Linear simplified multi-layer planar inductor calculations

Thijs van Liempd **2023**

Predicting coil inductances for the purposes of analytical design (as opposed to trial-and-error) has been a challenge to electrical engineers for several decades. In 1999, a paper was published which is now widely used and referenced, that details formulae for predicting the inductance values of single-layer planar inductors. This paper [1] allows for coil designs in 4 different shapes, but is limited to single layers, as predicting the multi-layer coupling was omitted entirely.

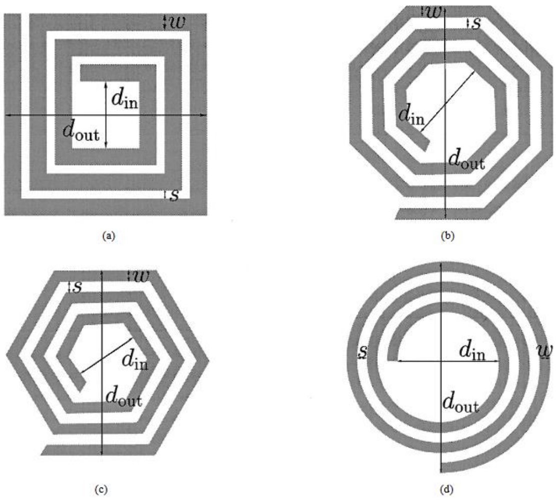


Figure 1: the 4 different coil shapes described in paper [1]. Source: [1]

Since then, several papers have tried, with varying degrees of success and difficulty. For example, a paper from 2021 [2] uses derived Maxwell equations to model every line-segment individually. Unfortunately, while their accuracy results are impressive, the complexity of their method (and the lack of provided tools for doing so) is not significantly better than using FEM simulation software. The industry standard approach appears to be largely focused on the use of Finite Element Method simulation software, which usually achieves the most accurate predictions. The downside of FEM software is in the complexity of use, and the high license fees associated with professional design software tools. To avoid both these issues, I attempted to create an open-source python-based tool for generating coils, using simpler formulae than [2]. A 2010 paper [3] (although it's more of a brief article) describes a simplified method, based on a few polynomial equations. It does not, however, provide a general formula for using the calculated coupling factors, which is where paper [4] comes in. To validate the combination of these 3 papers ([1], [3] and [4]), approximately 50 samples (spread over 6 PCBs) of different PCB coil designs were tested and compared to the predictions.

The aforementioned open-source python-based tool: https://github.com/thijses/PCBcoilGenerator
For more information about the production of these samples, please refer to 'QRD21343-0007.docx'.

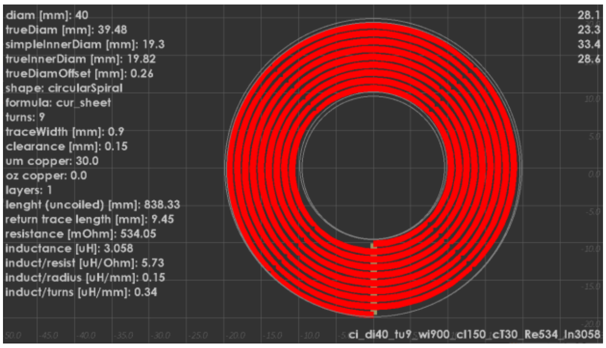


Figure 5: circular coil example (screenshot of custom python-based generation tool developed for this study)

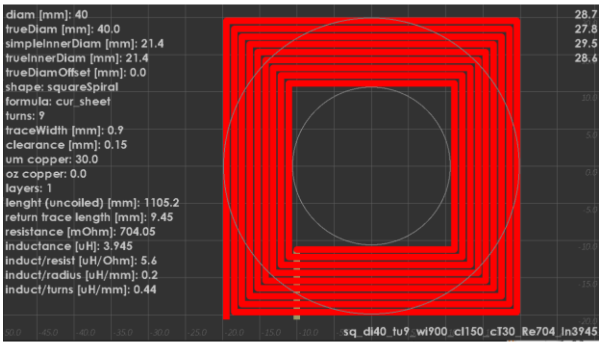


Figure 4: square coil example (screenshot of custom python-based generation tool developed for this study)

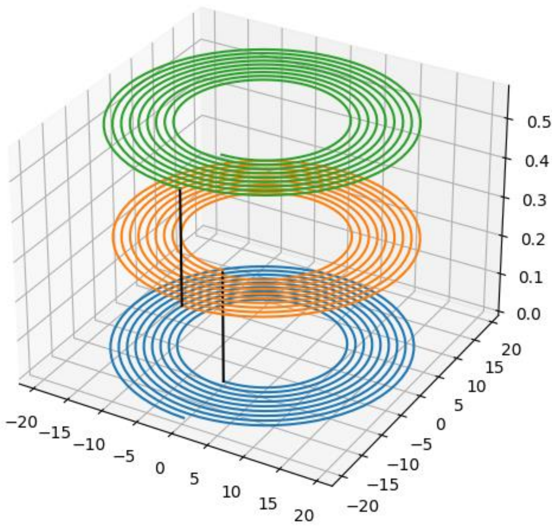


Figure 3: example rendering of a 3-layer circular coil

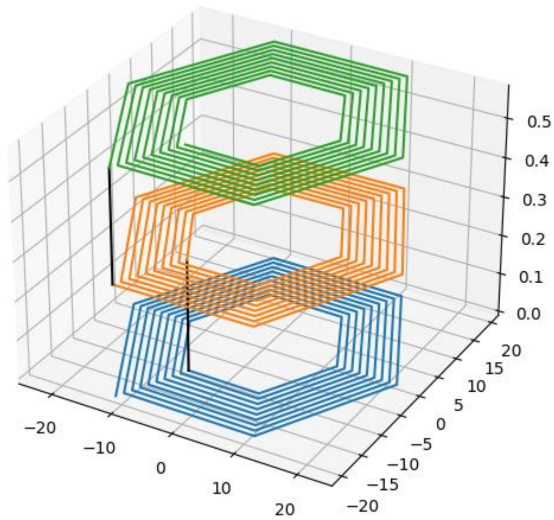


Figure 2: example rendering of a 3-layer hexagonal coil

From my test samples, initial observations reveal that the single-layer predictions from [1] are impressively accurate. The problems start when attempting to find a single formula for multilayer coils. The data I collected strongly indicates that the constants from [3] and/or formulas from [4] is incorrect in some way. The predicted inductance deviates from the measured inductance as the number of layers increases. Unfortunately, [3] does not explain how they arrived at their constants. Assuming the formulas from [4] are derived from theoretical physics, the coupling factors 'k' (calculated using the parameterized expressions from [3]) are likely the source of the problem.

The inductance of multi-layer coils is not simply the sum of the individual layers. The layers will create a mutual inductance, which (depending on the polarity of the coils), will either increase- or decrease the final inductance value. The inductance for two coils (polarized in the same direction) is:

$$L_{total} = L_1 + L_2 + 2 * M$$

Where $M = k * \sqrt{L_1 * L_2}$ L_n is the inductance of each coil (in Henry) k is a coupling factor

Assuming that the coils have the same inductance (and are polarized in the same direction), the formula can be simplified to:

$$L_{total} = L_n * (2 + 2k)$$

Note: the coupling factor k is a number between 0.0 and 1.0.

The original approach

It's important to note that [3] provides formulae for calculating the coupling factor, but omits a formula for calculating the inductance of an arbitrary multi-layer coil (using the coupling factors). To fill this gap in the mathematics, I used a formula from [4]. The combined mathematics works out to the following:

Result:	**Formula:**	**Named variables:**
Inductance (single-layer) Source: [1]	$L = \frac{C_0 \mu_0 {n_t}^2 d_{avg}}{2}\left[\ln\left(\frac{C_1}{\varphi}\right) + C_2\varphi + C_3\varphi^2\right]$ $d_{avg} = (d_o + d_i)/2$ $\varphi = (d_o - d_i)/(d_o + d_i)$	μ_0: magnetic constant [N/A] n_t: number of turns d_{avg}: average diameter [m] φ: fill factor d_o: outer diameter [m] d_i: inner diameter [m] C_i: constants from paper [1] L: inductance [H]
Inductance (multi-layer) Sources: [3] & [4]	$s_l\langle i\rangle = \frac{thick_{PCB} - thick_{copper}}{N_l - 1} * i$ $K_i = \frac{{n_t}^2}{\left(X_{k_0}{s_l}^3 + X_{k_1}{s_l}^2 + X_{k_2}s_l + X_{k_3}\right) * \left(Y_{k_0}{n_t}^2 + Y_{k_1}n_t + Y_{k_2}\right) * Z_k}$ $L = L_{single} * \left(N_l + 2\left(\sum_{i=1}^{N_l}(N_l - i)K_i\right)\right)$	X_k, Y_k, Z_k: constants from paper [3] s_l: space between layers [mm] (function of layer 'i' if N_l>2) n_t: number of turns K_i: coupling factor between layers (function of layer separation s_l) L_{single}: inductance of a single layer of the coil [H] N_l: number of layers L: inductance of the whole coil [H]

Or as expressed in Mathcad:

'current sheet' math (shown in 2 seperate papers!)

$N_{turns} := 9$

$\mu_0 := 4\,\pi \cdot 10^{-7}\ \frac{N}{A}$

$D := 39.5\ mm$

$d := 19.8\ mm$

$\varphi := \frac{D-d}{D+d} = 0.332$

$D_{avg} := \frac{D+d}{2} = 29.65\ mm$

$C_{square} := \begin{bmatrix} 1.27 \\ 2.07 \\ 0.18 \\ 0.13 \end{bmatrix}$

$C_{hexagon} := \begin{bmatrix} 1.09 \\ 2.23 \\ 0.00 \\ 0.17 \end{bmatrix}$

$C_{octagon} := \begin{bmatrix} 1.07 \\ 2.29 \\ 0.00 \\ 0.19 \end{bmatrix}$

$C_{circle} := \begin{bmatrix} 1.00 \\ 2.46 \\ 0.00 \\ 0.20 \end{bmatrix}$

$C := C_{circle}$ quick selection

$$L_{single} := \frac{\mu_0 \cdot {N_{turns}}^2 \cdot D_{avg} \cdot C_0}{2\ A} \cdot \left(\ln\left(\frac{C_1}{\varphi}\right) + C_2 \cdot \varphi + C_3 \cdot \varphi^2\right) = 3.055\ \mu H$$

/\ NOTE: 'A' is only there to ignore the Ampere
| resolve the lack of units in C

multilayer paper

$N_{turns} := 9$

$X_{total} := 0.6$

$X_{copper} := 0.030$

$N_{layers} := 2$

$X := \frac{X_{total} - X_{copper}}{(N_{layers} - 1)} = 0.57$

$A_k := \begin{bmatrix} 0.184 \\ -0.525 \\ 1.038 \\ 1.001 \end{bmatrix}$

paper [2] calls these A,B,C and D respecitvely. I think A_k is more legible

$B_k := \begin{bmatrix} 1.67 \\ -5.84 \\ 65 \end{bmatrix}$

$C_k := 0.64$

coupling value:

$$K_C(X_f) := \frac{N_{turns}^2}{\left(A_{k_0} X_f^3 + A_{k_1} X_f^2 + A_{k_2} X_f + A_{k_3}\right) \cdot \left(B_{k_0} N_{turns}^2 + B_{k_1} N_{turns} + B_{k_2}\right) \cdot C_k}$$

$$L_{multi} := L_{single} \cdot \left(N_{layers} + 2 \cdot \left(\sum_{i=1}^{N_{layers}} (N_{layers} - i)\ K_C(i \cdot X)\right)\right) = 9.704\ \mu H$$

$$K_{CA}(X_f) := A_{k_0} X_f^3 + A_{k_1} X_f^2 + A_{k_2} X_f + A_{k_3}$$

$$K_{CB}(N_{turns\_f}) := \frac{N_{turns\_f}^2}{B_{k_0} N_{turns\_f}^2 + B_{k_1} N_{turns\_f} + B_{k_2}}$$

constant for any number of layers, varies with turns

$$L_{multi} := L_{single} \cdot \left(N_{layers} + 2 \cdot \left(\sum_{i=1}^{N_{layers}} \frac{(N_{layers} - i)}{K_{CA}(i \cdot X)}\right) \cdot K_{CB}(N_{turns}) \cdot \left(\frac{1}{C_k}\right)\right) = 9.704\ \mu H$$

Identifying the source of the error

By designing test sample PCBs to only vary 1 parameter at-a-time, I attempted to calculate better-fitting constants. As mentioned earlier, [3] only mentions "*Experimenting over the range of inductor turns, N, with N equal to a 5- to 20-turns ratio, and the distance between the inductors on the two layers, X, with X equal to a 0.75- to 2-mm distance,*" for how they arrived at their constants. The formulas themselves are sufficiently complex to require significant effort to reverse-engineer, if one wanted to exclusively change the constants themselves (instead of replacing whole formulas). Fortunately, my dataset revealed several simplified relations between coil design parameters and inductance values. For example, the effects of the number of turns (on multi-layer coupling specifically, not on single-layer inductance) appears to be linear instead of polynomial:

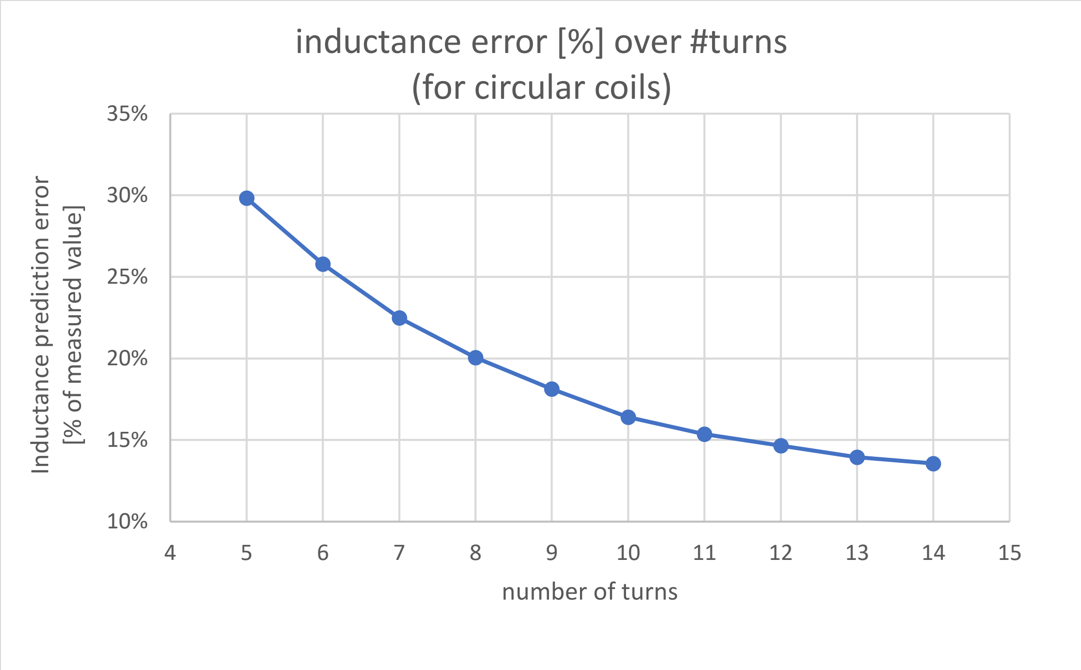


Figure 6: this graph is meant to indicate that there is an inaccuracy in the formula that uses number of turns as a parameter

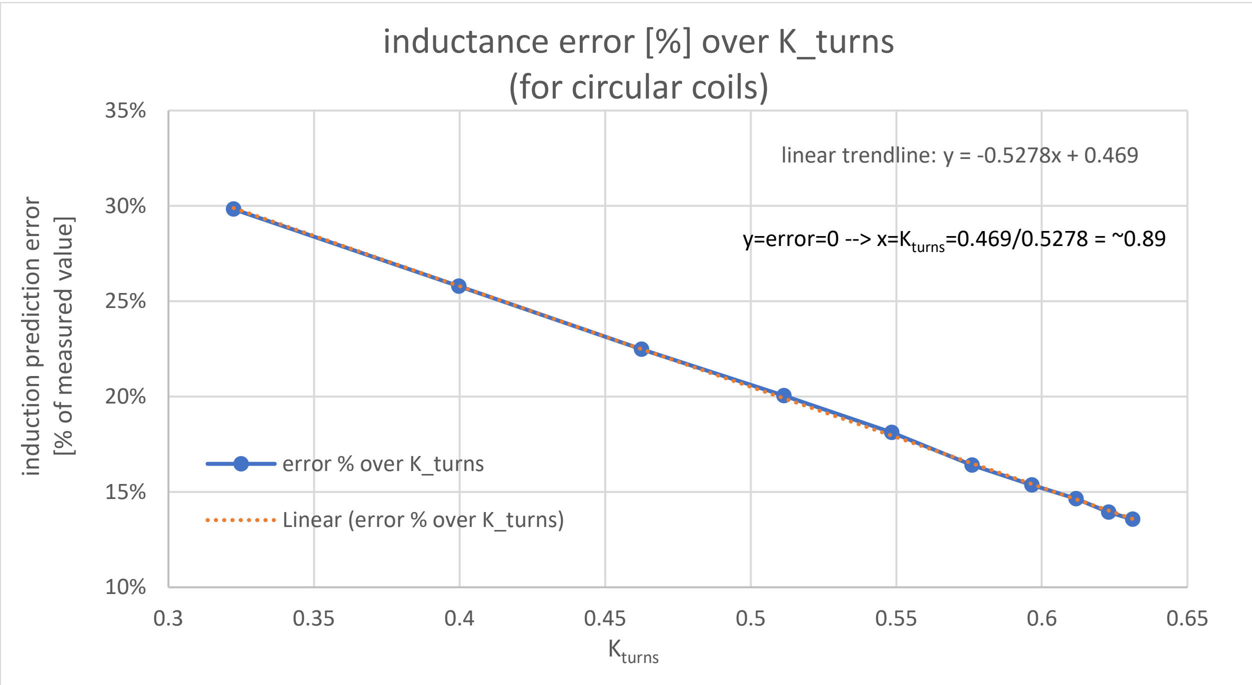


Figure 7: it appears there is likely a linear relation between the prediction error and the calculated factor 'K_{turns}'

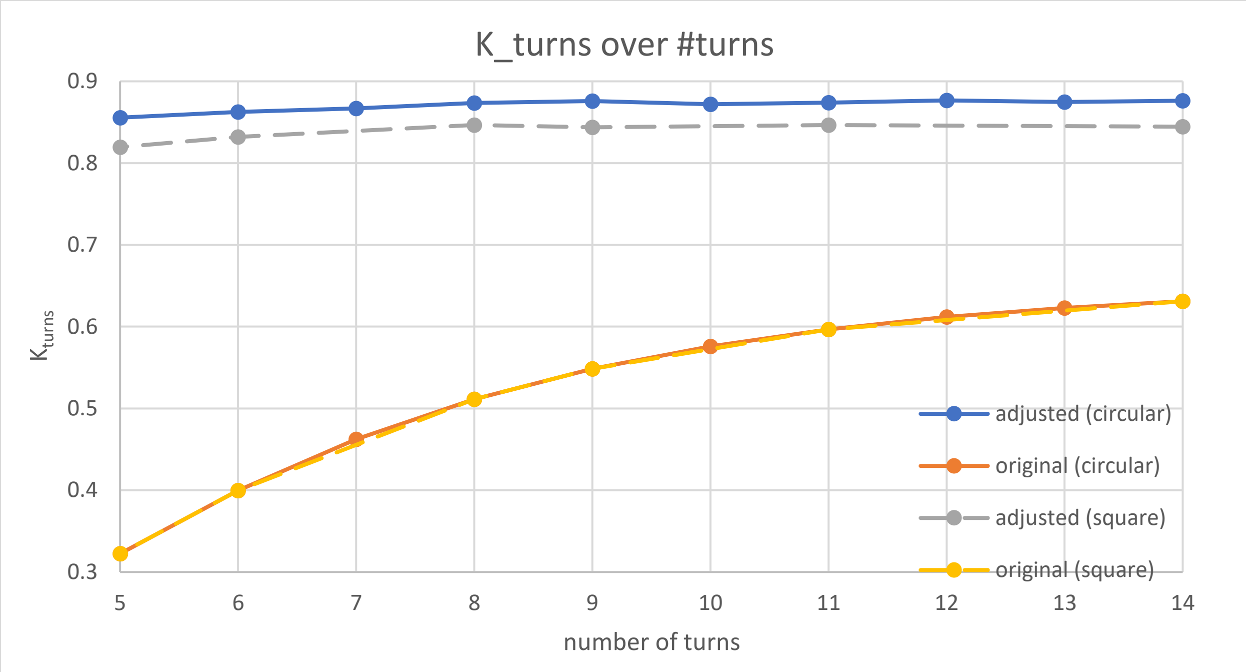


Figure 8: whilst the original K_{turns} is nonlinear curve, the adjusted values (based on my measured data) is roughly constant

As shown in Figure 8, when recalculating K_{turns} for each sample, the resulting curve is functionally constant. There is a noteworthy offset between circular and square coils, but both lines are similarly flat. A wider dataset (below 5 and above 14 turns) may introduce more pronounced curvature, but for the purposes of simplifying the mathematics, I will eliminate the number-of-turns as a parameter for multi-layer coupling (so not for single-layer) calculations, and instead adjust remaining constants to incorporate this constant factor.

The original formula places significant emphasis on the effect of layer spacing on calculation of k. However, it uses an inverted 4th order polynomial:

$$K_i = \frac{{n_t}^2}{\left(X_{k_0}{s_l}^3 + X_{k_1}{s_l}^2 + X_{k_2}s_l + X_{k_3}\right) * \left(Y_{k_0}{n_t}^2 + Y_{k_1}n_t + Y_{k_2}\right) * Z_k}$$

Which can be rewritten as: $K_i = K_{spacing} * K_{turns} * \frac{1}{Z_k}$ where $K_{turns} = \frac{{n_t}^2}{Y_{k_0}{n_t}^2 + Y_{k_1}n_t + Y_{k_2}}$

Which leaves

$$K_{spacing} = \frac{1}{X_{k_0}{s_l}^3 + X_{k_1}{s_l}^2 + X_{k_2}s_l + X_{k_3}} = \frac{1}{0.184{s_l}^3 - 0.525{s_l}^2 + 1.038s_l + 1.001}$$

As shown in Figure 8, the effect of the number of turns (K_{turns}) appears to be exaggerated, which leaves the layer spacing as the only relevant parameter in the original formula.

Finding new constants

To determine the effect of layer-spacing between 2 layers, the following data is grouped:

PCB # of layers	Sample name(s)	**Shape:**	**Layers:**	**Turns:**	**Trace width [mm]:**	**Clearance [mm]:**	**Diam. [mm]:**	**layer spacing [mm]:**	**Measured Inductance [µH]:**
6L	top in1	Circular	2	8	1	0.1	24	0.1245	3.224
	top in2	Circular	2	8	1	0.1	24	0.4897	3.101
	top in3	Circular	2	8	1	0.1	24	0.6137	3.058

	top in4	Circular	2	8	1	0.1	24	0.9789	2.947
	top bot	Circular	2	8	1	0.1	24	1.1034	2.904
	in2 in3	Circular	2	8	1	0.1	24	0.124	3.236
	in1 in2	Circular	2	8	1	0.1	24	0.3652	3.137
	in1 in4	Circular	2	8	1	0.1	24	0.8544	2.977
2L	top bot	Circular	2	9	0.9	0.3	40	0.57	10.415
4L	top in1 & in1 in2	Circular	2	9	0.9	0.3	40	0.23	10.597
	top in2	Circular	2	9	0.9	0.3	40	0.4655	10.415
	top bot	Circular	2	9	0.9	0.3	40	0.701	10.23

Table 1: 2-layer test samples with varying layer-spacing

Which results in the following graph:

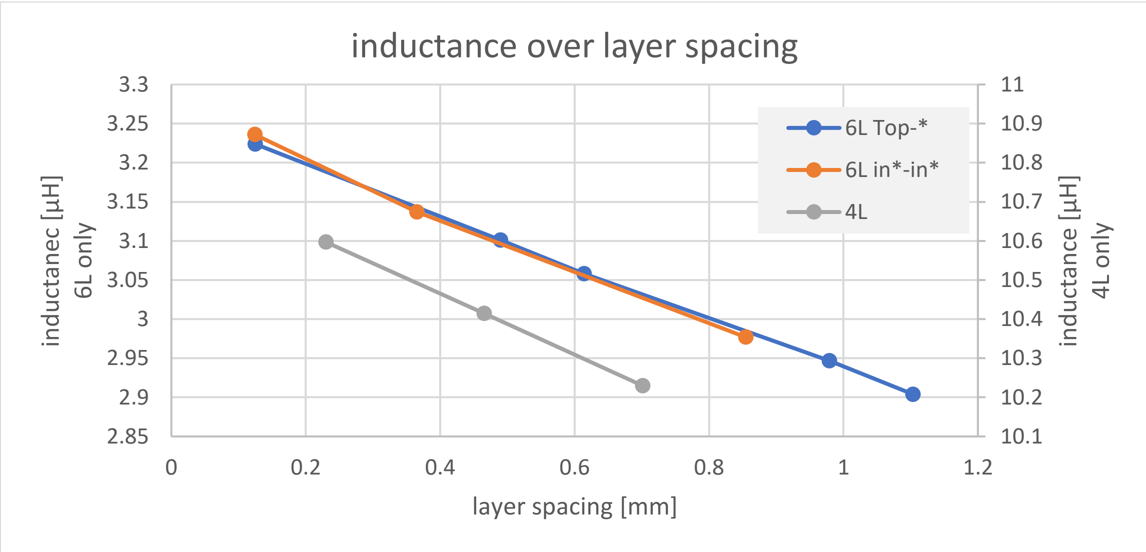


Figure 9: relation between layer spacing and inductance for 2-layer samples

Figure 9 indicates that the relationship between inductance and layer spacing is approximately linear, within the confines of the current dataset. It is possible that the relation becomes non-linear outside this range (below 0.2mm or above 1.0mm). However, standard PCB designs should fall within this range, as is demonstrated by JLCPCB's 'instant quote' order options, shown below.

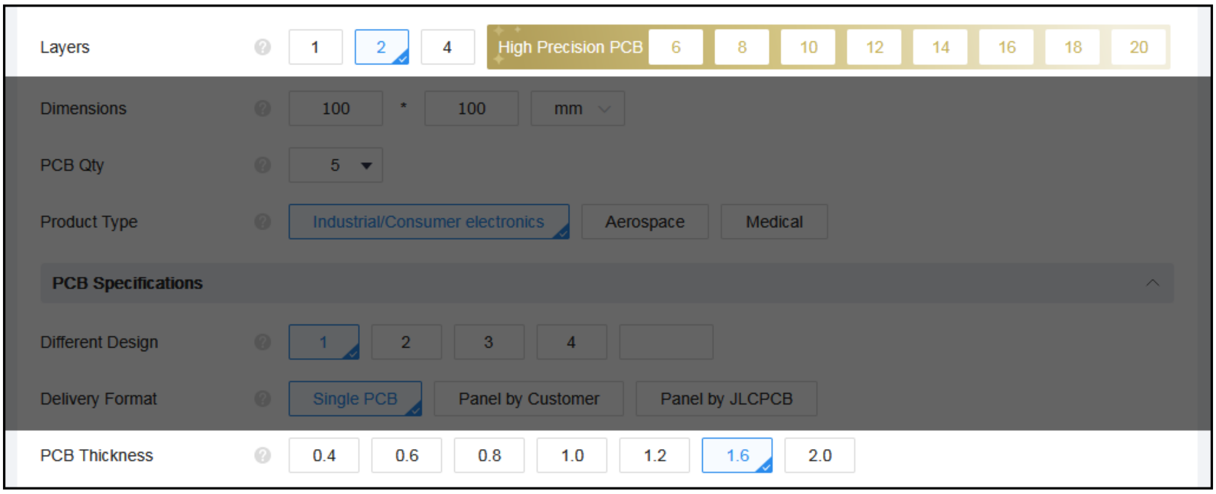


Figure 10: partial screenshot of JLCPCB's 'instant quote' webpage. Demonstrates common PCB thickness options.

By normalizing the data using the single-layer inductance predictions, coupling factor 'k' as a function of layer spacing can be established:

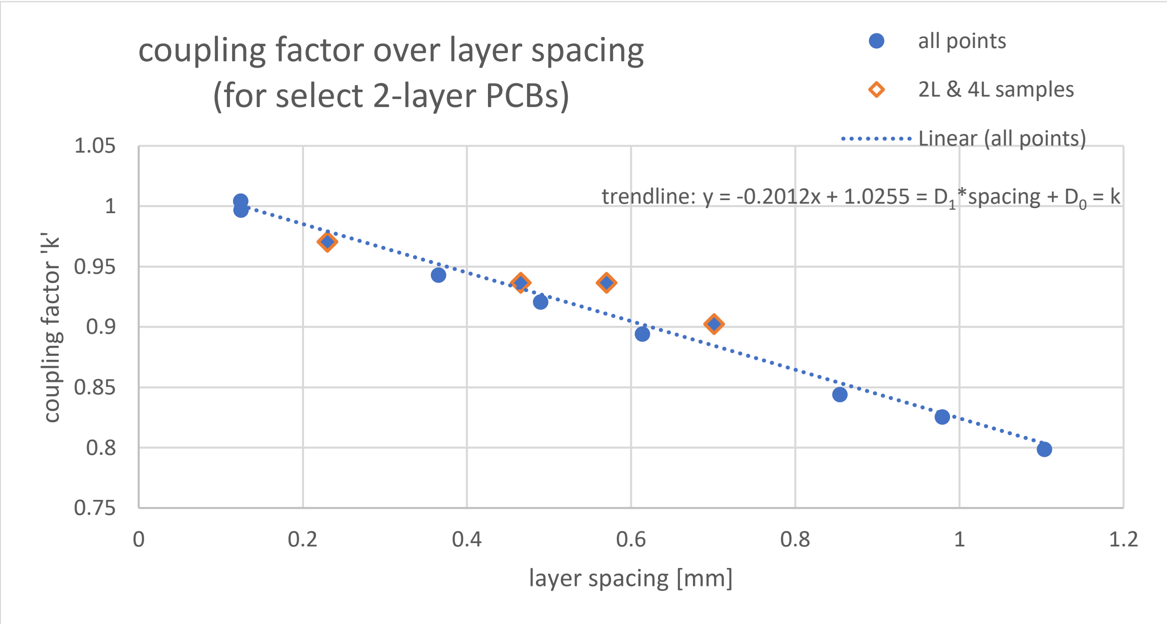


Figure 11: linear trendline of layer spacing over coupling factor 'k', for select 2-layer PCBs. Note: the majority of the samples comes from the 6-layer PCB, so the samples from the 2-layer and 4-layer PCBs are tagged (see legend).

The newly proposed formula for multilayer inductance becomes:

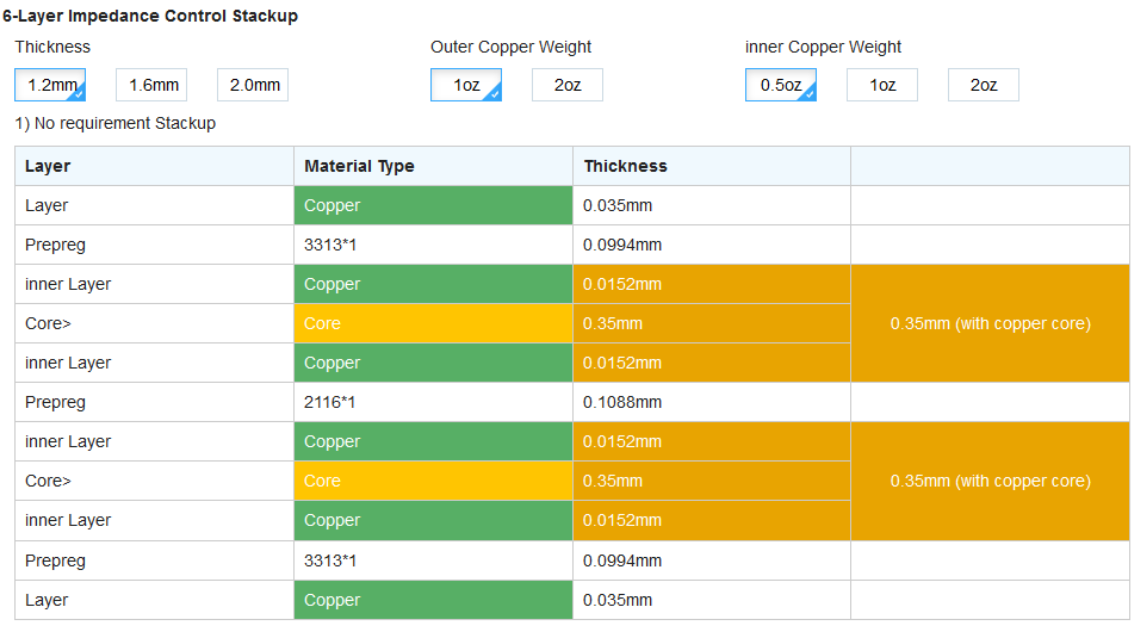
Variable:	**Formula:**	**Named variables:**
Inductance (multi-layer)	$K_i = D_1 s_l + D_0$ $L = L_{single} * \left(N_l + 2 * \sum_{i=1}^{N_l} (N_l - i) K_i \right)$	D_1, D_0: new constants s_l: space between layers [mm] (function of layer 'i', at this stage) K_i: coupling factor between layers (function of layer separation s_l) L_{single}: inductance of a single layer of the coil [H] N_l: number of layers L: inductance of the whole coil [H]

Using the best-fit linear trendline (least-squares) to find the 1st order polynomial (linear function), the new predictions come out to:

PCB # of layers	Sample name(s)	Layers	layer spacing [mm]	Measured Inductance [µH]	New predicted Inductance [µH]	New predict. error [% of measured]	Old predict. error [% of measured]
6L	top in1	2	0.1245	3.224	3.230	0.2%	14.3%
	top in2	2	0.4897	3.101	3.111	0.3%	18.3%
	top in3	2	0.6137	3.058	3.071	0.4%	18.8%
	top in4	2	0.9789	2.947	2.952	0.2%	19.3%
	top bot	2	1.1034	2.904	2.912	0.3%	19.1%
	in2 in3	2	0.124	3.236	3.230	0.2%	14.6%
	in1 in2	2	0.3652	3.137	3.151	0.5%	17.4%
	in1 in4	2	0.8544	2.977	2.993	0.5%	19.0%
2L	top bot	2	0.57	10.415	10.276	1.3%	18.0%
4L	top in1 & in1 in2	2	0.23	10.597	10.643	0.4%	13.5%
	top in2	2	0.4655	10.415	10.389	0.3%	16.5%
	top bot	2	0.701	10.23	10.134	0.9%	18.1%

Table 2: results of new 2-layer predictions (using initial D_n constants)

To apply the newly derived formula to samples with more than 2 layers, the layer spacing calculations become slightly more complex. The 4-layer PCB has nearly-equidistant layer spacing, but the 6-layer varies significantly, as shown in the 'stack-up' screenshot below.


6-Layer Impedance Control Stackup

Thickness: 1.2mm | 1.6mm | 2.0mm

Outer Copper Weight: 1oz | 2oz

inner Copper Weight: 0.5oz | 1oz | 2oz

1) No requirement Stackup

Layer	Material Type	Thickness	
Layer	Copper	0.035mm	
Prepreg	3313*1	0.0994mm	
inner Layer	Copper	0.0152mm	0.35mm (with copper core)
Core>	Core	0.35mm	
inner Layer	Copper	0.0152mm	
Prepreg	2116*1	0.1088mm	
inner Layer	Copper	0.0152mm	0.35mm (with copper core)
Core>	Core	0.35mm	
inner Layer	Copper	0.0152mm	
Prepreg	3313*1	0.0994mm	
Layer	Copper	0.035mm	

Figure 12: layer stack-up used for '6L' sample PCB. Source: https://jlcpcb.com/impedance

To solve this problem, we can rewrite the whole inductance formula to allow for arbitrary spacing. The original formula $\sum_{i=1}^{N_l}(N_l - i)K_i$ can be visualized using the following infographic:

$$3K(1s) + 2K(2s) + 1K(3s) = \sum_{i=1}^{N_l}(N_l - i)K(i * s)$$

Where N_l is the number of layers, K is the formula for the coupling factor for a given layer spacing and s is the spacing between two adjacent layers.

Figure 13: infographic to visualize inter-layer coupling permutations

Therefore, the new formula should be a summation of the individual coupling interactions (without repetition):

$$\sum_{i=1}^{N_l}\left(\sum_{j=i+1}^{N_l}\left(D_1 * \left(\sum_{m=i}^{j+1} s[m]\right) + D_0\right)\right)$$

Which simplifies to

$$sumK = D_1 * \sum_{i=1}^{N_l}\left(\sum_{j=i+1}^{N_l}\left(\sum_{m=i}^{j+1} s[m]\right)\right) + \left(\frac{N_l(N_l - 1)}{2}\right) * D_0$$

where N_l is the number of layers, D_1 and D_0 are constants based on the trendline (see above) and $s[m]$ is the separation between each layer (see Figure 12). Note: it may be helpful to view the spacings as a triangular matrix instead. In this particular case, summations allow for easier translation to code, so the matrix-form calculations are not shown here.

Which fits into the larger equation as: $L = L_{single} * (N_l + 2 * sumK)$

Note: the thicknesses of the copper layers themselves is omitted from the formula (for legibility) but this will be taken into account in the applications below. For a more detailed example, see the python code below.

```python
couplingConstant_D : tuple[float] = (1.025485443, -0.201166582) # like (D0,D1) where k = D1*s + D0 (1st order polynomial)

outerLayerCopperThickness = 0.035
innerLayerCopperThickness = 0.0152
# dielectricConstants: dict[str,float] = {'None':0.0, 'core' : 4.6, '7628' : 4.6, '2116' : 4.25, '3313' : 4.05} # do not seem to affect things significantly

layerStack : tuple[float] \
                = (outerLayerCopperThickness,
                   0.0994,
                   innerLayerCopperThickness,
                   0.35,
                   innerLayerCopperThickness,
                   0.1088,
                   innerLayerCopperThickness,
                   0.35,
                   innerLayerCopperThickness,
                   0.0994,
                   outerLayerCopperThickness,
                   )

Lsingle = 0.807214519539405 # inductance of a single layer (uH)
N = (len(layerStack)//2)+1   # number of layers

sumOfSpacings = 0.0
for i in range(0, len(layerStack)-2, 2): # Note: this could go to len(layerStack), but the last entry would skip the j for-loop below entirely.
    for j in range(i+1, len(layerStack)-1, 2):
        spacing = sum(layerStack[i:(j+2)]) # total thickness (incuding copper) from layer i to layer (j+1)
        spacing -= (layerStack[i] + layerStack[(j+1)]) / 2 # subtract half the thickness of both copper layers, (so the calculations are done from the centers of the copper layers)
        sumOfSpacings += spacing
        # sumOfCouplingFactors += (couplingConstant_D[1] * spacing) + couplingConstant_D[0] # DEPRICATED. The sumOfSpacings is used instead (along with a triangular number for D[0], see code below)

triangularNumber = (N*(N-1))/2 # triangular number
sumOfCouplingFactors = (couplingConstant_D[1] * sumOfSpacings) + (triangularNumber * couplingConstant_D[0]) # preliminary formula (final formula may include more parameters)
Lmulti = Lsingle * (N + 2*sumOfCouplingFactors) # preliminary formula (final formula may include more parameters)
print(Lmulti)
```

Figure 14: example python code for arbitrary/variable layer spacing coupling calculations.

Using this new formula, the entire dataset (excluding single-layer samples, as those are unaffected) can be reviewed:

PCB name	sample name	Shape	Layers	Turns	Trace width [mm]	Clearance [mm]	Diam. [mm]	layer spacing [mm]	1-layer Predict. Induct. [µH]	Measured Inductance [µH]	new predict. Induct. [µH]	new prediction error [% of measurement]
2-layer PCB		Square	2	9	0.9	0.15	40	0.57	3.945	15.034	15.077	0.3%
	baseline	Circular	2	9	0.9	0.15	40	0.57	3.058	11.866	11.687	1.5%
		Square	2	6	0.9	0.15	40	0.57	2.402	9.092	9.179	1.0%
		Circular	2	6	0.9	0.15	40	0.57	1.855	7.144	7.090	0.8%
		Square	2	9	1.2	0.15	40	0.57	3.033	11.587	11.591	0.0%
		Circular	2	9	1.2	0.15	40	0.57	2.290	8.947	8.752	2.2%
		Square	2	9	0.9	0.3	40	0.57	3.510	13.393	13.416	0.2%
		Circular	2	9	0.9	0.3	40	0.57	2.689	10.415	10.276	1.3%
		Square	2	9	0.9	0.15	30	0.57	2.187	8.205	8.359	1.9%
		Circular	2	9	0.9	0.15	30	0.57	1.643	6.304	6.281	0.4%
		Square	2	5	0.9	0.15	40	0.57	1.866	7.013	7.130	1.7%
		Square	2	8	0.9	0.15	40	0.57	3.455	13.187	13.204	0.1%
		Square	2	11	0.9	0.15	40	0.57	4.804	18.334	18.360	0.1%
		Square	2	14	0.9	0.15	40	0.57	5.700	21.730	21.784	0.2%
		Circular	2	5	0.9	0.15	40	0.57	1.436	5.508	5.487	0.4%
		Circular	2	7	0.9	0.15	40	0.57	2.273	8.775	8.687	1.0%
		Circular	2	8	0.9	0.15	40	0.57	2.677	10.374	10.232	1.4%
		Circular	2	10	0.9	0.15	40	0.57	3.407	13.190	13.021	1.3%
		Circular	2	11	0.9	0.15	40	0.57	3.718	14.409	14.208	1.4%
		Circular	2	12	0.9	0.15	40	0.57	3.985	15.469	15.231	1.5%
		Circular	2	13	0.9	0.15	40	0.57	4.207	16.312	16.078	1.4%
		Circular	2	14	0.9	0.15	40	0.57	4.381	17.003	16.744	1.5%
4-layer PCB	top in1 in	Circular	3	9	0.9	0.15	40	multiple (replaced with stack-up table)	3.058	26.870	26.846	0.1%
		Circular	4	9	0.9	0.15	40		3.058	46.910	46.996	0.2%
		Square	4	9	0.9	0.15	40		3.945	59.380	60.628	2.1%
		Circular	4	6	0.9	0.15	40		1.855	28.220	28.509	1.0%
	top bot	Circular	2	9	0.9	0.3	40		2.689	10.230	10.134	0.9%
	top in2	Circular	2	9	0.9	0.3	40		2.689	10.415	10.389	0.3%
	in1 in2	Circular	2	9	0.9	0.3	40		2.689	10.583	10.643	0.6%
	top in1	Circular	2	9	0.9	0.3	40		2.689	10.611	10.638	0.2%
	top in1 bo	Circular	3	9	0.9	0.15	40		3.058	26.530	26.267	1.0%
6-layer PCB	top bot	Circular	2	8	1	0.1	24		0.807	2.904	2.912	0.3%
	top in1	Circular	2	8	1	0.1	24		0.807	3.224	3.230	0.2%
	top in2	Circular	2	8	1	0.1	24		0.807	3.101	3.111	0.3%
	top in3	Circular	2	8	1	0.1	24		0.807	3.058	3.071	0.4%
	top in4	Circular	2	8	1	0.1	24		0.807	2.947	2.952	0.2%
	top in1 in	Circular	3	8	1	0.1	24		0.807	7.138	7.070	0.9%
	top in1 in	Circular	4	8	1	0.1	24		0.807	12.280	12.446	1.3%
	all except	Circular	5	8	1	0.1	24		0.807	18.956	19.002	0.2%
	all 6 layer	Circular	6	8	1	0.1	24		0.807	26.430	27.012	2.2%
	in1 in2	Circular	2	8	1	0.1	24		0.807	3.137	3.151	0.5%
	in1 in4	Circular	2	8	1	0.1	24		0.807	2.977	2.993	0.5%
	top in2 in	Circular	3	8	1	0.1	24		0.807	6.838	6.753	1.2%
	in2 in3	Circular	2	8	1	0.1	24		0.807	3.236	3.230	0.2%
4-layer PCB		Circular	4	9	0.4	0.1	12		0.509	7.172	7.822	9.1%
		Square	4	9	0.4	0.1	12		0.698	9.567	10.726	12.1%

Table 3: all multi-layer test sample results (using final formula with variable layer-spacing using first D_n constants)

Furthermore, now that variable-spacing calculations are used, we can use the whole dataset (excluding single-layer samples and last 2 samples (12mm diameter)) to recalculate the constants:

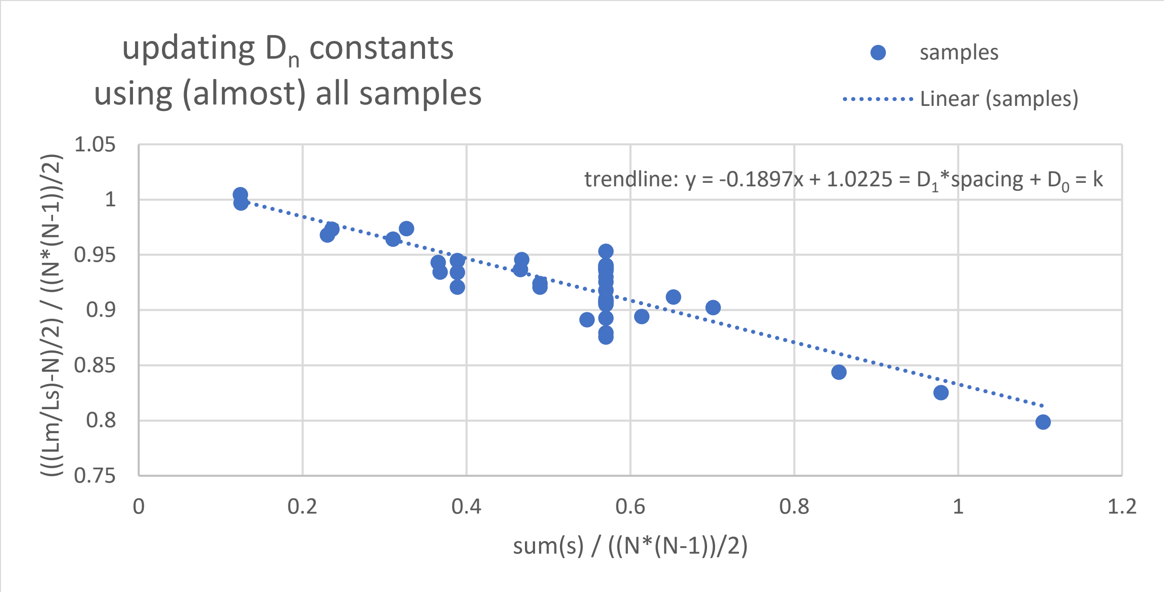


Figure 15: updating D_n constants using (almost) all samples. Note: the axis titles are the formulas used to normalize for any number of layers, which involves using the sum of spacings, and the triangular number of N_{layers} .

Unfortunately, the newly derived constants yield slightly worse results overall:

Comparison metric Regarding prediction error (as % of measurement)	**original formula**	**original formula with K_{turns} fixed**	**my formula (1st D_n constants)**	**my formula (2nd D_n constants)**
average absolute error	19.47%	3.37%	1.25%	1.28%
sum of squared error	187.10%	12.54%	2.77%	2.83%
root mean square of error	20.17%	5.22%	2.45%	2.48%
std. dev. of error	5.33%	4.03%	2.13%	2.14%
max error within dataset	36.30%	18.40%	12.12%	12.25%
max error within dataset, 12mm samples excluded	36.30%	9.94%	2.20%	2.51%

Table 4: statistical comparison of the results of the different formulae/constants

Figure 15 and the results of the 2nd D_n constants in Table 4 indicate that another parameter likely affects the multi-layer coupling in some way. However, given the limited size of the dataset, and the measurement errors inherent to inductance measurements [5], the current dataset is unfit to determine which parameter(s) should be taken into account to correct for this error. Considering this unidentified missing adjustment, the formula & constants obtained from these samples will be exclusively applicable to coils with largely similar design parameters. As is demonstrated by the smaller samples used (see last 2 rows of Table 3), the error of the predicted values increases when the design parameters of the coil deviate significantly from the rest of the dataset. It is possible that several parameters should be taken into account, which would lead to compounding error as multiple parameters deviate simultaneously.

That having been said, the original purpose of this experiment was to improve the analytical predictions of PCB coils, specifically for wireless power transfer (e.g. 'Qi charging') applications. The datasheet of TI's bq51222 wireless power receiver IC mentions that "*The typical choice of the inductance of the receiver coil for a dual mode 5-V solution is between 6 to 8 µH.*" [6]. Notably, similar wireless power ICs from TI list example circuits with coils between ~10 and ~36 [7], particularly in applications with reduced power draw (e.g. ~2.5W max). For coils near this approximate range of inductance values, the formula- and constants derived here should maintain a prediction error below ~10%.

For a refence of how accurate predictions such as this could possibly become, see paper [5], which investigates most of the relevant sources of variance.

Constant	**1st trendline results (limited dataset) (best results)**	**2nd trendline results (larger dataset) (worse results)**	**Unit**
D_0	1.025485443	1.022541055	Unitless (1.0 with currently unexplained offset)
D_1	-0.201166582	-0.189715888	$1/mm$

Table 5: final constants and their units (for reproducibility)

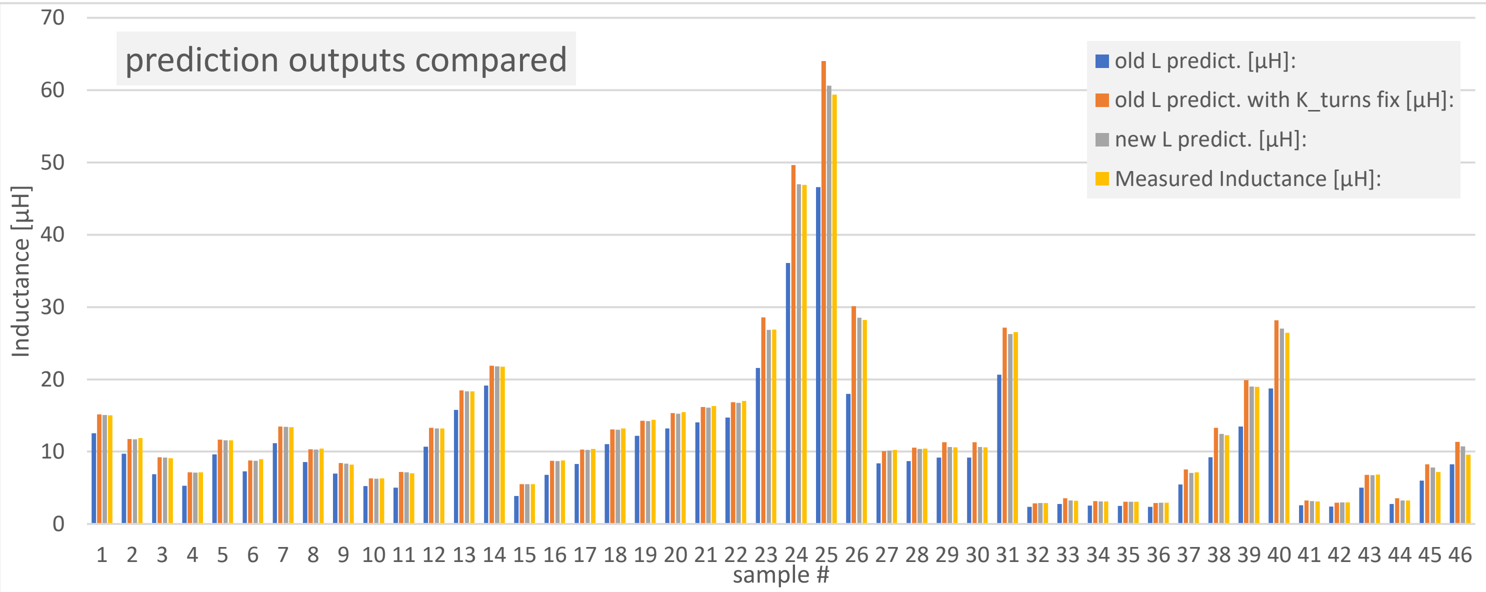


Figure 17: prediction output compared

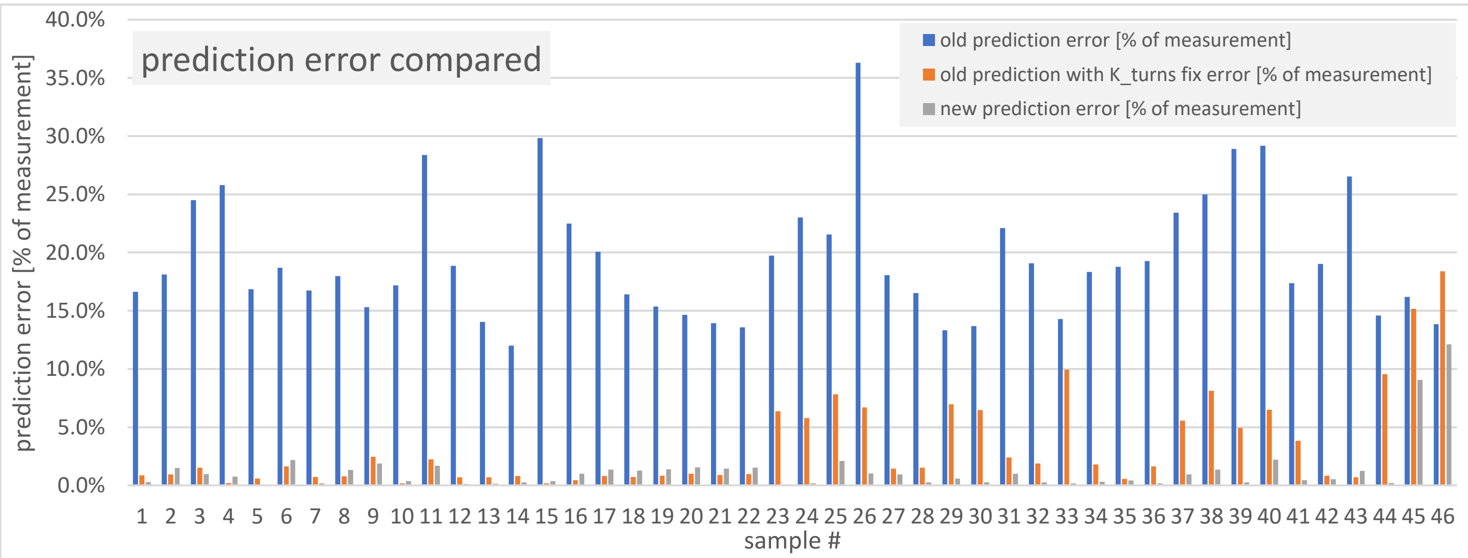


Figure 16: prediction error compared

References

[1] M. d. M. H. S. P. B. a. T. H. L. Sunderarajan S. Mohan, "Simple Accurate Expressions for Planar Spiral Inductances," *EEE JOURNAL OF SOLID-STATE CIRCUITS,* vol. 34, no. 10, pp. 1419-1424, 1999.

[2] A. &. M. L. &. F. C. D. &. A. F. &. C. J. Faria, "A Fast and Precise Tool for Multi-Layer Planar Coil Self-Inductance Calculation," *Sensors,* vol. 21, no. 14, p. 4864, 16 July 2021.

[3] J. Zhao, "A new calculation for designing multilayer planar spiral inductors," *Edn -Boston then Denver then Highlands Ranch Co-,* vol. 55, pp. 37-40, July 2010.

[4] S. K. I. F. S. T. Ashraf B. Islam, "Design and Optimization of Printed Circuit Board Inductors for Wireless Power Transfer System," *Circuits and Systems,* vol. 04, no. 02, pp. 236-244, 18 April 2013.

[5] M. &. B. T. &. L. T. &. P. Y.-W. Noh, "Analysis of Uncertainties in Inductance of Multi-Layered Printed-Circuit Spiral Coils," *Sensors,* vol. 22, no. 10, p. 3815, 18 May 2022.

[6] Texas Instruments, *bq51222 Dual Mode 5-W (WPC v1.2 and PMA) Single Chip Wireless Power Receiver,* 2016.

[7] Texas Instruments, *INTEGRATED WIRELESS POWER SUPPLY RECEIVER, Qi (WIRELESS POWER CONSORTIUM) COMPLIANT,* 2012.

All the mathematics used here are applied in code: https://github.com/thijses/PCBcoilGenerator

As well as in the Excel spreadsheet with the measurement data: measurement data in excel.xlsx