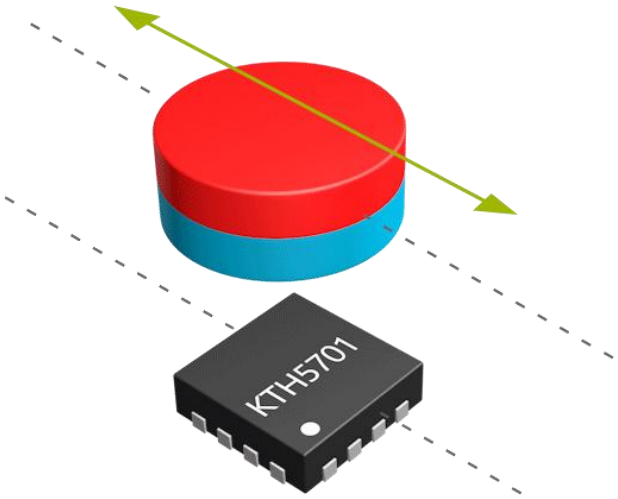
键盘磁轴应用笔记



关于本文档

范围和目的

本文简要介绍了昆泰芯3D磁传感器在磁轴键盘应用的检测算法和实测数据。

目录

1 位移计算原理和测试评估

1.1 位移计算公式分析

首先根据实际的磁轴结构和芯片及磁铁进行磁仿真，结果如下图2所示，YZ平面磁场B与磁轴位移之间呈非线性关系。

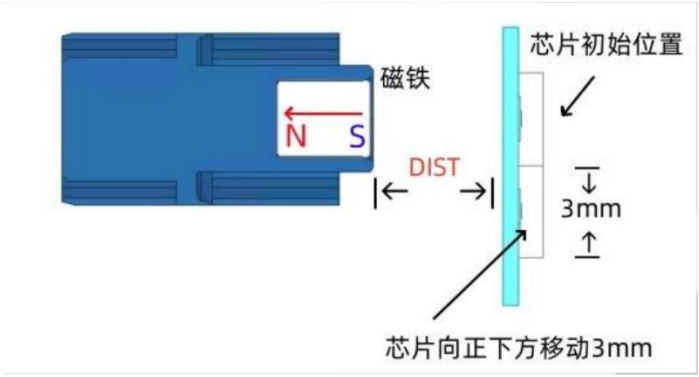


图1 磁轴与芯片的相对位置关系

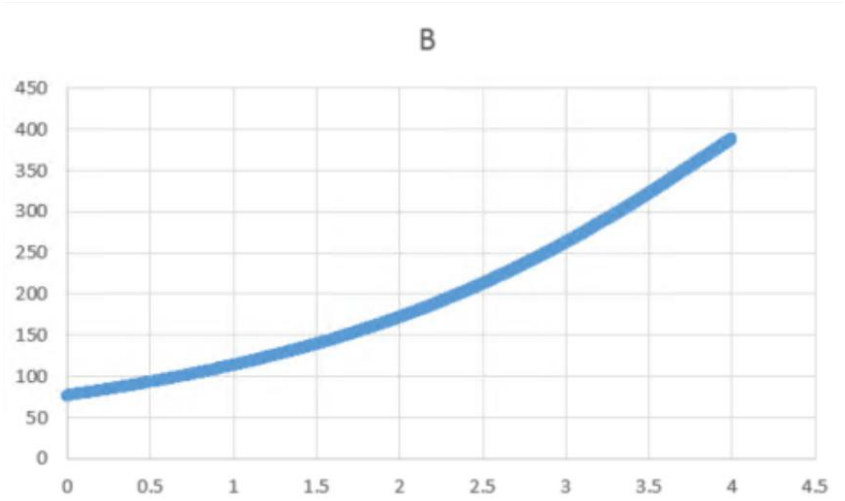


图2 仿真YZ平面磁场B值与位移的输出曲线

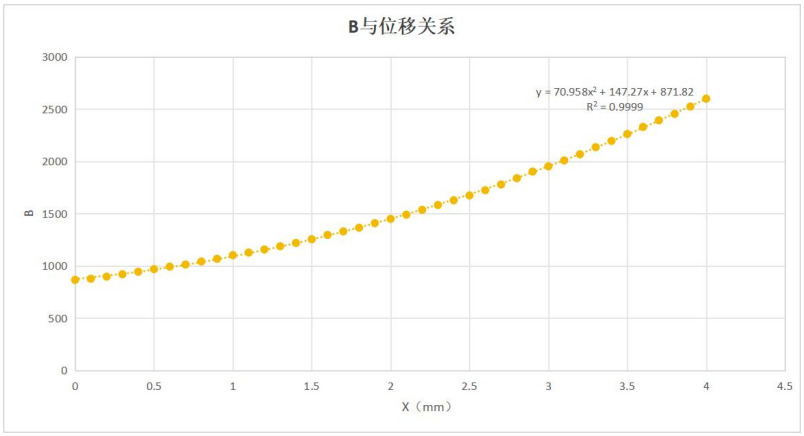


图3 实测YZ平面磁场B值与位移的输出曲线

对实测数据进行拟合分析可以发现，磁场B与位移之间呈现二次多项式关系，如果由B值拟合线性位移，需要计算二次多项式的三个系数，这样校准时就需要测量3个点的数据，对后续实际的生产与校准带来不便，我们对该曲线进行再次变换，如下图4所示位移和磁场之间呈现幂对数关系，并且只有两个系数，可以通过最小与最大行程对应的B值，解算出该系数。

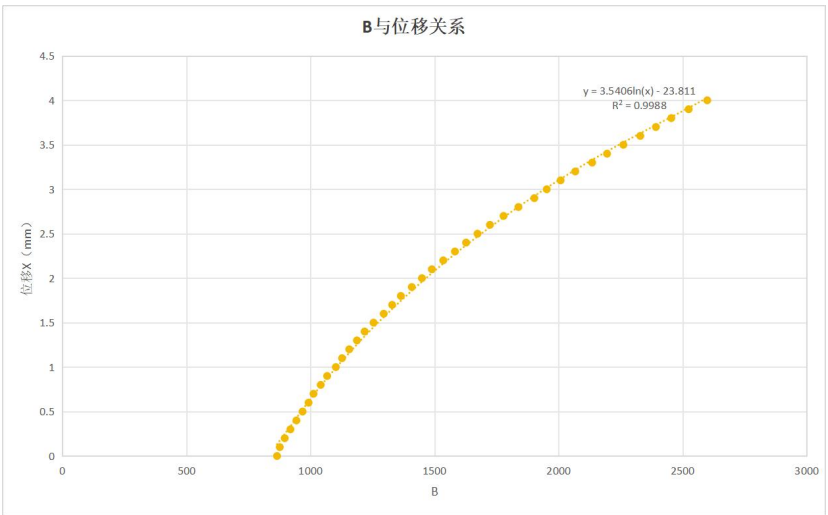


图4 对换坐标系后实测YZ平面磁场B值与位移的输出曲线

1.2 轴体线性位移实测

通过图5所示测试设备，我们对轴体进行了步进0.1mm 的位移测量，图6与图7分别是轴体样品1和2的实测结果，从误差曲线可以看到，采用上述方式的拟合计算方式，可以把最大线性误差控制在0.2mm以内。

图5 磁轴位移测试设备

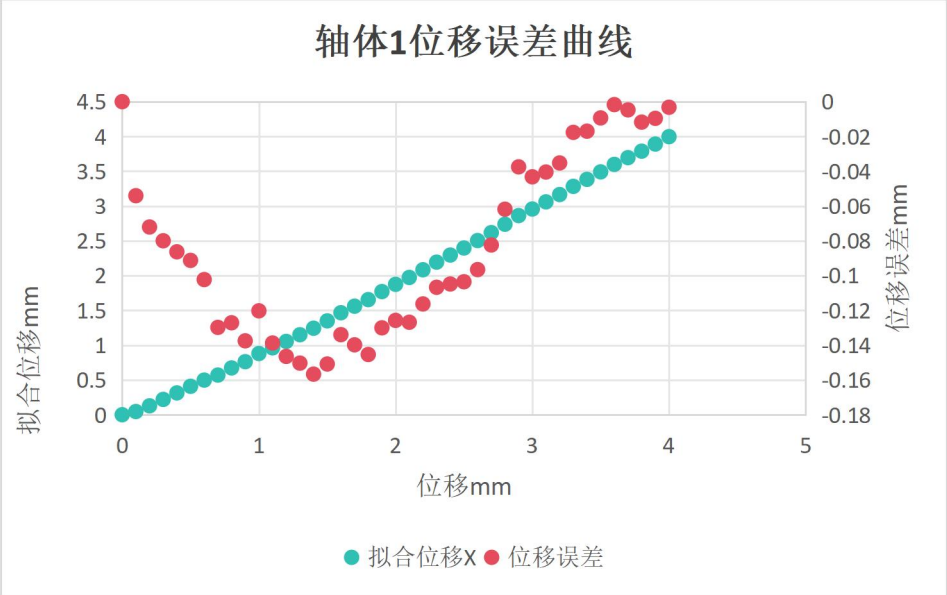


图6 轴体1实际位移和拟合位移误差曲线

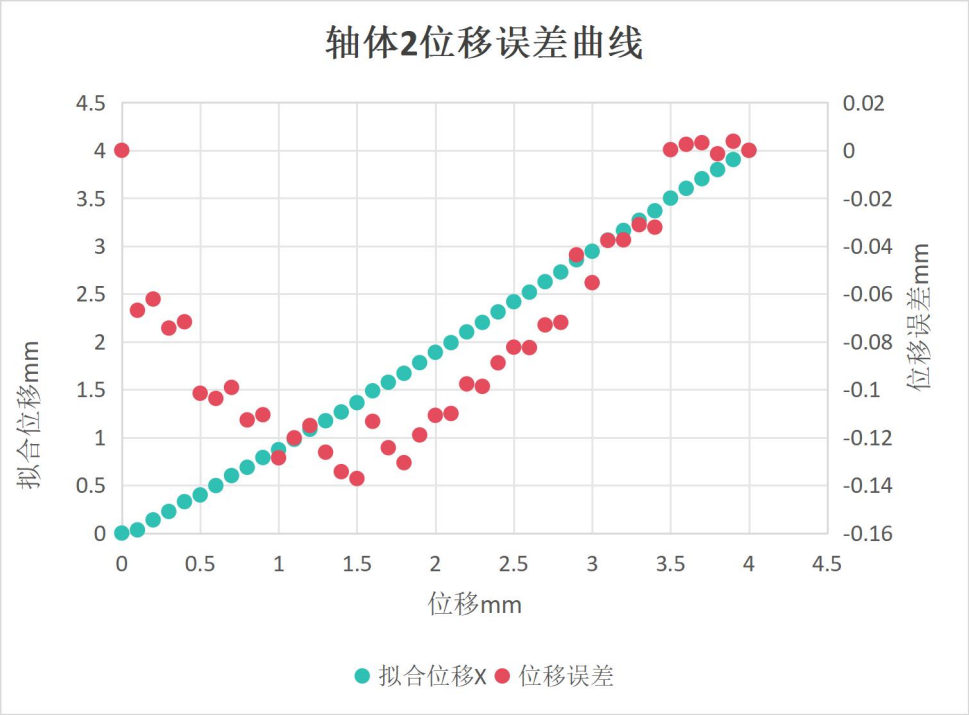


图7 轴体2实际位移和拟合位移误差曲线

对位移误差曲线分析可以发现，误差与实际位移呈现三角函数关系并且参数相对比较固定，如果对位移精度要求更高的话，可以再拟合位移的基础上进行二次补偿，最终可以使得线性位移误差小于0.1mm。

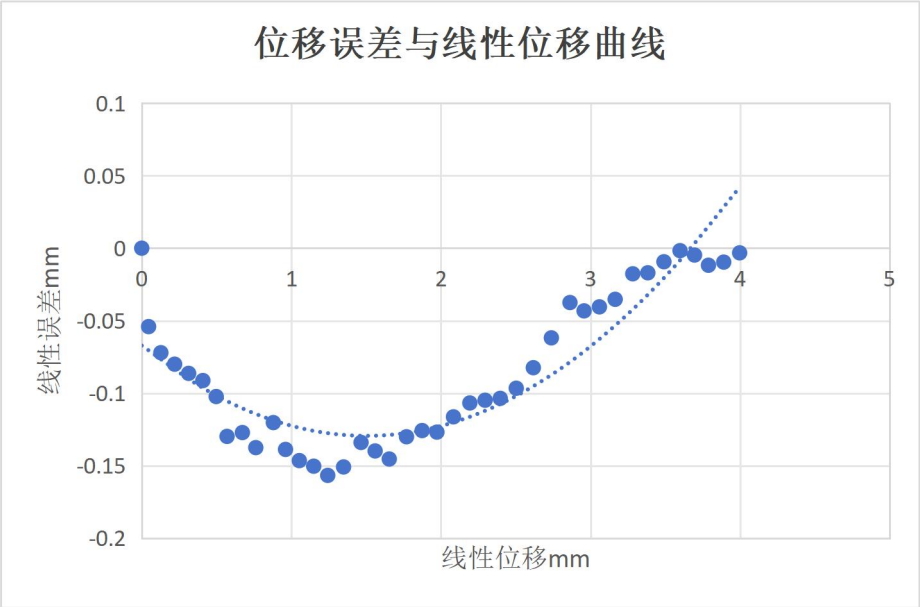


图8 位移误差和线性位移曲线

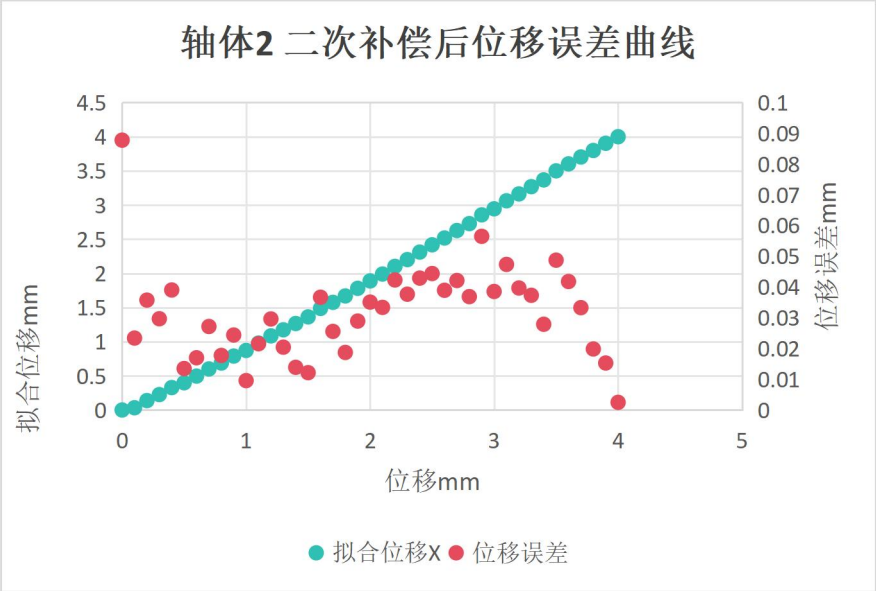


图9 轴体2 二次补偿后线性位移与误差

1.3 位移噪声实测

对轴体各个线性位置进行多次测量采样，评估噪声对位移测量的影响，结果如下图所示，结果显示，当前配置下，死区需要设置在0.1mm以上，否则会受噪声干扰，如果需要更小的死区设置，需要在软件端进一步滤波处理。

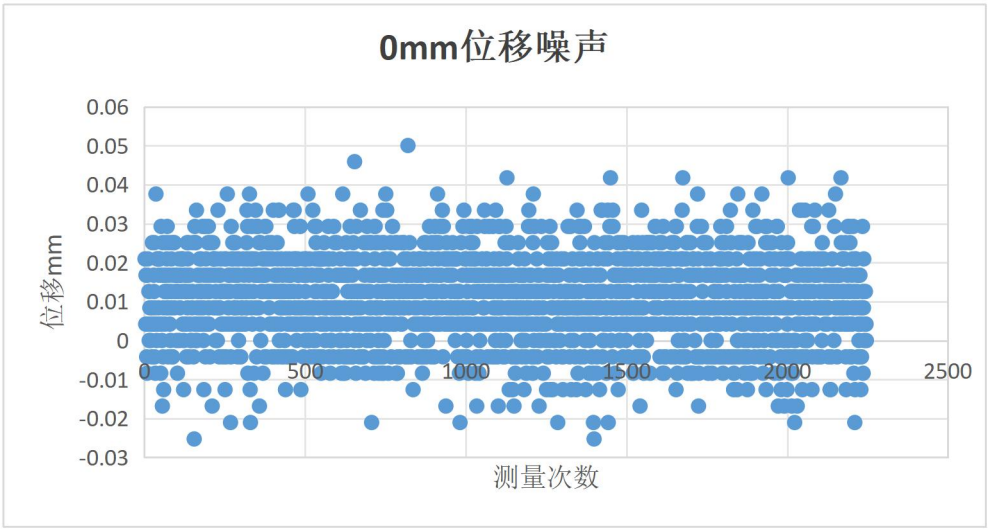


图9 0mm位移噪声

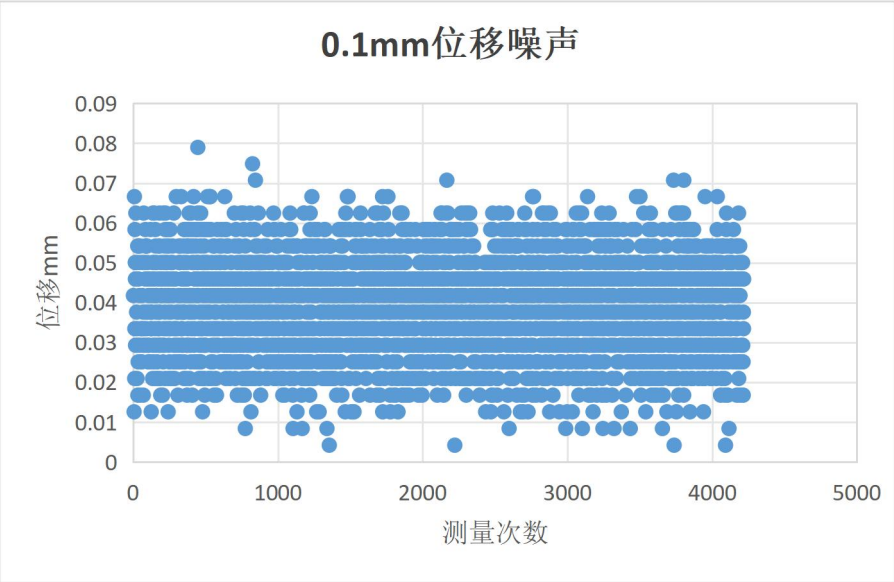


图10 0.1mm位移噪声

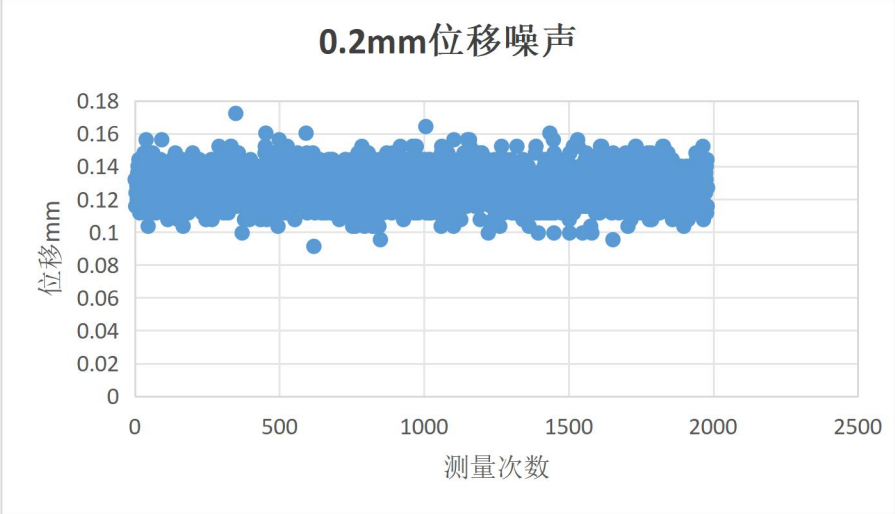


图11 0.2mm位移噪声

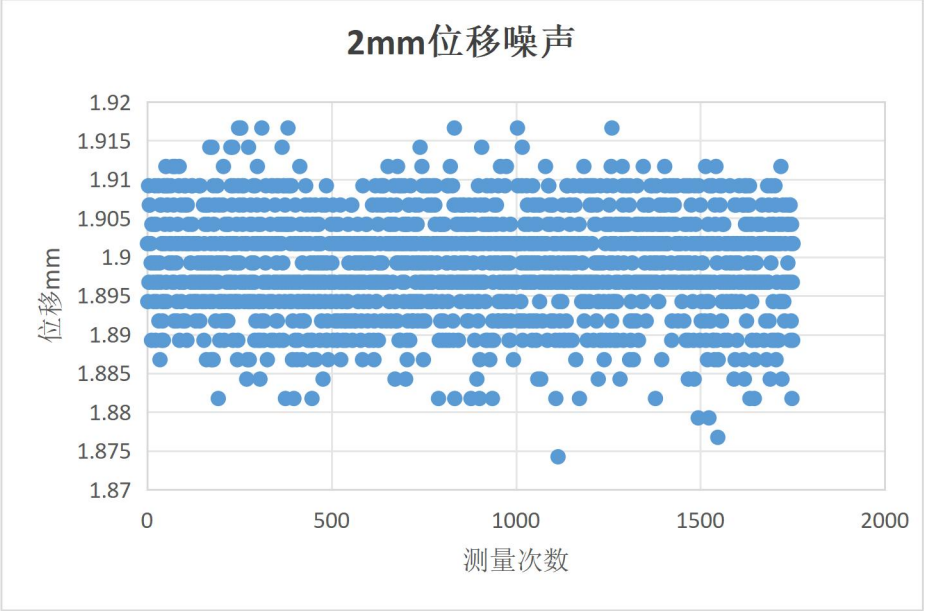


图12 2mm位移噪声

2 算法实现与代码示例

如上文所述，厂家及用户在使用之前，需要对当前每个工位键轴与传感器的组合进行参数校准，以得出计算式中的参数a值与参数b值。这需要用户仅提供两个参数，以实现第一步修调。当按键未按下时，记录当前KTH5701输出磁场B值记为B1， 当按键按到底时，记录KTH5701输出磁场B值记为B2。（采集这两个值时，为避免误差，可以取多次平均值）将B1、B2带入 DistanceOutputInitial 函数得到参数a、b的值，实现B1、B2与两点位置的映射关系。

```
/*
*函数名：void DistanceOutputInitial(uint16_t B_startvalue,uint16_t B_endvalue,double *parameter_a,double *parameter_b)
*输入参数：uint16_t B_startvalue ->按键未按下的B值，例：d'850
*uint16_t B_endvalue ->按键按下的B值，例：d'2900
*输出参数：double *parameter_a ->参数a地址
*double *parameter_b ->参数b地址
*返回值：
*函数作用：根据首尾位置B值计算校准参数a、b
*/

void DistanceOutputInitial(uint16_t B_startvalue,uint16_t B_endvalue,double *parameter_a,double *parameter_b)
{
  *parameter_a = (END_POSITION - START_POSITION)/(log(B_endvalue) - log(B_startvalue));
  *parameter_b = START_POSITION - *parameter_a*log(B_startvalue);
}
```

图13 DistanceOutputInitial a、b参数初始化函数

DistanceOutputInitial 函数中，校准点1位置为按键未按下时的位置，校

准点2位置为按键按至底部时的位置，键程为4mm，若有修改，需调整START_POSITION、END_POSITION宏定义数值。

```
//按键开始位置
#define START_POSITION 0
//按键末位置
#define END_POSITION 4
```

图14 映射位置修改

在得到a、b两个校准参数之后，根据diastance = a *ln(B)+b公式，将KTH5701读取出的B值代入上式，可以很轻松地得到当前diastance 位置。经过此方式校准后得到测量位置与实际位置最大误差为0.15mm。

```
distance1 = log(B_VALUE)*trimparameter_a + trimparameter_b; //计算当前按键位置
```

图15 位置计算

此外，使用上述校准方法，通过对比测量位置与实际位置的误差关系可以看出，上述校准误差在使用不同磁轴时保持同样的变化规律。并且误差值基本符合sinx函数的变化特征，厂家在出厂校准时，可以根据选择的不同磁轴，对distance进行二次校准。示例代码中使用DistanceTrim进行校准后，得到测量位置与实际位置最大误差为0.05mm（排除0mm位置）。

```
/*
* 函数名：DistanceTrim(double distance)
* 输入参数：double distance -> 一次校准后的输出值
* 输出参数：
* 返回值：二次校准结果
* 函数作用：根据当前计算位置，进行二次修调。（最大绝对位置误差为0.05mm）
*/

double DistanceTrim(double distance)
{
  return 0.15*sin((distance+1)/1.6);
}
```

图16 二次校准

另外，为提高按键在全磁场范围的低噪声，提高稳定度，示例代码中提供了一种软件滤波算法，使用滤波算法可以在按键未按下时得到更小的死区，但是这会损失一定的响应速度，使用时应综合考虑该影响。

```c
/*
 * 函数名：float kalmanFilter_B(float inData)
 * 输入参数：inData -> 数据参数
 * 输出参数：
 * 返回值：float
 * 函数作用：对输入B值进行滤波
*/
float kalmanFilter_B(float inData)
{
    static float prevData=0;
    //其中p的初值可以随便取，但是不能为0（为0的话卡尔曼滤波器就认为已经是最优滤波器了）
    static float p=0.01, q=P_Q, r=M_R, kGain=0;
    p = p+q;
    kGain = p/(p+r);
    inData = prevData+(kGain*(inData-prevData));
    p = (1-kGain)*p;
    prevData = inData;
    return inData;
}
```

图17 软件滤波

```c
/**
  * @brief  向寄存器初始化
  * @param
  *
  * @retval 寄存器初始化是否成功
  *         返回REG_OK表示寄存器初始化成功
  *         返回REG_FAIL表示寄存器初始化失败
  */
uint8_t KTH57XXRegInitial(void)
{
    uint8_t staCheck;


  KTH57XXWriteRegister (0x5631,0x1c);//osr dig
  KTH57XXWriteRegister (0x8000,0x1e);//angle output

  KTH57XXWriteRegister (0x400,0x1d);//temp trim
  KTH57XXWriteRegister (32768,0x14);//offset trim
  KTH57XXWriteRegister (32768,0x15);//offset trim
  KTH57XXWriteRegister (32768,0x16);//offset trim

    staCheck = REG_OK;
    return staCheck;
}
```

图18 开启灵敏度温度补偿

上述示例代码提供实现思路，对于MCU算力有一定要求，如果对响应速度有较大影响，可以考虑校准完成后，输出一组插值点数据存在Flash，之后采用查表插值的方式进行快速计算，节约系统运算时间，提高响应速度。