这里建一个粗略的模型进行分析，模型假设如下：
1）弹丸中垂直于轴线的截面上磁场强度相等，即等于轴线上的磁场强度。
2）由于脉冲电流产生的外磁场非常强，因此弹丸在磁场中迅速达到磁饱和，饱和磁感应强度为 J_m
3）弹丸简化为圆柱形
4）由于这类电磁炮的效率在 1%数量级，因此弹丸对电路的影响非常小，在这里忽略不计。
1.受力分析

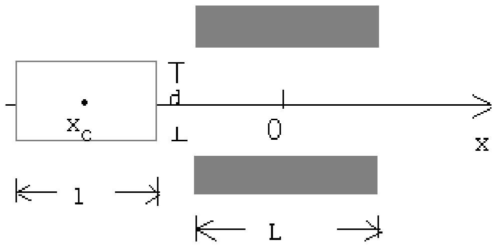


图 1-2-1：简化模型图

简化模型如图 1-2-1。设在 t 时刻，x 位置线圈产生的磁场强度为 H(x,t)，弹丸位置中点为 x_c，弹丸长度为 l. 则线圈的前端面与后端面的极化磁荷面密度分别为：

$$\begin{aligned}\sigma_1 &= -J_m \\ \sigma_2 &= J_m\end{aligned} \tag{1.1}$$

因此弹丸受到的合力为：

$$F = \frac{\pi \cdot d^2}{4} J_m (H(x_c + \frac{l}{2}, t) - H(x_c - \frac{l}{2}, t)) = m \frac{d^2 x_c}{dt^2} \tag{1.2}$$

2.等效电路分析

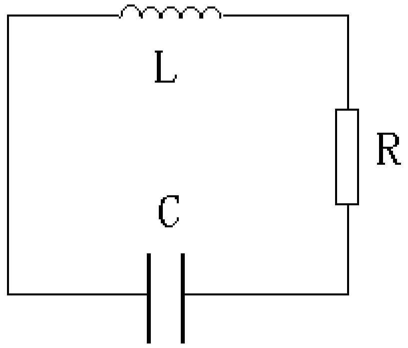


图 1-2-2：等效电路图

这里脉冲电流的来源为电容的瞬间放电，因此，整个电路可以等效为 LRC 电路因此有：

$$LC\frac{d^2U}{dt^2} + RC\frac{dU}{dt} + U = 0 \tag{1.3}$$

该方程的解为：

$$u = u_0 e^{-\frac{R}{2L}t} \cos(\frac{t}{\sqrt{LC}} + \varphi_0) \tag{1.4}$$

即：

$$i(t) = \frac{RCu_0}{2L} e^{-\frac{R}{2L}t} \cos(\frac{t}{\sqrt{LC}} + \varphi_0) + \frac{Cu_0}{\sqrt{LC}} e^{-\frac{R}{2L}t} \sin(\frac{t}{\sqrt{LC}} + \varphi_0) \tag{1.5}$$

（初始条件为 i(0)=0）

3.线圈附近的磁场分布

对于半径为 r_i 的电流环，其磁场强度分布为：

$$H_{i0}(x) = \frac{i \cdot r_i^2}{2(r_i^2 + x^2)^{\frac{3}{2}}} \tag{1.6}$$

对于半径为 r_i，导线直径为 d_0，匝数为 N 的线圈，其磁场分布为：

$$H_i(x)=\sum_{k=-[\frac{N}{2}]}^{[\frac{N}{2}]} H_{i0}(x-kd_0) \qquad (1.7)$$

对于 m 层线圈，若其内径为 R，则其磁场强度分布为：

$$H(x)=\sum_{i=1}^{m} H_i(x) \qquad (1.8)$$

其中第 i 层的半径 r_i 满足：

$$r_i=R+(i-\frac{1}{2}d) \qquad (1.9)$$

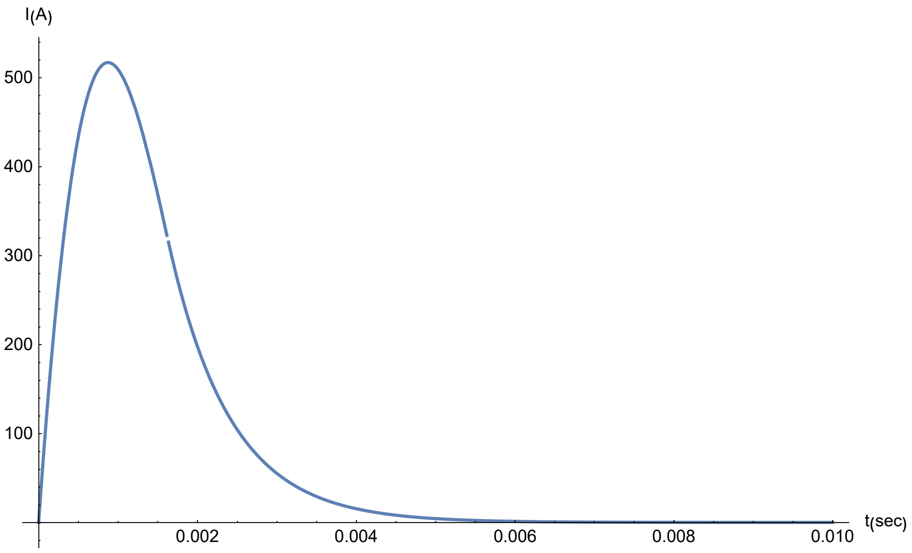
至此，模拟计算所需的量已经全部得出，仅需要代入数值用计算机求解。

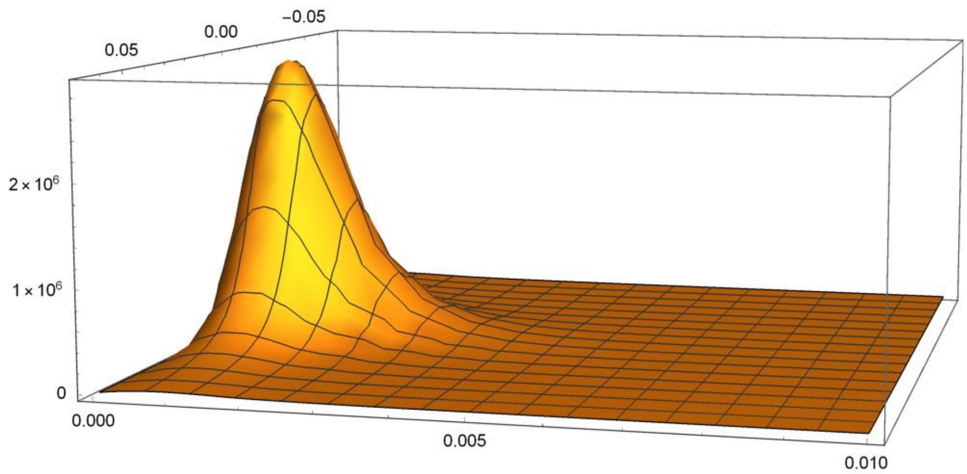
Mathematica 代码：

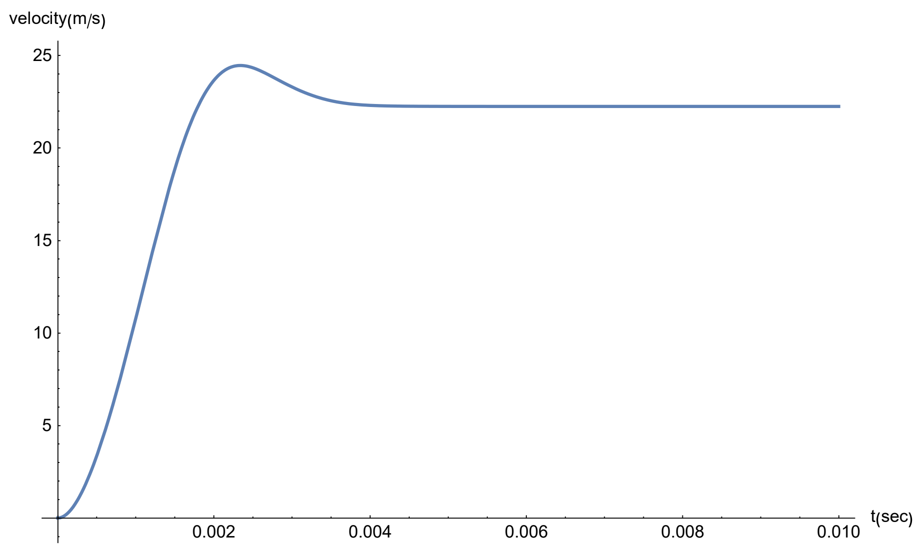
```
L=313.328*10-6 ; (*电感*) c=1600*10-6;(*电容*)
R=0.4;(*电阻*) r=0.01;(*线圈内径*)
d0=0.001;(*铜丝直径*) d=0.006;(*亚克力内直
径*)(*电流用 J 表示*)
l=0.035;(*弹丸长度*) m=0.007;(*子弹质量*)
Rd=0.4;(*二极管近似电阻*)
Vi=0;(*初速度*)
Xi=-0.03;(*初始位置*)


(*计算电流*)
{Itt}=Simplify[-c*U'[t1]/.
                    DSolve[{L*c*U''[t1]+R*c*U'[t1]+U[t1]0,U[0]400,U'[0]0},U,t1]];
{Utt}=Simplify[U[t1]/.DSolve[{L*c*U''[t1]+R*c*U'[t1]+U[t1]0,
     U[0]400,U'[0]0},U,t1]];
It1[t_]:=Itt/.t1→t;(*LRC 振荡*)
Ut1[t_]:=Utt/.t1→t;
{t3,t2}=x/.Solve[Ut1[x]0];(*解出电压为 0 时的 t*)
                    -Rd
It2[t_]:=It1[t2]*Exp ____*t-t2;(*LR 放电*)
                     L
It[t_]:=Piecewise[{{It1[t],t≤t2},{It2[t],t≥t2}}];(*分段合成后的电流*)
Plot[It[t],{t,0,0.01},AxesLabel→{"t(sec)","I(A)"}]
H[x_,t_]:=
 8                      1422
 
i=1k=-13  (1 (2 * ((r + (i - 1/2) *d0 +x-k*d0 2 ) 3/2 )) * It[t] * (r + (i - 1/2) * d0)
Plot3D[H[x,t],{x,-
              1/4       2                                   1/2
                                  0.07,0.07},{t,0,0.01},Plot
                                  Range→All] l
F=NDSolve π*d  *1.8* Hxc[t]+_,t-Hxc[t]- ,t m*xc''[t],
                          2
   xc[0]Xi,xc'[0]Vi,xc,{t,0,0.01};
Plot[Evaluate[D[xc[t],t]/.F],{t,0,0.01},PlotRange→All,
 AxesLabel→{"t(sec)","velocity(m/s)"}]
```

Solve::ifun : Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. ⍰

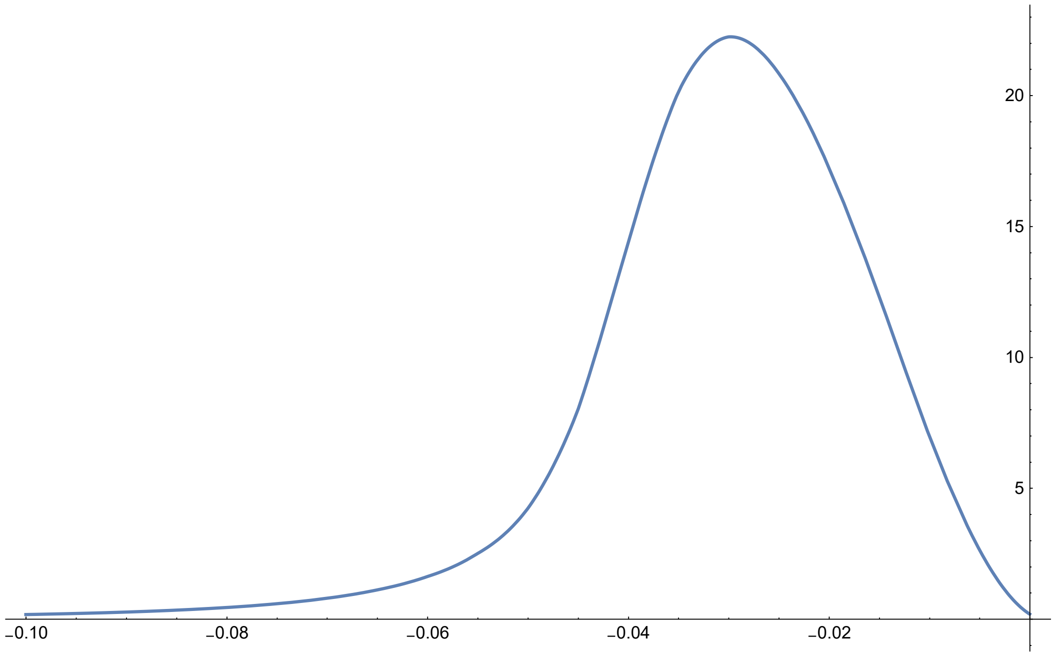




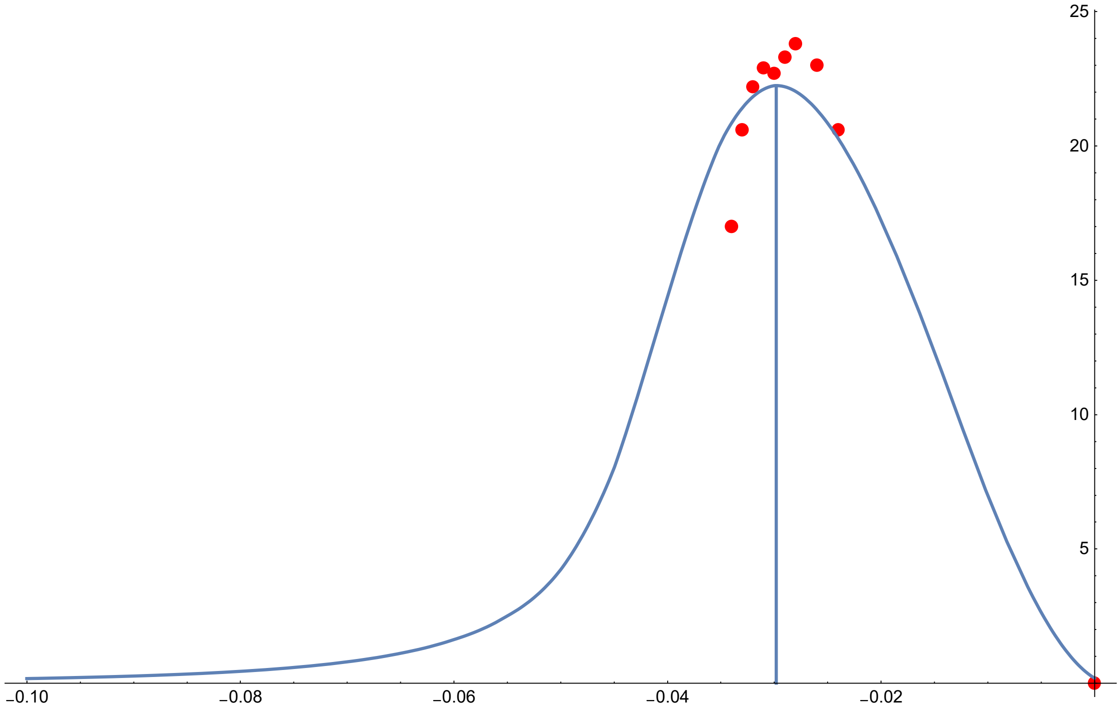


(*末速与初始位置分析*)

```
data1=
                                                      _             (             _               _    )
                                                      1                      ²*1.8*H[xc[t]+l,t]-H[xc[t]-l,t] 
  Flatten[Table[xc'[0.01]/.NDSolve[[π*d
                                                                 42                            2
        m*xc''[t],xc[0]a,xc'[0]Vi],xc,{t,0,0.01}],{a,-0.12,0,0.005}]];
data2=Table[a,{a,-0.12,0,0.005}]; data={data2,data1}//Transpose;
Va=Interpolation[data];
Plot[Va[a],{a,-0.1,0}]
```




```
data5={{-0.034`,17.`},{-0.033`,20.6`},
             {-0.032`,22.2`},{-0.031`,22.9`},{-0.03`,22.7`},{-0.029`,23.3`},
        {-0.028`,23.8`},{-0.026`,23.`},{-0.024`,20.6`},{0,0}};
Maxf={x/.FindMaximum[Va[x],{x,-0.1,-0.001}][[2]],
   FindMaximum[Va[x],{x,-0.1,-0.001}][[1]]};
Show[
 ListPlot[data5,PlotStyle→{Red,PointSize[0.012]}],
 Plot[Va[a],{a,-0.1,0}],
 ParametricPlot[{Maxf[[1]],y},{y,0,Maxf[[2]]}], PlotRange→All
]
```



(*red points are actual ejecting speed measured by photogate while the blue curve is the predicting line*)

装置成品图：