## GPR模型用于轮廓预测估计思路

1、预先选定一定量的理论数据作为预测的初始点

2、读入实际轮廓的位置信息

3、滑动GPR预估点位置，由已知历史数据做预测进行估计

4、根据预测的误差进行控制器设计

高斯过程回归。对于输出训练数据集D={(xi,yi)}=(X,y)。其中xi为d维输入向量，X为d x n维输入向量，y为相应的输出标量。已知输入数据集合X，可构成一个随机变量集合{f(x1), f(x2), ,f(xn)}，且具有联合高斯分布，该高斯过程全部统计数字特征可由均值函数和协方差函数确定，并指定合适的核函数。

轨迹预测时，将测试轨迹数据集输入预测模型得到预测输出轨迹。

对于N条具有相同运动模式的轨迹，利用高斯过程建模，典型运动模式的轨迹概率模型为：。

其中，，分别为均值和协方差矩阵。

## 预想工作思路步骤

1、将历史轨迹预处理后转化为轨迹矢量存储

2、利用高斯过程回归模型对历史数据进行训练以得到预测模型

3、根据输入的新轨迹数据预测移动过程中最可能的轨迹，以求出轮廓误差

（针对同一条轨迹时，可以将运动模式看做单一运动，高斯过程回归模型即可。）

## 基于高斯模型的轨迹预测算法实现

输入：训练轨迹数据集 Dtrain = {T1, T2, ... , Tn};

测试轨迹数据集 Dtest = {T1\*, T2\*, ... , Tm\*};

输出：轨迹预测误差均值RMSE。

1、T\* = {s1, s2, ... , sn}; // 已知轨迹序列

2、model\_Inital = Kmeans(Dtrain); // Kmeans算法初始化模型

3、m = Train(Dtrain,model\_inital); // 迭代训练生成新模型

4、k = n-d; // 求取预测步数，d表示观测轨迹点数量

5、for i=1 to k

6、 *p* = *Predict*(*m*); // 预测连续k步未来轨迹点

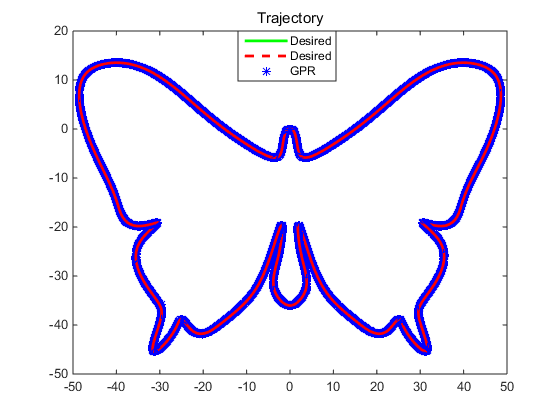
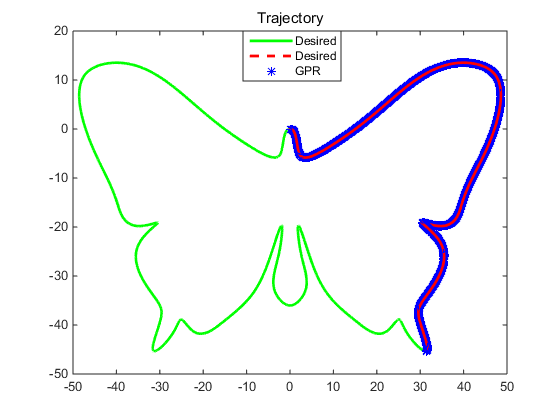
7、 e[i] = CalRMSE(p,pr); // 计算每步的预测误差，pr表示真实轨迹点

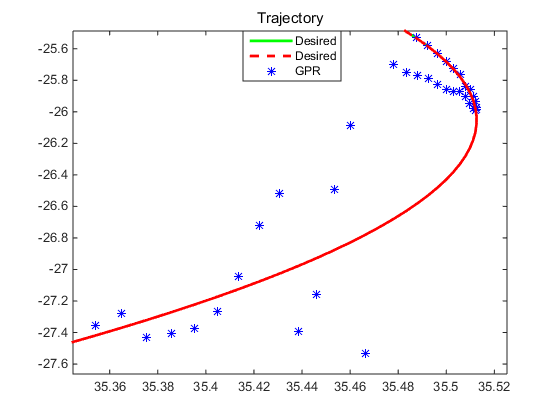
8、end

9、RMSE = Sum(e)/k; // 求取误差的均值

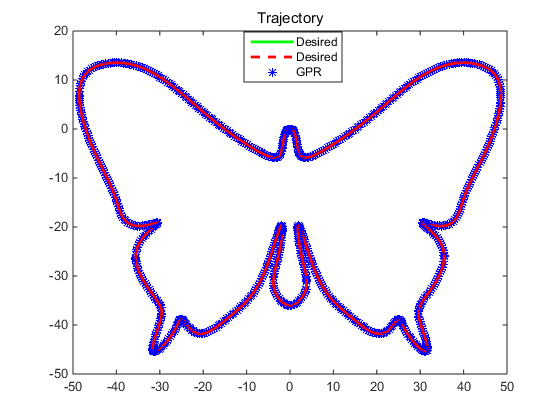
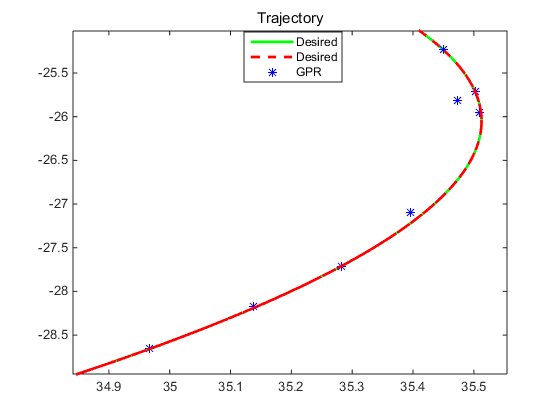
## 假设滑动过程中以规划轨迹为历史状态进行预测

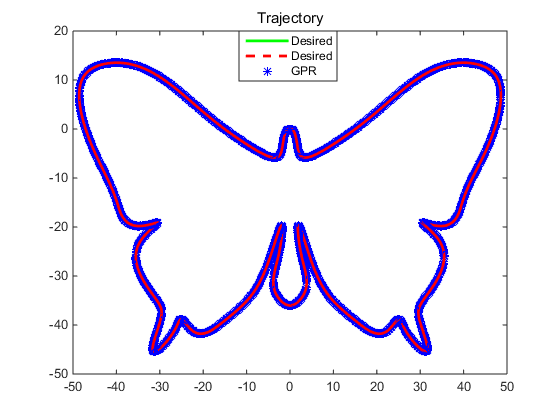
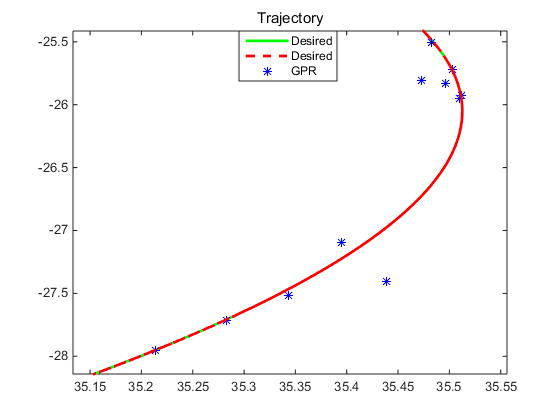
因为计算量过大，导致计算延迟尤为严重，是以在较少采样点时观看局部

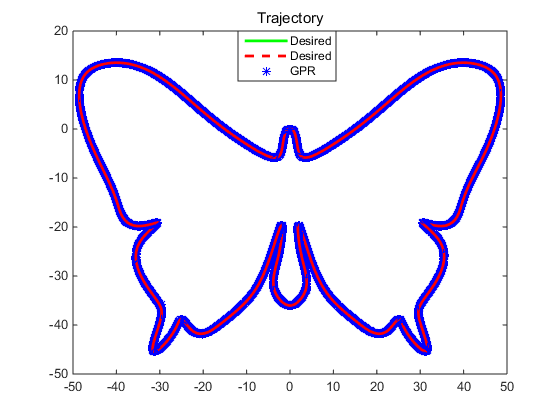
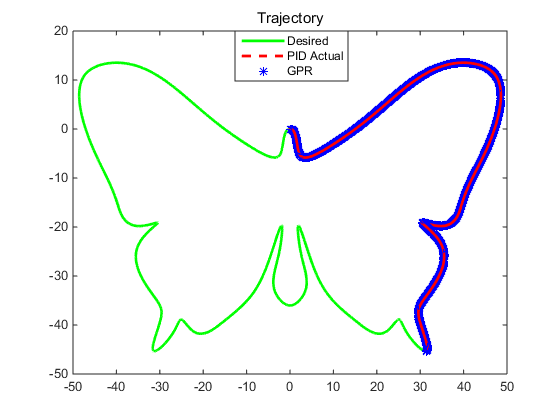


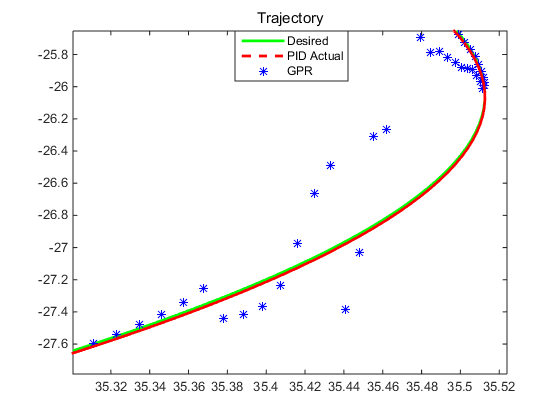
因为计算量过大，导致计算延迟尤为严重，是以试着以扩大采样周期为代价换取时间，预测周期由1ms增加至10ms与5ms时，其对应的计算时间由123.36s减小到3.27s和7.24s

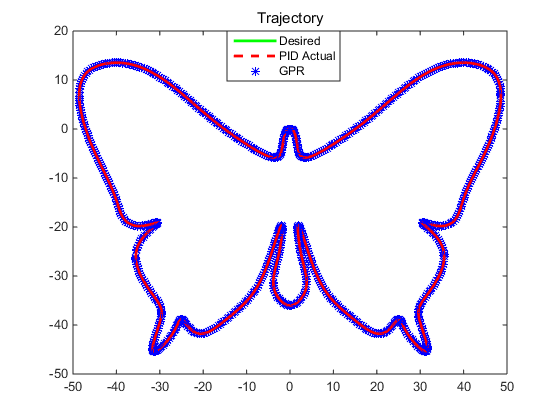
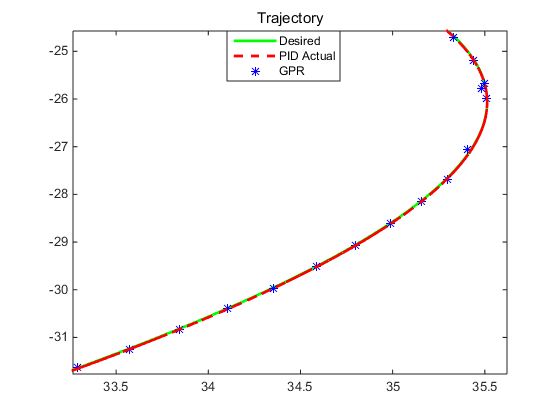
 

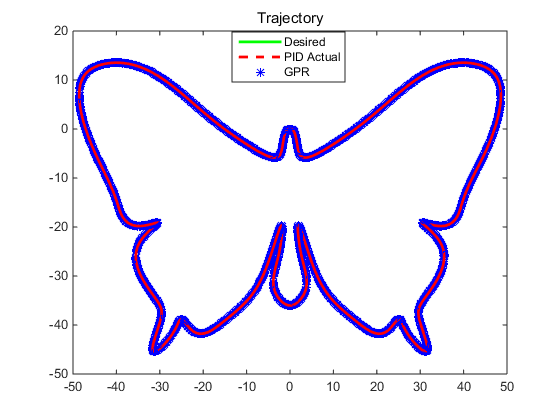
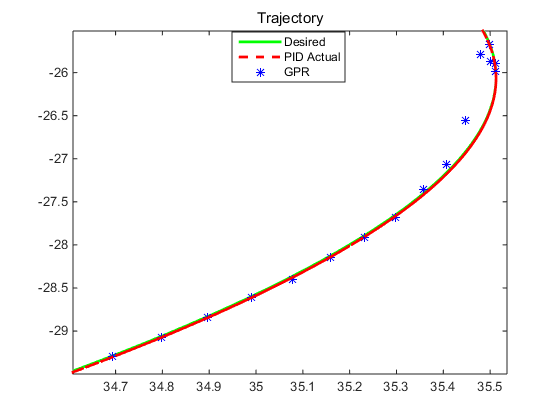
## 假设滑动过程中以预先PID运动轨迹为历史状态进行预测



因为计算量过大，导致计算延迟尤为严重，是以试着以扩大采样周期为代价换取时间，预测周期由1ms增加至10ms与5ms时，其对应的计算时间由123.36s减小到3.33s和7.64s

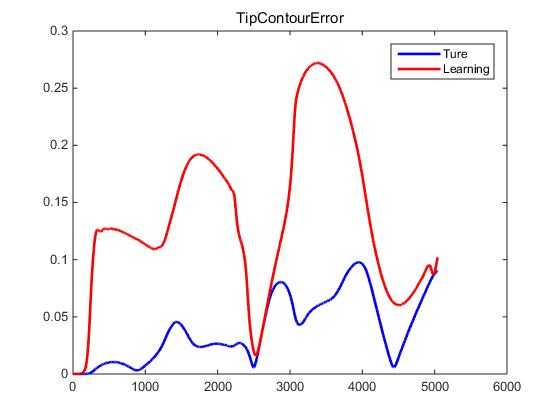
 

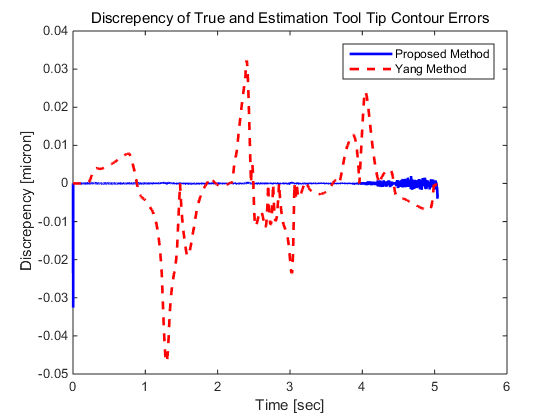
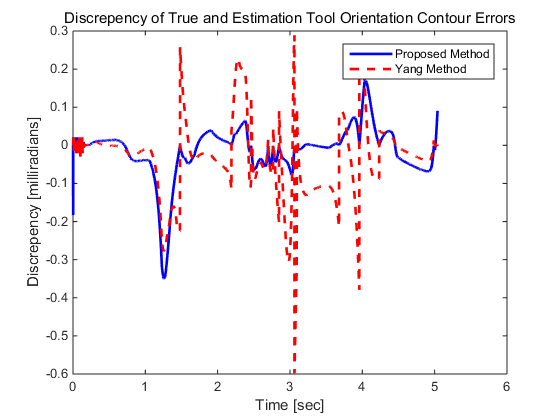
由图可以看出，以10ms采样历史数据点时，其运行时间大大缩减，且在小范围内避免了奇异点的出现。

## 由离线数据训练并学习

训练数据点数目：n=3, coe = -1

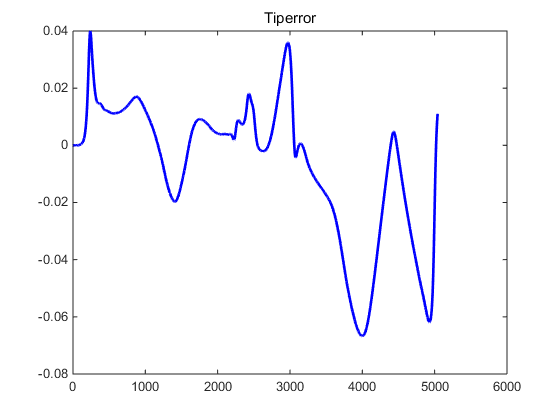
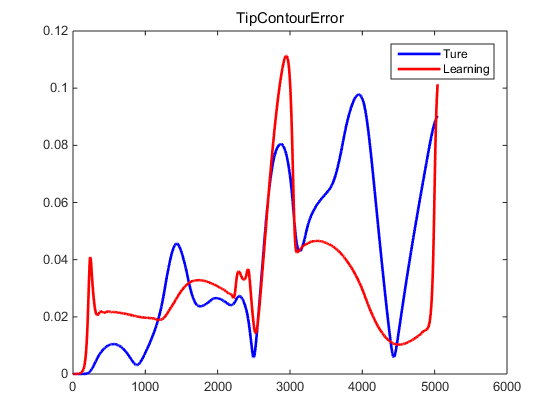


对比于之前的所采用的误差估计方法其误差分别为：

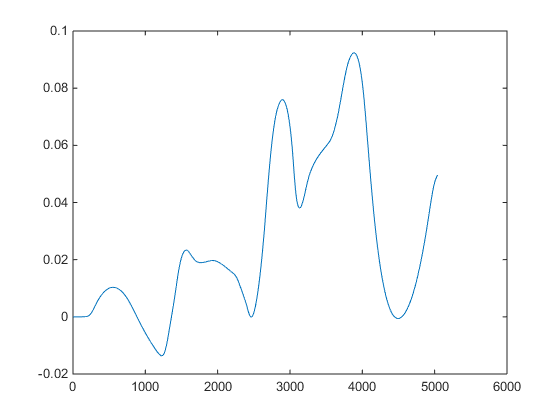
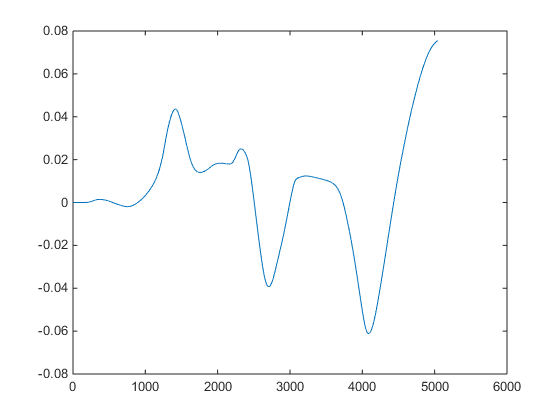
因为高斯核函数可以自由选取，是以分别选取不同的系数进行实验:

训练数据点数目：n=3, coe = -255

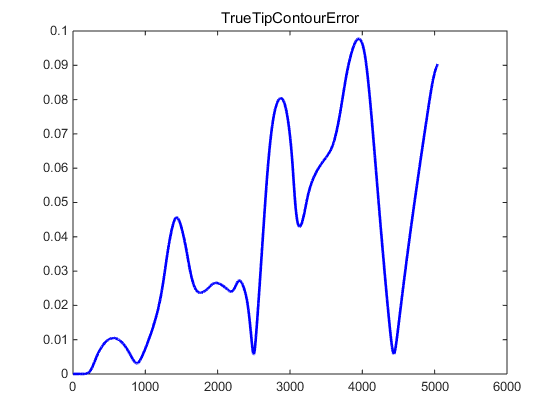
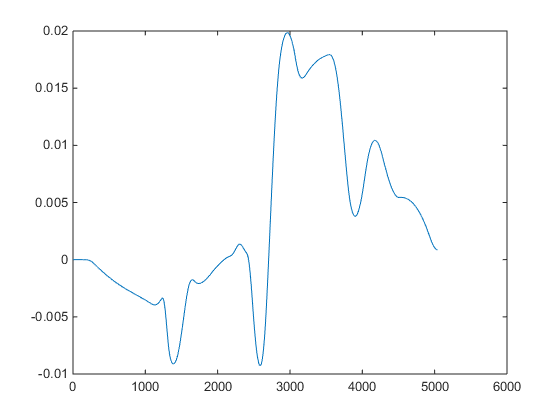


2017-06-06

X轴的轮廓误差： Y轴轮廓误差：

Z轴轮廓误差： 多轴轮廓误差：



学习轮廓误差：

