This iLogic code is from Luke Davenport's Cadline Blog – it allows user to optimize the length of all the lines in a sketch (excluding construction lines) to achieve a target loop length.

```
' ************************** START OF iLOGIC CODE ***********************************************

'Note - I must subscribe to Luke Davenport's blog on Cadline Community
'http://www.cadlinecommunity.co.uk/Blogs/LukeDavenport/Default.aspx
'And also maybe I'll follow him on twitter @LukeCadline

'© Luke Davenport 2013

Imports  Inventor.UnitsTypeEnum

Parameter.Quiet = True
MultiValue.Quiet = True

'Set a reference to the active part document
Dim oDoc As PartDocument
oDoc = ThisApplication.ActiveDocument

'Check the Document type is a part and not an assembly or drawing
If oDoc.DocumentType <> kPartDocumentObject Then
    MessageBox.Show("This rule can only be run in a part document!", "Cadline iLogic", _
    MessageBoxButtons.Ok,MessageBoxIcon.Exclamation, _
    MessageBoxDefaultButton.Button1)
    Return
End If

'Check that only one sketch exists in the part
If oDoc.ComponentDefinition.Sketches.Count < 1 Then
    MessageBox.Show("You need a sketch in the part to run this rule!", "Cadline iLogic", _
    MessageBoxButtons.Ok,MessageBoxIcon.Exclamation, _
    MessageBoxDefaultButton.Button1)
    Return
Else If oDoc.ComponentDefinition.Sketches.Count > 1 Then
    MessageBox.Show("There must be only a single sketch in the part to run this rule!" _
    , "Cadline iLogic",MessageBoxButtons.Ok,MessageBoxIcon.Exclamation, _
    MessageBoxDefaultButton.Button1)
    Return
End If

'Get string for document units
Dim StrDocUnits As String
If oDoc.unitsofmeasure.LengthUnits = 11272 Then
    StrDocUnits = "inches"
Else
    StrDocUnits = "mm"
End If

'Define stuff
Dim oDef As PartComponentDefinition
oDef = oDoc.ComponentDefinition
Dim oParams As Parameters = oDef.Parameters
Dim oUserParams As UserParameters = oParams.UserParameters
Dim oParam As Parameter

'Create a parameter that holds names of all other parameters
Dim CadlineParameterNames As UserParameter  = oUserParams.AddByValue("CadlineParameterNames" _
,oParams.Item(1).Name, kTextUnits)
Dim MyParams as New ArrayList
```

```
'Format String to construct lines in form
Dim Format As String = "{0,-16} {1,14}"
Dim Newline As String

'Construct lines in form
For aa = 1 To oParams.Count-1
    If oParams.Item(aa).Units = "mm" Or oParams.Item(aa).Units = "in" Then
        Newline = String.Format(Format, (oParams.Item(aa).Name), _
        (oParams.Item(aa).Value)*10 & " " & (oParams.Item(aa).Units))
        MyParams.Add(Newline)
    End If
Next

'Add all parameter names to multivalue parameter list
MultiValue.List("CadlineParameterNames")= MyParams

AskAgain:
'Ask user what parameter should be modified
StrParam = InputListBox("What parameter would you like to modify?", _
MultiValue.List("CadlineParameterNames"),"None", Title := "iLogic", ListName := _
"Existing Parameters")

If StrParam = "" Then
    Doh = MessageBox.Show("Please select a parameter to modify!", "Cadline iLogic", _
    MessageBoxButtons.OKCancel,MessageBoxIcon.Information)
    If Doh = 2 Then
        Return
    End If
End If

'Store value of chosen parameter in case the sketch breaks
For bb = 1 To oParams.Count-1
    If StrParam.Contains(oParams.Item(bb).Name) Then
        StrParamOld = oParams.Item(bb).Value
        ParamNumber = bb
    End If
Next

'-----------------------------------------------

'Measure all lines and arcs and circles in sketch
Dim oSketch As Sketch = oDef.Sketches.Item(1)

Dim oSketchArcs As SketchArcs = oSketch.SketchArcs
Dim oSketchArc As SketchArc
Dim ArcLength As Double = 0

For Each oSketchArc In oSketchArcs
    If oSketchArc.Construction = False Then
        ArcLength = ArcLength + oSketchArc.Length*10
        InventorVb.DocumentUpdate()
    End If
Next

Dim oSketchLines As SketchLines = oSketch.SketchLines
Dim oSketchLine As SketchLine
Dim LineLength As Double = 0

For Each oSketchLine In oSketchLines
    If oSketchLine.Construction = False Then
        LineLength = LineLength + oSketchLine.Length*10
    End If
Next

Dim oSketchCircles As SketchCircles = oSketch.SketchCircles
Dim oSketchCircle As SketchCircle
Dim CircleLength As Double = 0
```

```
For Each oSketchCircle In oSketchCircles
    If oSketchCircle.Construction = False Then
        CircleLength = CircleLength + oSketchCircle.Length*10
    End If
Next

NewLength = ArcLength + LineLength + CircleLength

'--------------------------------------------------

'Ask user for desired length of loop
enteragain:
ReqLength = InputBox("Enter Required Loop Length" & vbLf & vbLf & _
"Current length of loop is " & Round(NewLength,2) & " " & StrDocUnits, "iLogic" _
, "Enter Here")
Try
    If ReqLength = "" Then
        Exit Sub
    Else If IsNumeric(ReqLength) And ReqLength > 0 Then
        Else
        Goto enteragain
    End If
Catch
    Goto enteragain
End Try

'--------------------------------------------------

'Iterate through to achieve desired length
Iterations = 0
ParamChange = 200

Do 'Start iteration

    NewDifference = ReqLength - NewLength

    Try
    If Round(NewDifference,3) = 0 Then
        MessageBox.Show("Sketch Adjusted Successfully!" & vbLf & vbLf & _
        "Total length of loop is now " & Round(NewLength,3) & " " & StrDocUnits & _
        " to 3 decimal places", "iLogic",MessageBoxButtons.Ok,MessageBoxIcon.Information, _
        MessageBoxDefaultButton.Button1)
        Exit Do
    End If
    Catch
        MessageBox.Show("Error - Please check iLogic code to resolve this error", "Title")
    End Try

    'Modify desired parameter
    If Iterations = 0 Then
    For Each oParam In oParams
        If StrParam.Contains(oParam.Name) Then
            oParam.Value = oParam.Value + 0.0001
        End If
    Next
    End If

    'Set gradient
    If Iterations = 1 Then
        Gradient = ((NewLength-OldLength)/0.001)
        'Reverse = "No"
    End If

    If Iterations > 1 Then
        For Each oParam In oParams
            If StrParam.Contains(oParam.Name) Then
```

```
                oParam.Value = oParam.Value + NewDifference/(Gradient*20)
            End If
        Next
    End If

    'Update doc
    RuleParametersOutput()
    InventorVb.DocumentUpdate()

    Iterations = Iterations + 1

    'Cut out after certain number of iterations
    If Iterations > 100 Then
        cc=MessageBox.Show("Unable to adjust sketch successfully" & vbLf & vbLf & _
        "Final length of loop is " & Round(NewLength,3) & " " & StrDocUnits & vbLf & vbLf & _
        "Check input info, and that sketch is constrained correctly" & vbLf & vbLf & _
       "Check that sketch updates as expected when dimension is manually changed" & vbLf & vbLf & _
        "Do you want to reset Dimension?","iLogic",MessageBoxButtons.YesNo, _
        MessageBoxIcon.Question,MessageBoxDefaultButton.Button1)
        If cc = vbYes Then
            oParams.Item(ParamNumber).Value = StrParamOld
            RuleParametersOutput()
            InventorVb.DocumentUpdate()
        End If
        Exit Do
    End If

    'Calculate new loop length
    ArcLength = 0
    For Each oSketchArc In oSketchArcs
        If oSketchArc.Construction = False Then
            ArcLength = ArcLength + oSketchArc.Length*10
        End If
    Next

    LineLength = 0
    For Each oSketchLine In oSketchLines
        If oSketchLine.Construction = False Then
            LineLength = LineLength + oSketchLine.Length*10
        End If
    Next

    CircleLength = 0
    For Each oSketchCircle In oSketchCircles
        If oSketchCircle.Construction = False Then
            CircleLength = CircleLength + oSketchCircle.Length*10
        End If
    Next

    OldLength = NewLength
    NewLength = ArcLength + LineLength + CircleLength

Loop

'Finally delete parameter holder as no longer required
Try
    oParams.Item("CadlineParameterNames").Delete
Catch
End Try


' ************************** END OF iLOGIC CODE ****************************************
```