Classical Calculation for Mutual Inductance of Two Coaxial Loops in MKS Units

Kurt Nalty

www.kurtnalty.com
Austin, TX 78750 USA

March 6, 2011 *

Abstract

The mutual inductance of two parallel coils is a simple combination of complete elliptic integrals. This note walks through this calculation using MKS units.

Flux Φ and the Vector Potential $\mathbf{A}$

For a general loop of wire, the flux encompassed by that wire is

$$\begin{aligned}
\Phi &\approx \mathbf{B}A \\
&= \int \mathbf{B} \cdot d\mathbf{S} \\
&= \int \nabla \times \mathbf{A} \cdot d\mathbf{S} \\
&= \oint_s \mathbf{A} \cdot d\mathbf{s}
\end{aligned}$$

where s is length along the curve and S is area enclosed by the curve.

If the field $\mathbf{A}$ is due to a current I flowing in path $\mathbf{t}$, we have the flux cutting path $\mathbf{s}$ as

$$\begin{aligned}
\Phi &= \oint_s \mathbf{A} \cdot d\mathbf{s} \\
&= \oint_s \left(\oint_t \frac{\mu I d\mathbf{t}}{4\pi r} \right) \cdot d\mathbf{s} \\
&= \oint_s \oint_t \frac{\mu I}{4\pi r} d\mathbf{t} \cdot d\mathbf{s}
\end{aligned}$$

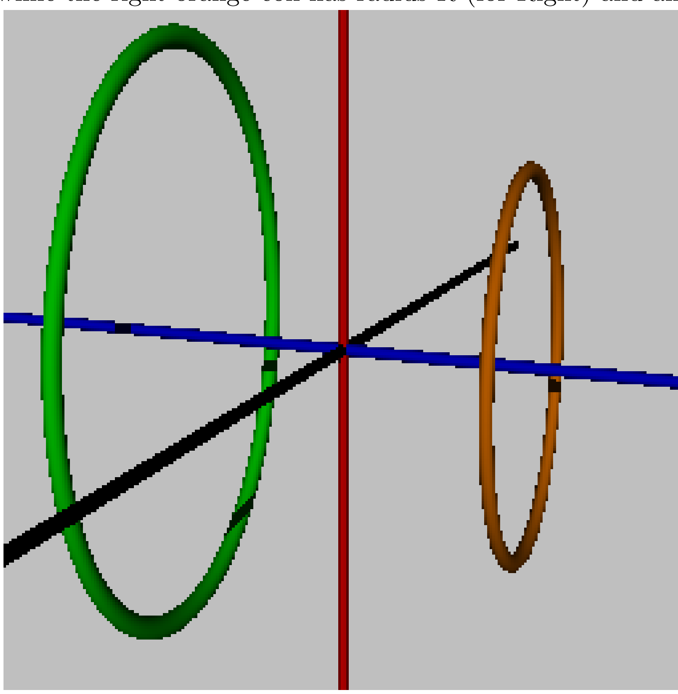
*I appreciate Clifford Curry pointing out a missing divisor of 4π.

Mutual Inductance

Mutual inductance and flux are related by $\Phi = MI$. We thus have the Neumann formula

$$M = \mu \oint_s \oint_t \frac{d\mathbf{t} \cdot d\mathbf{s}}{4\pi r}$$

For this exercise, separate the two coils along the z axis, at $\pm l/2$. For generality, let the left green coil have radius L (for Left), and angle parameter ϕ, while the right orange coil has radius R (for Right) and angle parameter θ.



We have

$$\begin{aligned} d\mathbf{s} &= L\left(-\sin\phi \mathbf{a}_x + \cos\phi \mathbf{a}_y\right) d\phi \\ d\mathbf{t} &= R\left(-\sin\theta \mathbf{a}_x + \cos\theta \mathbf{a}_y\right) d\theta \\ d\mathbf{s} \cdot d\mathbf{t} &= LR\left(\sin\phi \sin\theta + \cos\phi \cos\theta\right) d\phi d\theta \\ &= LR\left(\cos\left(\phi - \theta\right)\right) d\phi d\theta \end{aligned}$$

The positions and separations are

$$
\begin{aligned}
\mathbf{r}_{\text{left}} &= L\cos\phi\mathbf{a}_x + L\sin\phi\mathbf{a}_y - \frac{l}{2}\mathbf{a}_z \\
\mathbf{r}_{\text{right}} &= R\cos\theta\mathbf{a}_x + R\sin\theta\mathbf{a}_y + \frac{l}{2}\mathbf{a}_z \\
\Delta\mathbf{r} &= (L\cos\phi - R\cos\theta)\,\mathbf{a}_x + (L\sin\phi - R\sin\theta)\,\mathbf{a}_y - l\mathbf{a}_z \\
r^2 &= L^2 + R^2 + l^2 - 2LR\,(\cos\phi\cos\theta + \sin\phi\sin\theta) \\
&= L^2 + R^2 + l^2 - 2LR\cos(\phi - \theta) \\
r &= \sqrt{L^2 + R^2 + l^2 - 2LR\cos(\phi - \theta)}
\end{aligned}
$$

We now can write our expression for the mutual inductance.

$$
\begin{aligned}
M &= \mu \oint_s \oint_t \frac{d\mathbf{t}\cdot d\mathbf{s}}{4\pi r} = \frac{\mu}{4\pi}\oint_s \oint_t \frac{d\mathbf{t}\cdot d\mathbf{s}}{r} \\
&= \frac{\mu}{4\pi}\oint_\phi \oint_\theta \frac{LR\,(\cos(\phi-\theta))\,d\phi d\theta}{\sqrt{L^2 + R^2 + l^2 - 2LR\cos(\phi - \theta)}} \\
&= \frac{\mu}{4\pi}\oint_\phi \left(\oint_\theta \frac{LR\,(\cos(\phi-\theta))\,d\theta}{\sqrt{L^2 + R^2 + l^2 - 2LR\cos(\phi - \theta)}}\right) d\phi \\
&= \frac{\mu}{4\pi}\oint_\phi \left(\oint_\theta \frac{LR\,(\cos(\theta-\phi))\,d\theta}{\sqrt{L^2 + R^2 + l^2 - 2LR\cos(\theta - \phi)}}\right) d\phi
\end{aligned}
$$

where the last step uses the even nature of the cosine to absorb a minus sign.

During the inner integration over θ, ϕ is kept constant. We now do a change of variable,

$$
\begin{aligned}
\gamma &= \theta - \phi \\
d\gamma &= d\theta \quad \text{inside the } \theta \text{ integral}
\end{aligned}
$$

We can simplify the mutual inductance expression.

$$
\begin{aligned}
M &= \frac{\mu}{4\pi}\oint_\phi \left(\oint_\gamma \frac{LR\cos\gamma d\gamma}{\sqrt{L^2 + R^2 + l^2 - 2LR\cos\gamma}}\right) d\phi \\
&= \frac{\mu}{4\pi}\oint_\phi d\phi \left(\oint_\gamma \frac{LR\cos\gamma d\gamma}{\sqrt{L^2 + R^2 + l^2 - 2LR\cos\gamma}}\right) \\
&= \frac{\mu}{2}\oint_\gamma \frac{LR\cos\gamma d\gamma}{\sqrt{L^2 + R^2 + l^2 - 2LR\cos\gamma}}
\end{aligned}
$$

This integral is known in terms of the complete elliptic functions K and E. The reference integral is

$$
\begin{aligned}
\oint \frac{\cos\theta d\theta}{\sqrt{a - b\cos\theta}} &= \frac{4\sqrt{a+b}}{b}\left[\left(1 - \frac{\beta^2}{2}\right)K(\beta) - E(\beta)\right] \\
\text{where } \beta &= \sqrt{\frac{2b}{a+b}}
\end{aligned}
$$

We thus have

$$
\begin{aligned}
a &= \frac{L^2 + R^2 + l^2}{L^2 R^2} \\
b &= \frac{2}{LR} \\
\beta &= \sqrt{\frac{2b}{a+b}} \\
M &= 2\mu\frac{\sqrt{a+b}}{b}\left[\left(1 - \frac{\beta^2}{2}\right)K(\beta) - E(\beta)\right]
\end{aligned}
$$

Sample calculation with two 1 meter diameter coils, 1 meter apart.

$$
\begin{aligned}
L &= 0.5 \\
R &= 0.5 \\
l &= 1.0 \\
a &= \frac{L^2 + R^2 + l^2}{L^2 R^2} = 24 \\
b &= \frac{2}{LR} = 8 \\
\beta &= \sqrt{\frac{2b}{a+b}} = 0.7071 \\
\mu &= 4E - 7\pi \\
K(0.7071) &= 1.8541 \\
E(0.7071) &= 1.3506 \\
M &= 7.1041E - 8
\end{aligned}
$$

Comparison to Simple Integrator

A comparison of closed formula to numerically summed piecewise modelling follows.

```
#include <stdio.h>
#include <math.h>

// calculated mutual inductance of two parallel coils,
// 1 meter in diameter, 1 meter apart, 1 turn


#define nseg 10
#define pi 3.1415925
```

```
typedef struct  {
double x;
double y;
double z;
} vector;

double dot(vector a, vector b)
{
return (a.x*b.x + a.y*b.y + a.z*b.z);
}


double mu_prime(vector sensor_coord, vector sensor_tangent)
{
double theta; // angle along path, cylindrical coordinates
double dtheta; // differential angle along path
double r=0.5; // radius of loop = 1 m diameter
double z_source= 0.5; // position of source loop
double z_sensor=-0.5; // position of sensor loop
double s; // path length along coil
double ds; // differential path length
double mag_deltar; // difference between source and sensor
double mu; // mu_prime
double d_mu; // incremental mu_prime
vector source_coord, source_tangent,deltar;
int nturns = 1; // number of turns
long i,j;

s = 0.0;
mu = 0.0;

for (i=0; i<nseg*nturns; i++) { // integrate along source path
dtheta = 2.0*pi/nseg;
theta = dtheta*i;
ds = r*dtheta;
s += ds;

//  do the contribution from inner radius of trace

source_coord.x = r*cos(theta);
source_coord.y = r*sin(theta);
source_coord.z = 0.5; // plane

source_tangent.x = -sin(theta);
source_tangent.y =  cos(theta);
source_tangent.z = 0;
```

```
deltar.x = source_coord.x - sensor_coord.x;
deltar.y = source_coord.y - sensor_coord.y;
deltar.z = source_coord.z - sensor_coord.z;

mag_deltar = sqrt(dot(deltar,deltar));
d_mu = dot(source_tangent, sensor_tangent)/mag_deltar;
mu += d_mu*ds;

}

return (mu);
}


double mu(void) // get mutual inductance
{
double theta; // angle along path, cylindrical coordinates
double dtheta; // differential angle along path
double r=0.5; // radius of loop = 1 m diameter
double z_source= 0.5; // position of source loop
double z_sensor=-0.5; // position of sensor loop
double s; // path length along coil
double ds; // differential path length
double mag_deltar; // difference between source and sensor
double mu; // mu_prime
double d_mu; // incremental mu_prime
vector source_coord, source_tangent,deltar;
vector sensor_coord, sensor_tangent;
int nturns = 1; // number of turns
long i,j;

s = 0.0;
mu = 0.0;

for (i=0; i<nseg*nturns; i++) { // integrate along sensor path
dtheta = 2.0*pi/nseg;
theta = dtheta*i;
r = 0.5;

sensor_coord.x = r*cos(theta);
sensor_coord.y = r*sin(theta);
sensor_coord.z = -0.5; // plane

sensor_tangent.x = -sin(theta);
sensor_tangent.y = cos(theta);
```

```
sensor_tangent.z = 0;


printf("."); if((i%80) == 0) printf("\n"); // activity
ds = r*dtheta;
mu += ds*mu_prime(sensor_coord, sensor_tangent);
s += ds;
}
printf("\ns = %g\n",s);
mu *= 1.0e-7; // scale by mu_0/(4 pi)
return (mu);
}

int main(void)
{
double M;

M = mu();
printf("M = %e\n\n",M);

double L,R,l,a,b,beta,mu,K,E,M;
L = 0.5;
R = 0.5;
l = 1.0;
a = (L*L + R*R + l*l)/(L*L*R*R);
b = 2.0/(L*R);
beta = sqrt(2.0*b/(a+b));
mu = 4.0e-7*pi;
K = 1.8541;
E = 1.3506;

M = 2.0*mu*(sqrt(a+b)/b)*((1.0 - 0.5*beta*beta)*K - E);
printf("a = %f\n",a);
printf("b = %f\n",b);
printf("beta = %f\n",beta);

printf("M = %e\n\n",M);


}

/*
Run time results
.
.........
s = 3.14159
```

```
M = 7.092998e-08

a = 24.000000
b = 8.000000
beta = 0.707107
M = 7.104169e-08


Nice Agreement.

*/
```