```
void ZNOL()
{
if(enableCount==true) // start the Fast Autotune
  {
  count++; //start counting seconds based on interval(1000 msec)
  }
    yA=TR_C_Filtered; //filtered value of temperature
    slopeA=50*(yA-yA_old)/(sampleTime/1000); //amplify the slope calculations by 50 for plotting purposes only
    slopeA_Filtered=slopeA_filterFunction(5, 1,slopeA, 1000);
    if(slopeA_Filtered>50) //prevents huge initial filtered slope when plotting is started
        slopeA_Filtered=50;
    if(slopeA>50)//prevents huge initial slope //prevents huge initial filtered slope when plotting is started
        slopeA=50;
     if(enablePeakDetect==true)
     {
     findPeak(slopeA_Filtered); //for step response, find the peak value of slope which yields inflection point
     }
}
//*************ZNOL Functions******************
// Filter slope with a small filtering time constant
float slopeA_filterFunction(float timeConstant, float processGain, float blockIn, float intervalTime)
{
float static blockOut;
blockOut=blockOut+(intervalTime/1000/(timeConstant+intervalTime/1000))*(processGain*blockIn-blockOut);
return blockOut;
}

void findPeak(float currentValue) // current value is the slope of the step response curve
{
  if (readOnce==true) //read Temp at start of Step change
  {
    startingValue=yA;
  }

  if (currentValue>peakValue) // Detect peak value to find inflection point (slope starts to decrease)
  {
    peakValue=currentValue;
    YA_inflect=yA; //YA_inflect is the value of temperature at the inflection point
    readOnce=false;
    findPID(slopeA, count, YA_inflect, startingValue, m_Size_Percent, &ZProp, &ZInteg, &ZDeriv);
  }
}
// these functions will be called continuously as the response rises generating erroneous results until the inflection
//point. At the inflection point the operator clicks the remote which automatically loads the P,I, and D into the controller
void findPID(float peakSlope, int timeT2, float yA_Peak, float yA_Start, float M_step, float *PB, float *I, float *D)
{
  float R, L, timeT1; // timeT1 is base of triangle formed by peakSlope and (yA_Peak-yA_Start)
  R=(0.24*peakSlope); //reaction rate in %/min, (6*peakSlope/5)/500*100
  timeT1=(yA_Peak-yA_Start)/(peakSlope/50.0); // remove amplification factor of 50 and do the geometry
  //timeT2 is start of step to point of inflection, timeT1 is time from intersection of tangent to point of inflection
  L=timeT2-timeT1-2; //Effective Dead Time
  *PB=1.38833*L*R/M_step; //83.3*(L/60)*R/M_step
 *I=2.0*L;
  *D=0.5*L;
}
```