CWRU DSCI351-451: Semester Project Report 4

Andrew Loach

December 20, 2016

Analysis of Performance Heterogeneity in Time Series Measurements of PV I-V Characteristics: Report 4

Background

In the pursuit of sustainable energy, predicting the degradation of Photovoltaic (PV) modules from time series datasets is a topic of interest. There are two primary sources of time series data used for studying PV degradation. The first source is data collected in real world conditions, often while in use at power plants, while the second source is data collected in laboratory conditions. Creating the data in the lab has the benefits control over the stressors and more responses can be measured to give greater insight into the degradation processes. Real world data is typically more limited to measuring the Maximum Power Point (MPPT), weather conditions, and occasionally IV curves. Some benefits of the real world data include a larger number of samples/replicates and data collected much more frequently. This report will study the I-V curves, plots showing Current with respect to Voltage, for each module. I-V curves are often modeled using a simple five parameter diode model. This model may be appropriate for individual cells, but is not as well suited for modules. Modules typically include a total of 60 cells, which includes three sets of 20 cells in series and also bypass diodes. The I-V curves for modules do not always fit the five parameter model and often display a number of different shapes. In order to study the I-V curves using only the five parameter model, the curves that do not fit this model well are often excluded. As degradation occurs, the I-V curve often reflects the changes. When cells become less homogeneous due to differences in performance caused by degradation, bypass diodes will often turn on. This can be seen in the I-V curves when they exhibit a behavior described as “step I-V curves”" that no longer fits the five parameter model. The size and location of these steps reflect the heterogeneity of the cells and can be analyzed for insights into the degradation.

The Data Sets

All of the data from this project comes through the Solar Durabality and Lifetime Extension (SDLE) Center and their partners. Specifically the data exists for the MLEET project. The largest source of the times series measurements that include I-V curves is the data from Fraunhoffer. These measurements include maximum power points (MPPT) taken each minute. Every five minutes, an I-V curve is taken instead. All this data is stored in a comma delimited file (CSV) with the I-V curves being space delimited. Additionally, this data is efficiently stored in hadoop in an Hbase table. In addition the Fraunhoffer data, the SDLE Center also collects its own from the SDLE sunfarm.

DataBook

This is the DataBook for the Fraunhoffer Data, which is the data set which we will be analyzing. All the other data used in this project so far has only been to demonstrate or test the segmented regression functions. Note that these are only some of the columns. Other columns that have values such as shunt resistance, we are not sure how they have been calculated and prefer to use our own methods on the I-V curve to replace these values.

Column Index	Title	Units	Description
1	id	NA	An identification number for each I-V curve. This has not been used, with a preference for the timestamp.
2	V1	Volts	Space delimited data representing the Voltage of an I-V curve.
3	I1	Amps	Space delimited data representing the Current of an I-V curve.
4	P1	Watts	Space delimited data representing the Power; may be used for P-V curves.
5	tstamp	yyyy-mm-dd hh:mm:ss	The timestamp of the I-V curve in a format designated in the units column.
6	airmass	NA (ratio)	The optical path length of light through the atmosphere
7	aoi	Degrees	Angle of incidence of the sunlight to the PV module
12	irrad_eff	W/m^2	Effective irradiance on the module.
16	pmpp	Watts	The maximum power point
20	sample	NA	The name of the sample
21	sample_id	NA	A number corresponding to each sample.
22	temp_amb	Celsius	Ambient Temperature
23	temp_mod	Celsius	Module Temperature
29	zangle	Degrees	Zenith Angle

Exploratory Data Analysis and Data Visualization

Initial EDA

Exploratory Data Analysis starts with getting to know the data. There are a few million I-V curves and the CSV files are on the scale of many Gigabytes, so the data should be saved in a directory other than a Git repository. After changing our directory we can use R to easily read a CSV file directly into a data frame. For the purposes of this presentation, we will not be processing all of the data, only a small fraction as a subset as a demonstration. Processing all of the data required the use of the HPC Slurm cluster to complete batch jobs and is not suitable for an RMD. We will, however, briefly look at at a full data frame for one of the sites for some quick EDA. In addition, we will load the “segmented” package, which will be the focus of our data analysis.

```
#Change the Directory to where the data is located
setwd("C:/Users/Andrew/Desktop/IV-SegmentedFunction")
library("segmented")
#Read the files into CSVs
GCdf <- read.csv("modified_full_iv_curves3GC.csv", nrows = 5000)
GCdfsmall <- read.csv("SiteGC_weird_curve.csv")
```

As mentioned previously, the I-V curves are formatted as space delimited data inside the CSV. It was necessary to reformat this data into a useful R object, as a list of numeric.

```
#Current, Voltage, and Power Data Directly from the Data Frame after read.csv()
I <- GCdf$I1
V <- GCdf$V1
P <- GCdf$P1
#Select a row for this example
Iex <- I[140]
#Convert to a string, and then split it into a list of strings.
IexString <- as.character(Iex)
IList <- strsplit(IexString, ' ')
#Convert the list of strings into a list of numbers
Inum <- as.numeric(unlist(IList))

#Repeat above process for voltage
Vex <- V[140]
VexString <- as.character(Vex)
VList <- strsplit(VexString, ' ')
Vnum <- as.numeric(unlist(VList))
```

In order to get a good handle on our data frame, some useful R functions include head() and summary(). head() will show us the first few rows of our data frame so we can see how it is formatted in terms of what columns or data it contains and also what some of the values may be.

```
head(GCdf)
```

```
##   X      id  V1  I1  P1              tstamp airmass      aoi ff ff_raw
## 1 0 2616258 0.0 0.0 0.0 2012-08-24 00:00:00      NA 160.0347 NA     NA
## 2 1 2616259 0.0 0.0 0.0 2012-08-24 00:05:00      NA 160.7363 NA     NA
## 3 2 2616260 0.0 0.0 0.0 2012-08-24 00:10:00      NA 161.3810  0     NA
## 4 3 2616261 0.0 0.0 0.0 2012-08-24 00:15:00      NA 161.9630  0     NA
## 5 4 2616262 0.0 0.0 0.0 2012-08-24 00:20:00      NA 162.4759  0  0.934
## 6 5 2616263 0.0 0.0 0.0 2012-08-24 00:25:00      NA 162.9134 NA     NA
##   impp impp_raw irrad_delta  irrad_eff irrad_glob   isc isc_raw pmpp
## 1   NA    0.016         -40 -0.0323415       -0.5    NA   0.016    0
## 2   NA    0.016         100  0.0000000       -0.1    NA   0.016    0
## 3   NA    0.016         150  0.0000000       -0.5 0.016   0.016    0
## 4   NA    0.016          NA -0.0323415       -0.4 0.016   0.016    0
## 5   NA    0.016          NA  0.0000000       -0.4 0.016   0.017    0
## 6   NA    0.016          NA -0.0323415       -0.4    NA   0.016    0
##   pmpp_raw rp rs    sample sample_id temp_amb temp_mod temp_rc vmpp
## 1        0 NA NA 10521-M11        65       NA     22.7 23.0236    0
## 2        0 NA NA 10521-M11        65       NA     22.7 22.9871    0
## 3        0 NA NA 10521-M11        65       NA     22.8 23.0370    0
## 4        0 NA NA 10521-M11        65       NA     22.8 23.0302    0
## 5        0 NA NA 10521-M11        65       NA     22.8 23.0395    0
## 6        0 NA NA 10521-M11        65       NA     22.9 23.0402    0
##   vmpp_raw   voc voc_raw   zangle  tzone longitude latitude elevation
## 1    0.049 0.021   0.059 138.5295 +00:00    -15.42    27.82        10
## 2    0.045 0.022   0.055 138.9633 +00:00    -15.42    27.82        10
## 3    0.052 0.053   0.057 139.3657 +00:00    -15.42    27.82        10
## 4    0.052 0.054   0.058 139.7357 +00:00    -15.42    27.82        10
## 5    0.054 0.046   0.063 140.0725 +00:00    -15.42    27.82        10
## 6    0.052 0.016   0.059 140.3751 +00:00    -15.42    27.82        10
```

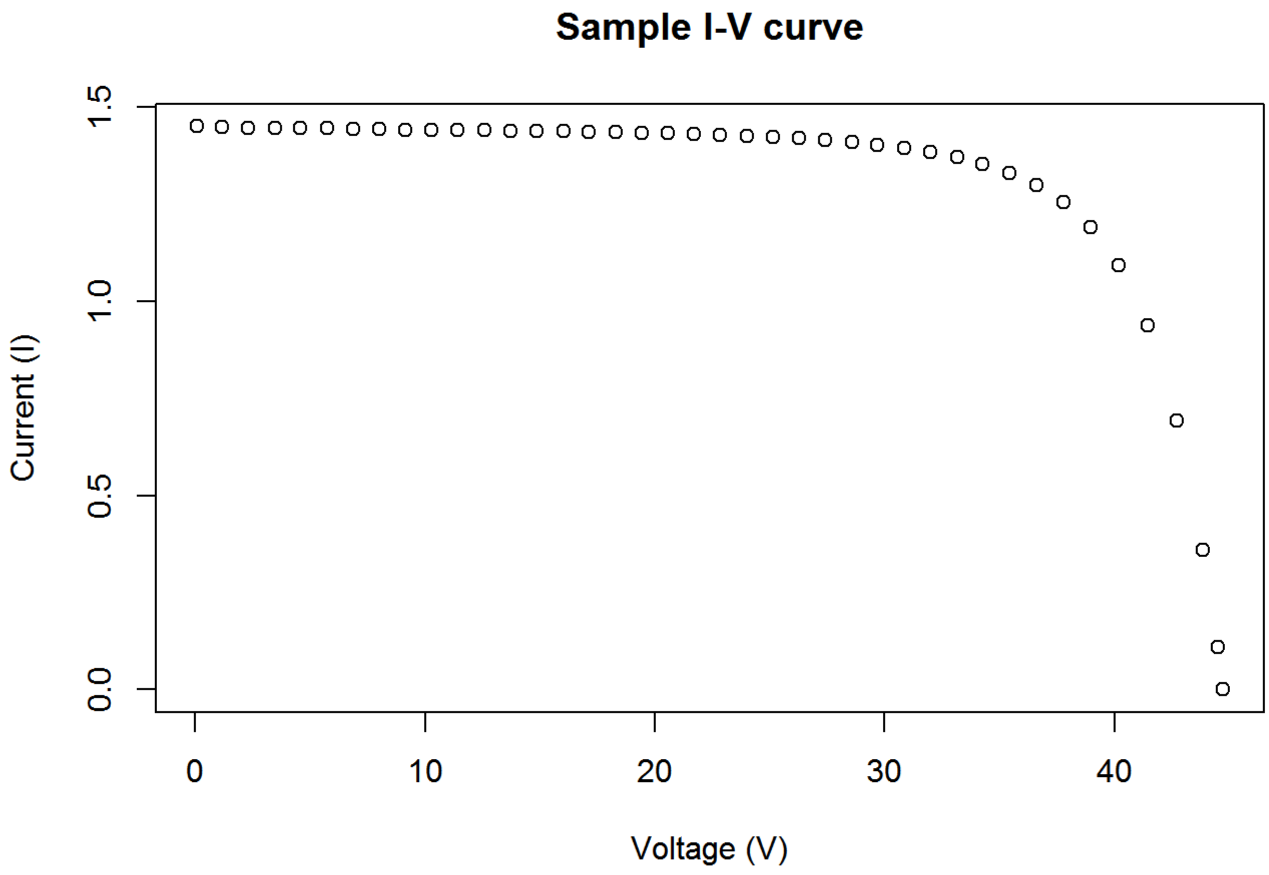
summary() can be used to get some quick statistics about the data frame. However, for the sake of demonstration we will only use this on one column because using summary on the entire data frame would create are rather large amount of output.

```
summary(GCdf$pmpp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000  38.180   9.824 296.200
```

The data that I will be using exhaustively is the I-V curves, which we have extracted above. A single I-V curve typically looks as follows. This is a Type I I-V curve and does not have any steps that are created by bypass diodes turning on. Base graphics is sufficient for EDA.

```
plot(Vnum, Inum, xlab = "Voltage (V)", ylab = "Current (I)", main = "Sample I-V curve")
```



Demonstration of Methods

Our methods revolve around the process of segmented regression, which is the process of fitting a number of straight lines to a curve as a piecewise function. The locations at which these lines intersect are known as the change points. In the context of I-V curves the changepoints can be used to find the number of steps, or bypass diodes turning on.

For this section we will show a demonstration of the methods used to process all of the I-V curves. I will show step by step of the process on a single I-V curve. In this case it will be a Type II I-V curve so that we can see how steps are handled. As we did with the previous data, we must extract the I-V from the space delimited data.

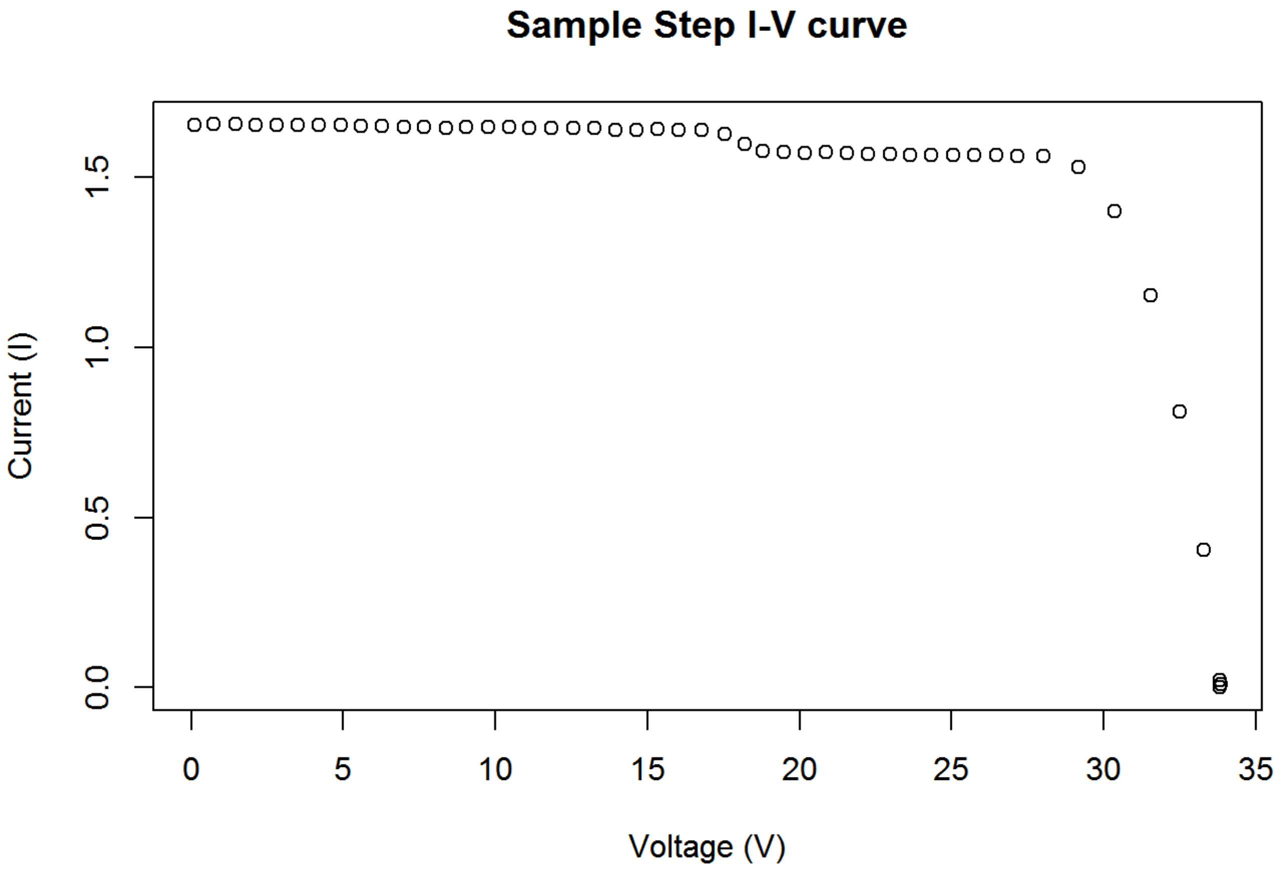
```
I <- GCdfsmall$current
V <- GCdfsmall$voltage

n <- 3

Iex <- I[n]
IexString <- as.character(Iex)
IList <- strsplit(IexString, ' ')
Inum <- as.numeric(unlist(IList))


Vex <- V[n]
VexString <- as.character(Vex)
VList <- strsplit(VexString, ' ')
Vnum <- as.numeric(unlist(VList))


plot(Vnum, Inum, xlab = "Voltage (V)", ylab = "Current (I)", main = "Sample Step I-V curve")
```

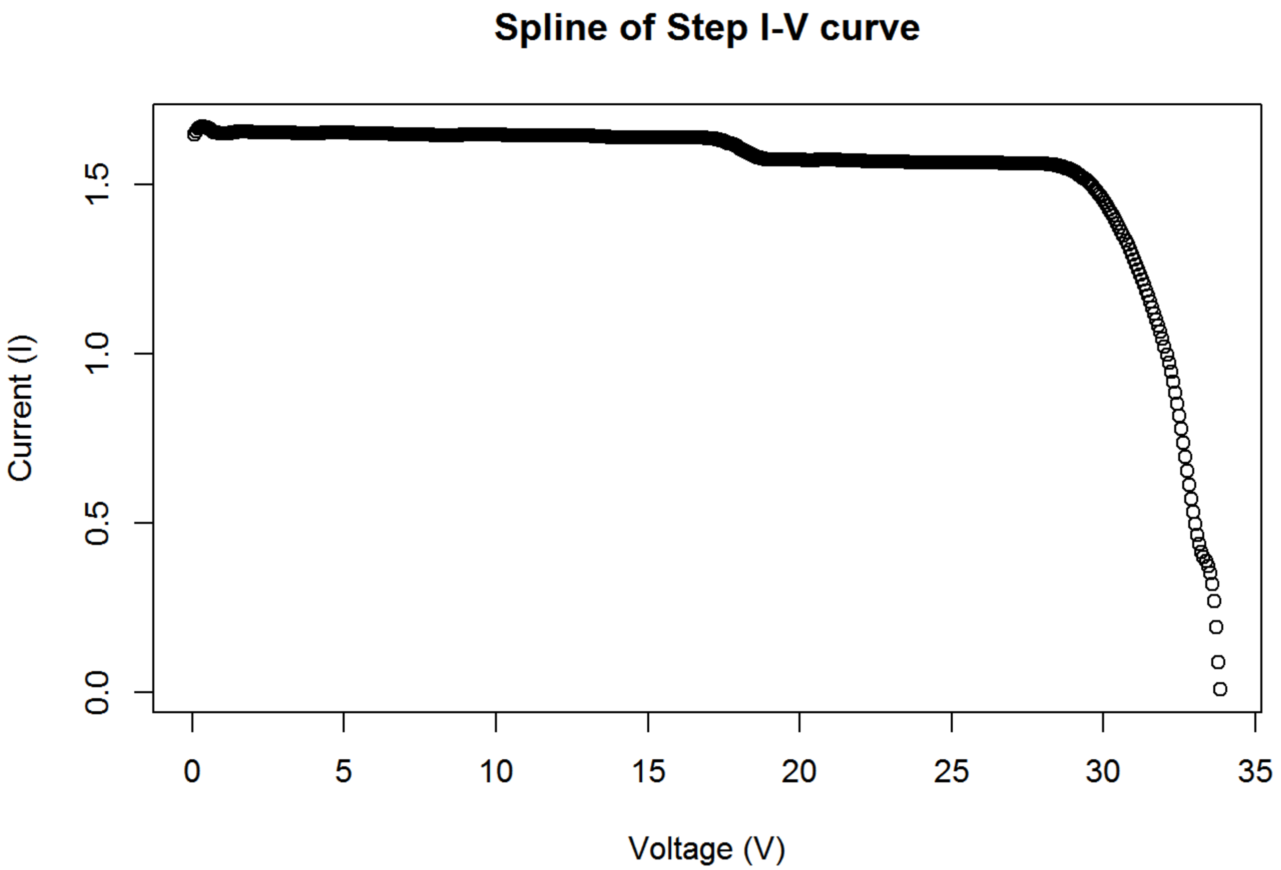


As you can see in the above plot of the data, the I-V curve is only populated by about 50 points. It is believed that using a cubic spline to inerpolate and create a larger number of points will help the segmented regression behave more consistently, so we will do so. I am confident that the spline represents the data reasonably accurately because I have not observed a significant amount of noise or error in any of the data.

The implementation of a cubic spline that I believe to be most efficient is located in base R. We first create values for the voltage over the same range. The number of these values we create is dependent on how many points we desire. It should be noted that not using a spline requires different implementation of the later steps; mostly because the I-V data is not ordered, but the interpolated data is.

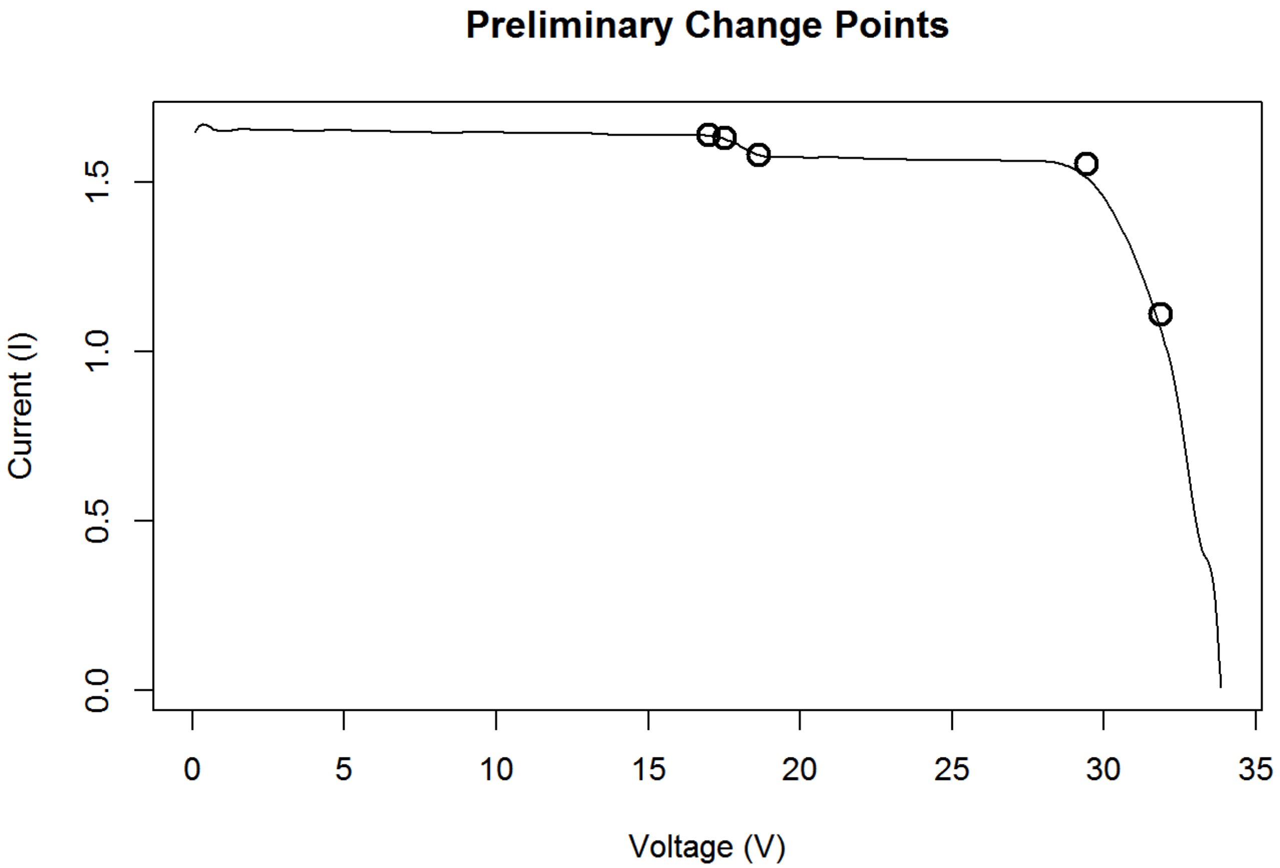
```
x <- Vnum
y <- Inum

#Create x data on the same range
xspl <- ((1:500) / 500) * max(Vnum)
#Create the new data using a spline
newDat <- predict(smooth.spline(x, y), xspl)
x <- unlist(newDat[1])
y <- unlist(newDat[2])
#Plot the data created by the spline
plot(x,y, xlab = "Voltage (V)", ylab = "Current (I)", main = "Spline of Step I-V curve")
```



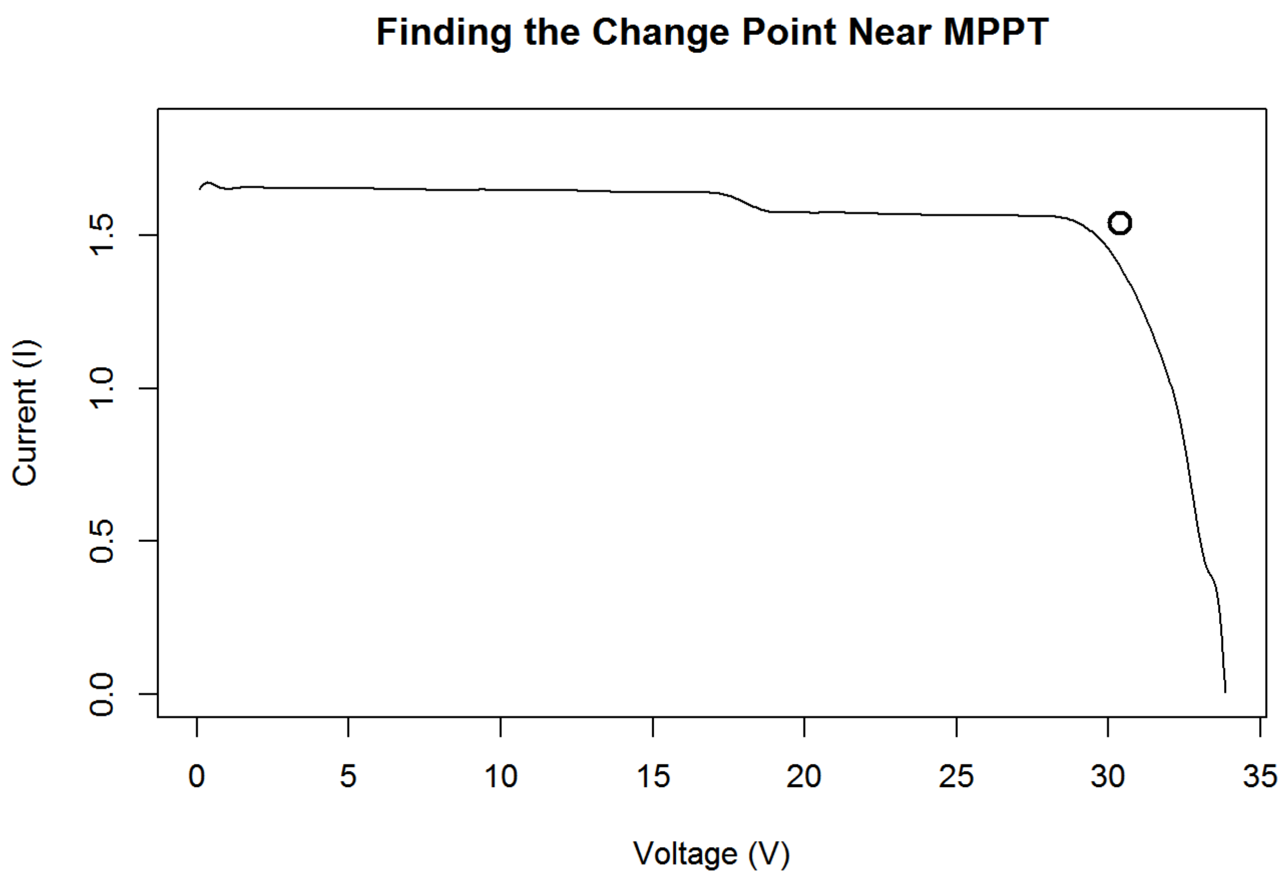
On every I-V curve there is an obvious changepoint located at near the higher voltages. This typically corresponds to the MPPT. If we run the segmented function on the I-V curve as it is, there is a high probability that it will find change points after the MPPT, and we are not interested in points of that area as they do not signify bypass diodes turning on. To demonstrate this, I will show the change points in which the segmented function will find on this I-V curve as it is.

```
f1 <- segmented.lm(lm(y~x),seg.Z = ~x, psi = NA, control=seg.control(K = 7, stop.if.error = FALSE, n.boot=0, it.max=20))
plot(x, y,type="l", xlab = "Voltage (V)", ylab = "Current (I)", main = "Preliminary Change Points")
points.segmented(f1)
```



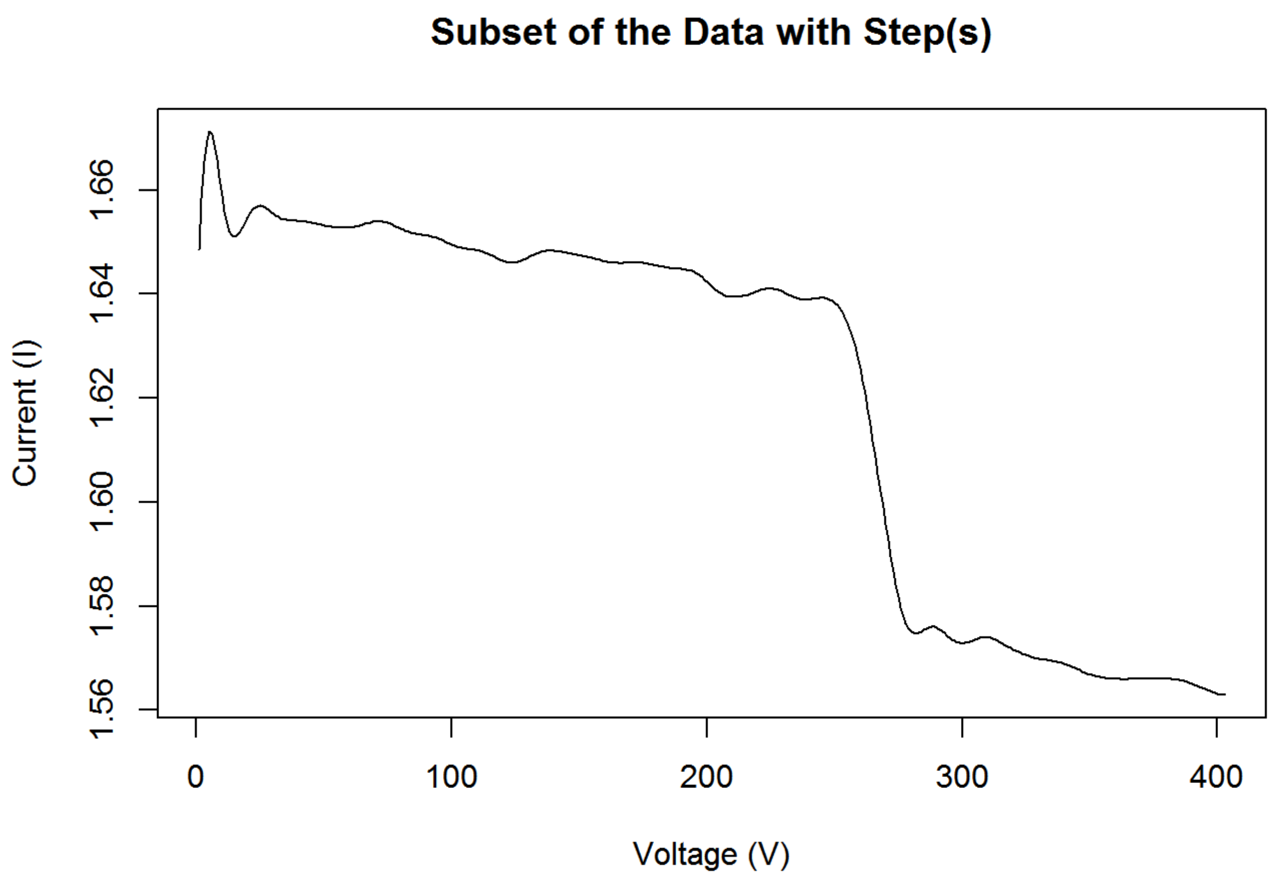
To eliminate this problem, we will run the segmented function and have it find only a single change point, which will be very close to the MPPT. Then were will select the first 90% of the data that is to the left of that changepoint and run a spline on it. The reason we are only using the data to the left of the MPPT is that this region is more linear and therefore it is more reasonable to run segmented regression upon. The region to the right may have some changepoints, but they will not represent bypass diodes turning on, and we are not yet interested in them.

```
f1 <- segmented.lm(lm(y~x),seg.Z = ~x, psi = NA, control=seg.control(K = 1, stop.if.error = FALSE, n.boot=0, it.max=20))
plot(x, y, ylim = c(0, 1.1 * max(y)), type="l", xlab = "Voltage (V)", ylab = "Current (I)", main = "Finding the Change Point Near MPPT")
points.segmented(f1)
```



```
breakPoint <- f1$psi[2]

x <- which(x < breakPoint)
x <- x[1:floor(.9 * length(x))]
y <- y[1:length(x)]
plot(x,y, type="l", xlab = "Voltage (V)", ylab = "Current (I)", main = "Subset of the Data with
 Step(s)")
```

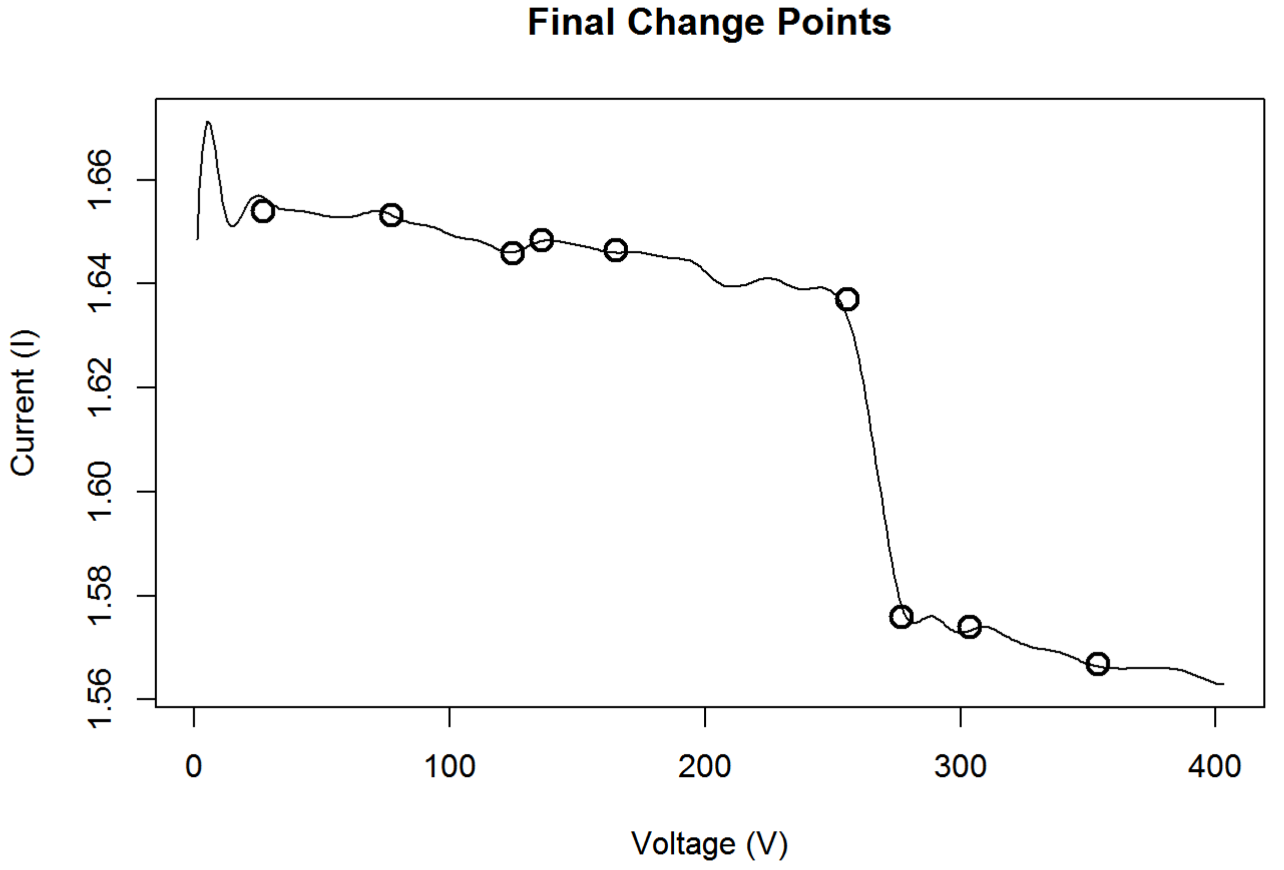


Now that we have the region in which we desire, we can use segmented regression to find our step. Notice that it will return a number of changepoints. Some of this is due to the spline amplifying some of the small noise or artifacts. Be sure to notice that teh scale on the y-axis is very small. However, when we check the summary of our segmented regression, we can see that the two points that we are looking for have much higher T values than the others. By having a threshold on the T value, we can find an record the number of change points and their location for each I-V curve. Ideally we would be able to set the tolerance of the P value to remove the less significant change points, but as you can see in the summary, it does not appear to be useful after a certain point and we must extract the T values from the segmented object.

```
f1 <- segmented.lm(lm(y~x),seg.Z = ~x, psi = list(x=NA), control=seg.control(stop.if.error = FAL
SE, n.boot=0, it.max=20))
```

```
## Warning: max number of iterations attained
```

```
plot(x,y, type="l", xlab = "Voltage (V)", ylab = "Current (I)", main = "Final Change Points")
points.segmented(f1)
```


Final Change Points
Current (I)
1.56
1.58
1.60
1.62
1.64
1.66
0
100
200
300
400
Voltage (V)

```
summary(f1)
```

```
##
##  ***Regression Model with Segmented Relationship(s)***
##
## Call:
## segmented.lm(obj = lm(y ~ x), seg.Z = ~x, psi = list(x = NA),
##     control = seg.control(stop.if.error = FALSE, n.boot = 0,
##         it.max = 20))
##
## Estimated Break-Point(s):
##            Est. St.Err
## psi1.x  26.916  2.672
## psi2.x  77.050  5.083
## psi3.x 124.746  2.932
## psi4.x 135.961  4.189
## psi5.x 165.113 22.160
## psi6.x 255.722  0.290
## psi7.x 277.194  0.353
## psi8.x 303.578 12.600
## psi9.x 353.822  7.846
##
## Meaningful coefficients of the linear terms:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  1.663e+00  6.825e-04 2436.375  < 2e-16 ***
## x           -3.257e-04  4.419e-05   -7.370 1.06e-12 ***
## U1.x         3.086e-04  4.703e-05    6.562       NA
## U2.x        -1.346e-04  2.426e-05   -5.549       NA
## U3.x         3.772e-04  1.622e-04    2.326       NA
## U4.x        -2.969e-04  1.650e-04   -1.799       NA
## U5.x        -3.175e-05  3.630e-05   -0.875       NA
## U6.x        -2.742e-03  5.721e-05  -47.935       NA
## U7.x         2.768e-03  7.196e-05   38.462       NA
## U8.x        -6.503e-05  4.719e-05   -1.378       NA
## U9.x         8.618e-05  2.342e-05    3.679       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.00169 on 383 degrees of freedom
## Multiple R-Squared: 0.998,  Adjusted R-squared: 0.9979
##
## Convergence attained in 20 iterations with relative change -0.01666681
```

Write Up Results

Interpret Results

After using the segmented regressing to find the change points on a large number of I-V curves, we find results that are very similar to a previous method used to classify the data. The non-parametric regression method used in I-V stats yielded a similar ratio of the types of (usable) I-V curves with approximately 90% being classified as Type I, 8% as Type II, and 2% as Type III. This data is only the results of I-V curves at a single site, and several more are ready to be analyzed and validated, but will require some more time to be run as batch jobs. This implies that not only does the method seem to work reasonably well for all of the individual cases that were specifically analyzed to test the method, but it also appears to be effective on the large scale as well, or at least similarly effective to the previous method. The proportion of I-V curves that were found to be Type III was also larger, which may imply that the segmented regression method is more effective at discerning the difference between Type II and Type III I-V curves as there is a possibility of misclassification because the first step of a Type III I-V curve is often very shallow.

Challenge Results

As we do not have any sure fire method of determining exactly the number of each type of I-V curve, there is no certainty that the method is anywhere near perfect and there still may be issues with misclassification. Occassions of misclassification may occur if the method is too sensitive and mistakes noise for change points. At the other extreme, the method may not deem changepoints as significant if they are too shallow, and may not notice that a bypass diode is turning on. In addition in may not be prudent to use segmented regression in this manner. Segmented regression is designed to find changes in linearity in a piecewise linear function. However, we are well aware that I-V curves are not linear, although the flat region to the left of the MPPT often is when it does not contain steps. A final concern is that using a cubic spline is a form of imputation and may not perfectly represent I-V curve. A final concern about the method is that the use of an arbitrary threshold of T values to distinguish significant changepoints from insignificant changepoints may or may not be scalable. The spline may need to always be used in the same manner if this threshold is to remain constant. It is also not entirely clear what the T-value is representing in this context, but it does seem to highly correlate with the signficance of the change points.

Conclusions

The segemented regression method of classifying I-V curves appears to work reasonably well. This classification will be useful in analyzing I-V curves in a time series manner for future projects. In addition, the method can be and has been slightly modified to find new parameters of interest, such as the voltages in which these changepoints tend to occur. The challenges to my method are only of slight concern. Imputation does not appear to be all that bad, as we are using it to interpolate data that appears to have very little error. Also, using segmented regression on the region to the left of the MPPT seems reasonable on a typical I-V curve because this region tends to be very linear when there are not any bypass diodes turning on. When there are bypass diodes turning on, segmented regression appears to find them consistently.

v0.00.19 - 16/12/20 - Edits before submission