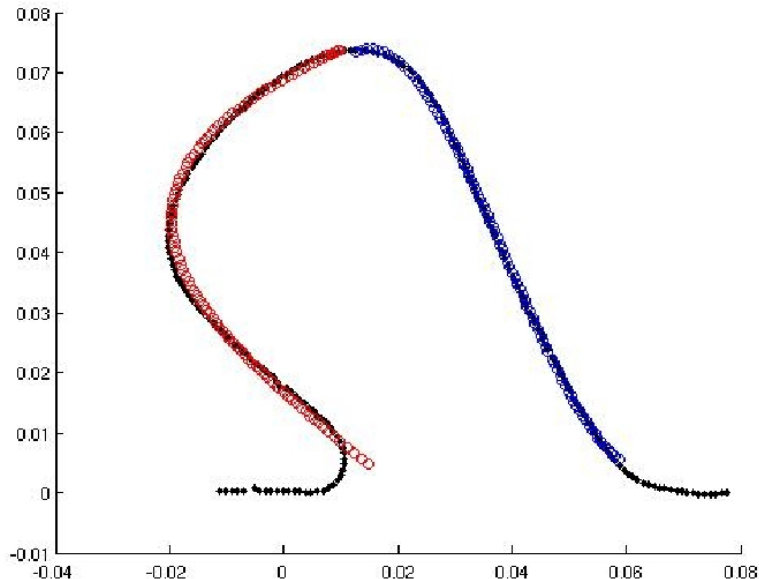


Figure 6: Two cubic polynomials fitted to a ribbon configuration excluding the welds.

$d(\cdot)$ denote the distance from $f(x)$ to $g(x)$,

$$\min d(g(x), f(x)) \text{ subject to}$$

$$\begin{aligned} g(x_i) &= y_i \\ g'(x_i) &= 0 \\ g(x_j) &= y_j \\ g'(x_j) &= 0. \end{aligned}$$

The issue here is how to define a sense of “closeness” between the boundary fitting polynomial $g(x)$ and the data fitting polynomial $f(x)$. The most direct and ideal metric would be if the polynomial $g(x)$ fit the data with as little error as possible, this prompts the metric:

$$d = \text{err}(g(x))$$

but the authors are unable to implement this in Matlab. We are able to implement the metric:

$$d(f(x), g(x)) = \sum_{i=1}^6 (p_i - q_i)^2$$

which is the sum of squares in the difference of coefficients. While less than ideal, this metric produced good results, both in value of the object function and observed fit, on all but the troublesome class of loops. See Figures 8, 9, 10 for examples of the fit to the right hand side data. Figure 11 demonstrates a poor fit on the wide-hump class of data, just as for the cubic polynomials.

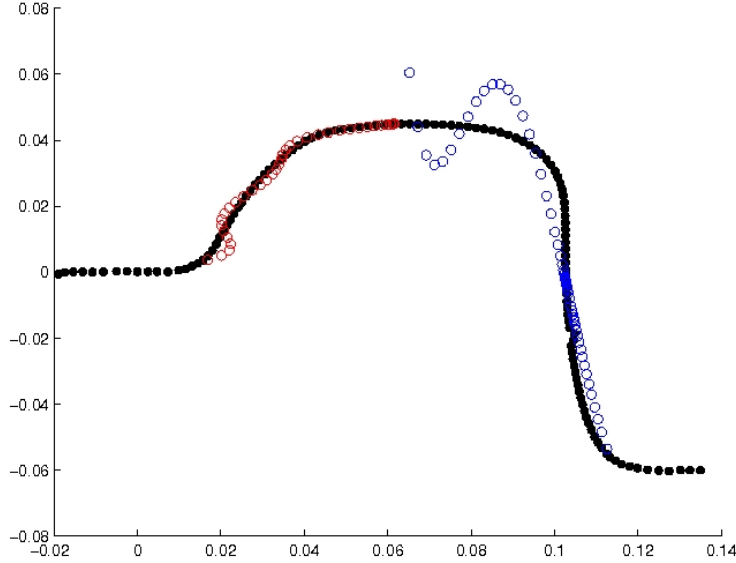


Figure 7: This “hump” class of loop profiles did not admit good fits with cubic polynomials.

The authors were not able to complete the parameters for the left hand side of the loop, which we leave as a future exercise. Together the coefficients of these two degree five polynomials will provide twelve parameters which describe the wire loop.

3.2 Mapping Functions

After finding the output parameters as needed, the next step is to find the mapping between the input and the output parameters. Thus, these mappings for the geometric and the shape models are discussed in the subsequent sections. As described before, output parameters for the complete model could not be finished so no work on mapping for it was performed.

3.2.1 Mapping for the Geometric Model

We had three sets of experiments in which the input parameters were varied as follows:

$$\text{Exp 1: } \begin{pmatrix} Step \\ Span \end{pmatrix}, \text{ Exp 2: } \begin{pmatrix} Span \\ ALH \\ ZLD \\ RF \\ RH \end{pmatrix}, \text{ Exp 3: } \begin{pmatrix} Step \\ Span \\ ALH \\ RF \\ RH \end{pmatrix}$$

We plugged in the data from these models into MINTAB and used response surface of DOE (Design of experiment) leading to the three mapping functions (for all three $L1$, $L2$, and L). The main problem now was to combine the three models. For this two strategies were used leading to two different mapping functions for the geometric model. The first strategy involved using nonlinear regression from Matlab, and the second involved using combo approach where all the second order or quadratic terms were thrown in the model and Minitab was used to find the coefficients for these models. The next two sub-sections discusses these two approaches. Another complementary approach to finding this mapping functions is using Neural networks. This approach did not yield promising results and

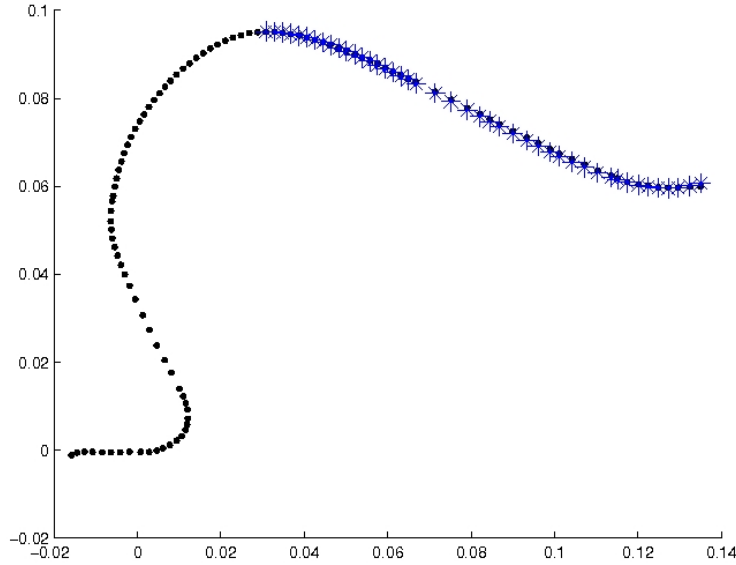


Figure 8: A good fit to the right of the loop using a quintic with boundary conditions.

is discussed in the last subsection.

Nonlinear Regression for the Geometric Model

Matlab provides two types of nonlinear regression; parametric models and regression trees. Regression trees were not helpful here because they give a range of values rather than a model (as needed). However, the parametric model approach seem appropriate here. That is pick all the terms from the three models (for all $L1$, $L2$, L individually), and combine it in one model. Then use this parametric model from Matlab to find the parameters (coefficients). To implement this “nlinfit” from Matlab was used. The function computes the least-squares parameter estimates for nonlinear models, and uses the Gauss-Newton at its core. An example of this fitting (for $L1$) is described below. Consider the three models:

- $L1 = 0.10 + 0.28Step - 0.21Span - 1.80Step^2 + 0.84Step * Span$
- $L1 = 0.10 - 1.04Span + 0.01ALH + 3.71ALH * Span$
- $L1 = 0.06 + 0.44Step - 0.30Span - 0.13ALH - 3.60Span^2 + 4.76ALH * Span$

Thus, the combined model is:

$$L1 = A + B * Step + C * Span + D * ALH + D * Step^2 + E * Span^2 + F * ALH * Span$$

After performing nonlinear regression, values for A through F are obtained. The small values of residuals indicated that models is a good fit to the given data. The final equations for $L1$ and $L2$ are given as:

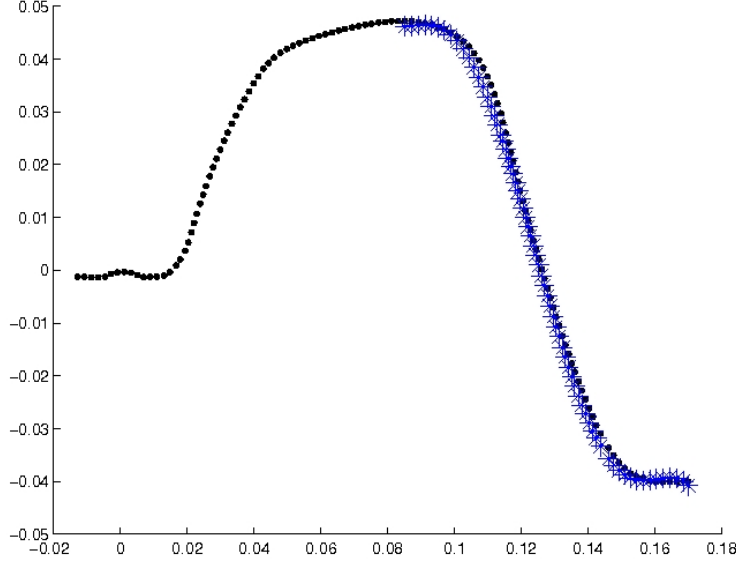


Figure 9: A good fit to the right of the loop using a quintic with boundary conditions.

$$L1 = -0.0798 + 0.3000 * Step - 2.6840 * Step^2 - 0.4716 * Span - 3.2063 * Span^2 + 0.7382 * Step * Span - 0.1680 * ALH + 5.0514 * Span * ALH \quad (3)$$

$$L2 = -0.0215 - 1.1027 * Step - 1.2272 * Step^2 + 0.7996 * Span - 2.8512 * Span^2 - 0.6087 * ALH + 1.0736 * RF + 2.8069 * RH + 5.5349 * Step * Span + 3.0353 * Span * ALH - 58.9305 * RF * RH + 0.0014 * Span * ZLD - 0.0003 * (Span * ZLD)^2 \quad (4)$$

The L obtained with this technique of first using Minitab to get a model and, then merging model together leads to a linear relationship between L and ALH . However, with some intuition in the process of how the loop is made, our mentor Michael Gray came up with the following model that seems to fit the given data nicely:

$$L = ((RF - ZLD * Span/100)^2 + (ALH)^2)^{0.5} - 0.002075 \quad (5)$$

Combo Model

To get a model that is more general and robust on a bigger region, one approach is to widen the database. We can do that by importing all the data we got into the response surface analysis. After the routine of choosing appropriate terms according to p-values of each term (this comes from Minitab, and for the parameter to have good effect its p-value should be less than 0.05), one can eventually obtain a model that covers wider area. As discussed before, for L a good model was found by our mentor and thus this exercise was performed only for $L1$ and $L2$.

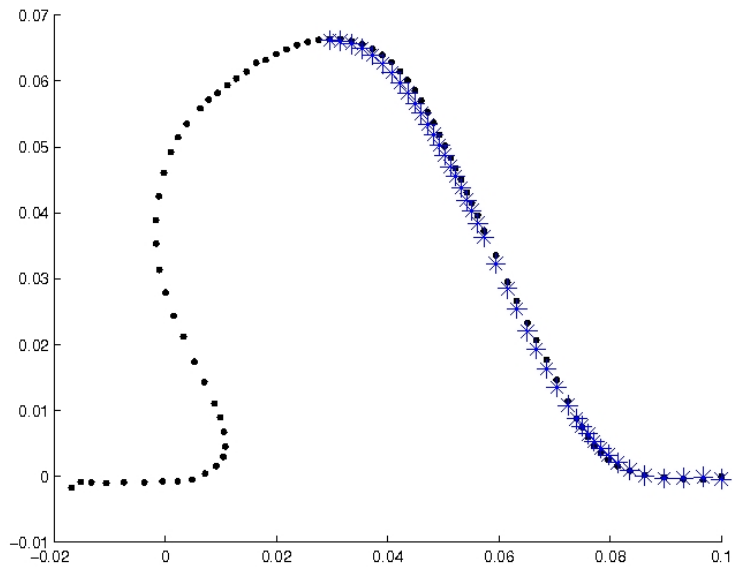


Figure 10: A good fit to the right of the loop using a quintic with boundary conditions.

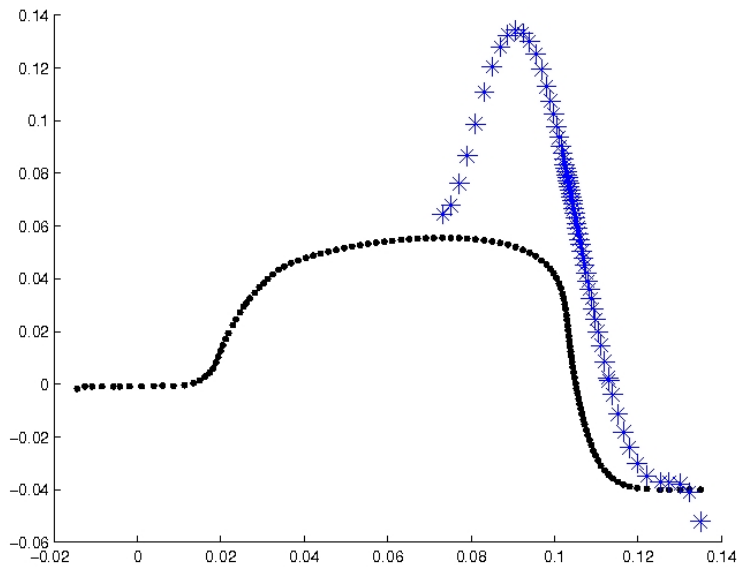


Figure 11: This “hump” class of loop profiles did admit good fits with quintic (or even higher degree) polynomials

	Term	Coef		Term	Coef
	<i>Constant</i>	-0.0920996		<i>Constant</i>	0.0352769
	<i>RH</i>	0.57184		<i>RH</i>	1.08484
	<i>RF</i>	2.41207		<i>RF</i>	-0.957534
	<i>ALH</i>	0.960984		<i>ALH</i>	-0.571882
	<i>ZLD</i>	-3.36E - 04		<i>ZLD</i>	-0.00134656
	<i>Step</i>	0.717422		<i>Step</i>	-2.70683
	<i>Span</i>	-0.522843		<i>Span</i>	0.88426
	<i>Angle</i>	-1.10E - 06		<i>Angle</i>	0.000203977
For L1:	<i>ALH * ALH</i>	-1.40193	For L2:	<i>Step * Step</i>	-1.7777
	<i>Step * Step</i>	-2.11697		<i>Span * Span</i>	-2.76033
	<i>Span * Span</i>	-3.2686		<i>Angle * Angle</i>	-2.44E - 07
	<i>Angle * Angle</i>	3.64E - 08		<i>RH * Angle</i>	-0.00523509
	<i>RH * ZLD</i>	0.0129024		<i>RF * ZLD</i>	0.0403931
	<i>RH * Step</i>	8.71624		<i>ALH * Step</i>	6.96839
	<i>RH * Span</i>	-4.78642		<i>ALH * Span</i>	2.49192
	<i>RH * Angle</i>	0.00148216		<i>Step * Span</i>	5.286
	<i>RF * ALH</i>	-15.2685			
	<i>RF * Step</i>	-19.6157			
	<i>RF * Span</i>	5.81131			
	<i>ALH * Span</i>	5.30272			
	<i>ALH * Angle</i>	-1.86E - 04			
	<i>Step * Span</i>	0.825457			
	<i>Step * Angle</i>	-7.37E - 05			
	<i>Span * Angle</i>	-1.02E - 04			

Neural Network

Neural Network Toolbox in Matlab can also be used to get the output parameters. Thus, this approach was explored. As discussed before, data from three experiments was available. There are 40 samples in Experiment 1, 20 samples in Experiment 2, and 32 samples in Experiment 3. By observing the data, we can find the values of ZLD are 0 except for 10 samples in Experiment 2. So we split the total data to training set and test set in three ways. First, we did not consider the 10 samples with ZLD, and put 56 of the remaining samples as training set, the other 26 as test set. Data in both training and test sets are uniformly picked from three experiments. Then we try to put those 10 samples which were deleted in the first way also in training and test set by 7 to 3. At last, we use Experiment 1 and 2 as training set to predict Experiment 3. We used the following Root-Mean-Square (RMS) of difference to verify the predictions:

$$\sqrt{\frac{1}{N} \sum (\hat{x} - x)^2}$$

where \hat{x} is the output given by a trained neural network, and x is the value obtained from experiment, which could be consider as true value. The results for the three different ways are listed in the Tables 1 through 3.

The mean of true values are also listed in the tables. It is clear that Cascade-forward backprop, Elman backprop, Feed-forward backprop and Linear layer design behave well in the first and second

Net Work Type	RMS of Length L	RMS of L1	RMS of L2
Cascade-forward backprop	$2.3027677025E - 03$	$2.7663337736E - 03$	$7.4841428216E - 03$
Elman backprop	$1.8446855437E - 03$	$3.0566501335E - 03$	$6.4746854574E - 03$
Feed-forward backprop	$1.8444550227E - 03$	$3.0566381240E - 03$	$6.4928695347E - 03$
Generalized regression	$1.4238081411E - 02$	$1.7057240254E - 02$	$1.8926290319E - 02$
Linear layer (design)	$1.8562988230E - 03$	$4.1034501047E - 03$	$7.2971651032E - 03$
Linear layer (train)	$1.3204285348E - 02$	$1.6041453028E - 02$	$1.8513416348E - 02$
Radial basis (exact fit)	$1.3077418181E - 02$	$2.2521124323E - 03$	$1.1513017462E - 02$
Mean Value	$2.3096126923E - 01$	$6.7665692308E - 02$	$5.5166576923E - 02$

Table 1: Results of first way

Net Work Type	RMS of Length L	RMS of L1	RMS of L2
Cascade-forward backprop	$1.9701090602E - 03$	$3.1808222813E - 03$	$8.2770561996E - 03$
Elman backprop	$2.0394472724E - 03$	$3.5238716897E - 03$	$7.0051527420E - 03$
Feed-forward backprop	$2.0390642925E - 03$	$3.5375324560E - 03$	$7.0051528253E - 03$
Generalized regression	$1.4681016023E - 02$	$1.6869156640E - 02$	$1.8908283531E - 02$
Linear layer (design)	$2.0441525840E - 03$	$4.3198027153E - 03$	$7.6253366737E - 03$
Linear layer (train)	$1.3587067814E - 02$	$1.5919766298E - 02$	$1.7945773035E - 02$
Radial basis (exact fit)	$1.3199013543E - 02$	$3.1000279228E - 03$	$1.4850660392E - 02$
Mean Value	$2.3175100000E - 01$	$6.8759241379E - 02$	$5.4779827586E - 02$

Table 2: Results of second way

Net Work Type	RMS of Length L	RMS of L1	RMS of L2
Cascade-forward backprop	$3.2760875198E - 01$	$3.0314959292E - 01$	$3.4292048299E - 01$
Elman backprop	$1.9368348821E - 02$	$2.6795692287E - 02$	$1.5185476131E - 02$
Feed-forward backprop	$9.0328087678E - 03$	$2.5552002369E - 02$	$1.4620851747E - 02$
Generalized regression	$2.3372236293E - 02$	$1.2117450721E - 02$	$2.1135208409E - 02$
Linear layer (design)	$2.8845804354E - 03$	$4.6975832903E - 03$	$4.8127271683E - 03$
Linear layer (train)	$2.1826631106E - 02$	$1.1888047296E - 02$	$2.0482658737E - 02$
Radial basis (exact fit)	$9.2172690013E - 01$	$6.5427811325E - 01$	$1.4927900117E + 00$
Mean Value	$2.1734640625E - 01$	$6.5796312500E - 02$	$5.1828218750E - 02$

Table 3: Results of third way

way, and only Linear layer design is properly behaved in the third one. It is observed that Neural networks predict L better than $L2$ and $L3$, since the last two have less scaling. Also, the first two ways are better because the training sets are chosen properly to cover more information in each experiment, while the last way does not include information from Experiment 3 directly.

Another fact we should consider here is that each input parameter in our data set does not has a large range and does not change a lot. For example, there are only five different values for RH , 0.0394, 0.0315, 0.03545, 0.041912, 0.028988, and most samples take the first three values. The same thing happens to the other input parameters. This could be a factor that brings good results in neural network, instead of the neural network itself. Finally, the analysis above do show there might be possible to predict $L1$, $L2$ and L accurately by Neural Network, when the training set is chosen carefully and the network function works well.

3.2.2 Mapping for the Shape Model

Here we try to find a mapping $f(X)$ to map a input parameter of the machine X to $\tilde{\kappa}_{max}(=max_s \tilde{\kappa}(s))$. But we cannot get a good model for all six input parameters. So we consider the simplified case where $RH/RF = 1$ and $ZLD = 0$. We rewrite X to be $(R, ALH, step, span)$ where $R = \sqrt{RH^2 + RF^2}$. We do response surface fitting for some linear combination of X , and the model we get is the following:

$$\begin{aligned} \hat{\kappa}_{max} = & 45.4718 + 144.206 * fac1 - 250.9309 * fac2 \\ & - 891.0382 * fac1 * fac2 \\ & - 1940.5402 * fac1^2 - 456.6026 * fac2^2 \end{aligned} \quad (6)$$

where

$$fac1 = 0.0095 - 0.0029 * R - 0.0107 * ALH + 0.9992 * step - 0.0384 * span \quad (7)$$

and

$$fac2 = -0.1913 + 0.0433 * R + 0.2125 * ALH + 0.0399 * step + 0.9754 * span \quad (8)$$

We do not know what exactly $fac1$ and $fac2$ mean. They may be an approximation to some fundamental underlying factors or just resulted from the setting of this specific experiment. But the performance is not bad. We have 69 available loops. We use the first 60 loops for model fitting and last 9 loops for testing. The fitting error is 10.3%. For 95% confident interval, 7 of 9 are predicted.

4 Optimum Loop Shape

From the ‘‘Geometric model’’ we obtained the expression for length (L), and from the ‘‘Shape model’’ (curvature model) we obtained the expression for maximum curvature (κ) in terms of the input parameters i.e. $(Step, Span, ALH, ZLD, RF, RH)$. The main idea then was to find the optimal input parameters such that the curvature is minimized and the length is maximized (under dozen different types of design constraints). The objective function is made of the following parts:

$$y = w_1 \widehat{\kappa}_{max} + w_2(1/L), \quad (9)$$

where w_1 is the weight factor given to the curvature and w_2 is the weight factor given to the length (typical values could be 0.8 and 0.2 respectively). The optimization routine consists of two different parts. One in which the statistical box (defined by values of Step and Span that can be varied) is

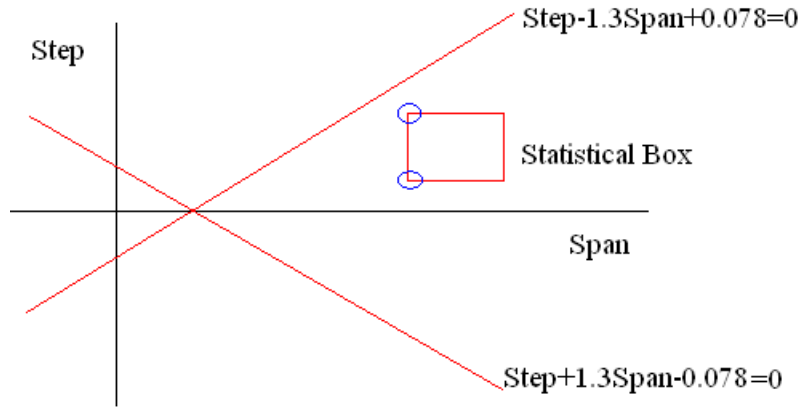


Figure 12: Machine and statistical bound for Step and Span.

fixed, and the other in which the this box is allowed to move in the global constraints enforced by the machine (as depicted in Figure 12).

The challenging aspect in this part was writing a optimization routine that satisfied all the constraints. “fmincon” function from Matlab was used to achieve the end goal. This function finds the minimum of a constrained nonlinear multivariable function. The constraints that were provided are:

- Global bounds on all input variables enforced by the machine i.e [Step, Span, ALH, ZLD, RF, RH];
- Lower and upper bounds on L1 for all combination of statistical lower and upper bound of Step and Span.
- Lower and upper bounds on L2 for all combination of statistical lower and upper bound of Step and Span.

For optimization, we used the Equation (6) for the maximum curvature and Equation (5) for the length. We tested the algorithm by passing the input values for the algorithm corresponding to the first experiment data and checking the optimized input parameters. The optimized input parameters closely matched the input parameters from first experiment. Another way to test this algorithm is to solve it to minimize for maximum curvature and compare the input parameters with those experiments for which we know the maximum curvature is small.

5 Incorporating Material Response Behavior in the Empirical Model

This section discusses the literature review of how different industries model the wire bonding process keeping the material response behavior in perspective. It also discusses how the empirical model developed for the problem at hand can be improved upon by considering mechanics of material.

In previous researches on wire bonding process, the parameters used to measure the loop height and the quality characteristic of wire bonding process include: ball height, ball size, pull strength, shear strength, the height of heat affected zone (HAZ). In fact, wire bonding loop height has a close

connection with wire bonding loop process technology, while the forming of wire bonding is mainly affected by the materials and looping parameters of wire bonding. The widely-used statistical research methods are: design of experiment (DOE) (Groover et al.,1994), response surface methodology (RSM) (Shu, 1992), finite element method (Chaudhry and Barez,1998) and Taguchi method (Chen, 2000) [2]. [3] presents a three-dimensional finite element model of the thermosonic wirebond looping process. An integration of artificial neural networks (ANN) with artificial immune systems (AIS) is proposed to optimize parameters for an IC wire bonding process in [1]. [4] also proposes a novel method using springs to simulate the condition of forces acting on a section of the bonding wire.

The previous researches on the forming of gold wire track, loop height and gold wire strength showed that gold wire (shape, materials, length, and radius), discharge strength (the length of HAZ), the design of lead frame (leads' width, thickness and length), the materials of lead frame, shape of wire loop (the width and length of wire loop, turns and the position of turn), the pull wire parameters of wire bonder, etc are the control factors affecting the quality of wire bonding. The correlative process parameters are: wire bonder, types of bonding probe, temperature of wire bonding, time of wire bonding, bonding force, bonding speed, ultrasonic power, bonding time, free air ball size, loop length, loop height, etc [2]. Medtronic ribbon has similarity to the previous wire bonding problem with Cu-Ni flat ribbon instead of gold round ribbon. While we already consider parameters such as loop length, height and curvatures, next we may find other ways to detect the properties of the ribbon and understand more mechanic principles to help building the model.

6 Acknowledgements

We would like to thank the Institute for Mathematics and its Applications and Medtronic Inc. for making this workshop possible. We would also like to thank our mentors J. Michael Gray and Robert Shimpa for presenting a very interesting and challenging problem and motivating us to do our best, and Dr.Grandine from Boeing for helping us out.

References

- [1] Tung-Hsu Hou, Chi-Hung Su and Hung-Zhi Chang, *Using neural networks and immune algorithms to find the optimal parameters for an IC wire bonding process*. Expert Systems with Applications 34, 427 – 436, 2008.
- [2] Yung-Hsiang Hung, *Optimal process parameters design for a wire bonding of ultra-thin CSP package based on hybrid methods of artificial intelligence*. Microelectronics International 24/3, 3 – 10,2007.
- [3] D.S. Liu, Y.C. Chao and C.H. Wang, *Study of wire bonding looping formation in the electronic packaging process using the three-dimensional finite element method*. Finite Elements in Analysis and Design 40, 263 – 286, 2004.
- [4] Yu-Lung Lo, Tien-Lou Ho, Jau-Liang Chen, Rong-Shean Lee, and Tei-Chen Chen, *Linkage-Spring Model in Analyzing Wirebonding Loops* . IEEE Transactions on Components and Packaging Technology, Vol.24, No.3 , September 2001.
- [5] J. Michael Gray and Robert Shimpa, *Ribbon Formation for Electrical Interconnection*. IMA Workshop 2008, August 6th, 2008.