9/12 Day One

Tuesday, January 21, 2020 11:13 AM

Notes 2019.09.12

TODO - myHR
- Time & Leave - Apply & Save
- Time sheets are due Monday @10AM

Employee stock purchase plan eligible if I go thru to July

Update W4 thru myHR

Done - Direct Deposit information

Waive insurance? Thru ENROLLNOW

Done - Manage my 401k match? Brady?

TODO - Check out fitness classes

Aliases:

IT help: MYIT
Course/Objectives/Organzational Structure: SF
Download apps: SAM
Changing passwords: PASSWORD
TD questions: TDONBOARDING

Acronym helper: LEXICON
- OES: Optical emission spectroscopy
- EPD: Endpoint detection
- LAM: LAM Research tool, thru dry etch
- THK: thickness
- ADI: After Develop inspect
- ACI: After Clean inspect
- CD: Critical Dimension
- JMP: John's Mactintosh Project, Statistial Anaysis Program
- PCS: process control system
- SPC: statistical process control
- 53 Periph EB (etch back) DE (dry etch)
- Hadoop: Open-Source Software Framework for storing and large scale processing of datasets on clusters of servers.
- CT: Cycle Time
- HC: Project Role, HC, Team Member? Head Count, I guess?
- FD: Fault Detection
- WIS: Wafer Intelligent Scan/Wafer Inspection Section? Tool scanning wafer
- FE: Front-End [manufacturing]
- TFT: Task Force Team
- BKM: Best known method
- FF Control: Feed Forward Control
- UVA: Univariate analysis - The statistical analysis is made out of a single variable. in contrast to MVA or

multivariate analysis where several variables are considered simultaenously for analysis
- PCA: Principal component analysis (PCA) is a mathematical procedure that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (i.e., uncorrelated with) the preceding components. Principal components are guaranteed to be independent if the data set is jointly normally distributed. PCA is sensitive to the relative scaling of the original variables. (Wikipedia)
- PLS: Regression software
- GDML: Global D? Middle Layer, Data?

VM Model building:

1. Pick out OESIB trace, EPD or stronger signal
2. Apply smoothing, and 1st & 2nd derivative (critical point time stamps -> inflection points?)
    - Identify inflection points (=material transition)
3. Plot Del_NitrideTHK (~160-260 [units?]) v Etch Time (~6-18 [units?])

Questions

-HELP- How much time does metrology add cycle-time to wafer processing? Is this per wafer? What is the goal time reduction virtual metrology will achieve?

9/17 Acronyms

Tuesday, January 21, 2020 11:12 AM

Notes 2019.09.17

Howard's Schedule: 6:00AM~2:30PM
Done - Request JMP
Done - Download Prism
Done - Look up Deming Philosophy
TODO - f4photoapps, will I be using this? Would it be helpful if I knew how to use this?
Done - Request E3

Acronym helper: LEXICON
- OES: Optical emission spectroscopy
- EPD: Endpoint detection
- LAM: LAM Research tool, thru dry etch
- THK: thickness
- ADI: After Develop inspect
- ACI: After Clean inspect
- CD: Critical Dimension
- JMP: John's Mactintosh Project, Statistical Analysis Program
- PCS: process control system
- SPC: statistical process control
- 53 Periph EB (etch back) DE (dry etch)
- Hadoop: Open-Source Software Framework for storing and large scale processing of datasets on clusters of servers.
- CT: Cycle Time
- HC: Project Role, HC, Team Member? Head Count, I guess?
- FD: Fault Detection
- WIS: Wafer Intelligent Scan/Wafer Inspection Section? Tool scanning wafer
- FE: Front-End [manufacturing]
- Green-to-green: From the green light to start a lot, send it through its process and do the follow-up qual on a tool to confirm the integrity of the preceding results, to the green light signaling ready for a new lot
- TFT: Task Force Team
- BKM: Best known method
- BD:?
- FF Control: Feed Forward Control
- UVA: Univariate analysis - The statistical analysis is made out of a single variable. in contrast to MVA or multivariate analysis where several variables are considered simultaneously for analysis
- PCA: Principal component analysis (PCA) is a mathematical procedure that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (i.e., uncorrelated with) the preceding components. Principal components are guaranteed to be independent if the data set is jointly normally distributed. PCA is sensitive to the relative scaling of the original variables. (Wikipedia)
- PLS: Regression software
- GDML: Global Data Middle Layer: provides virtualized access to data that exists throughout Micron
- RGB: Red Green Blue - Producing white light through combo of RGB

- MicA:? Micron Abatement? Thru the Y3 software using Hadoop?
- RDA: Real-Time Defect Analysis
- R2R: Run-to-Run Control: control strategy applied to successive runs of production
- E3: Enterprise Equipment Engineering? Software?
- Y3: Yield Cube Program: Statistical analysis?

Questions:

-Done- What's the difference between traveler & recipe? Traveler is the whole docket, whereas recipe is area-specific for an area-specific tool

-HELP- What is the significance of Hz is sampling? Frequency of writing out the data for a specific parameter

-HELP- Notch? Is there a notch on the wafers?

9/24 Onboarding

Tuesday, January 21, 2020 11:09 AM

Notes 2019.09.24

TODO - Follow-up on Traveler's classes

Done - Follow-up with IT if BOSECURE isn't working

Done - Complete SF courses, maybe follow up with Melanie regarding the orientation course?

TODO - Review Chris Turner's VM presentation prior to R:9/26 meeting @ 3:00PM

Questions:
-HELP- Current metrology limited to pre-defined (unity map) measurement sites... Tell me more about this unity map? Why is it limiting?

9/17 I Heart Acronyms

Tuesday, January 21, 2020 11:06 AM

Notes 2019.09.17

Acronym helper: LEXICON or TDACRONYM

- OES: Optical emission spectroscopy
- EPD: Endpoint detection
- LAM: LAM Research tool, thru dry etch
- THK: thickness
- ADI: After Develop inspect
- ACI: After Clean inspect
- CD: Critical Dimension
- JMP: John's Mactintosh Project, Statistial Anaysis Program
- PCS: process control system
- SPC: statistical process control
- 53 Periph EB (etch back) DE (dry etch)
- Hadoop: Open-Source Software Framework for storing and large scale processing of datasets on clusters of servers.
- CT: Cycle Time
- HC:? Project Role, HC, Team Member? -HELP- Head Count, I guess?
- FD: Fault Detection
- MCT: Manufacturing Central Team
- WIS: Wafer Intelligent Scan/Wafer Inspection Section? Tool scanning wafer
- FE: Front-End [manufacturing]
- Green-to-green: From the green light to start a lot, send it through its process and do the follow-up qual on a tool to confirm the integrity of the preceeding results, to the green light signaling ready for a new lot
- TFT: Task Force Team
- BKM: Best known method
- BD:? -HELP-
- FF Control: Feed Forward Control
- UVA: Univariate analysis - The statistical analysis is made out of a single variable. in contrast to MVA or multivariate analysis where several variables are considered simultaenously for analysis
- PCA: Principal component analysis (PCA) is a mathematical procedure that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (i.e., uncorrelated with) the preceding components. Principal components are guaranteed to be independent if the data set is jointly normally distributed. PCA is sensitive to the relative scaling of the original variables. (Wikipedia)
- PLS: Regression software
- GDML: Global Data Middle Layer: provides virtualized access to data that exists throughout Micron
- RGB: Red Green Blue - Producing white light through combo of RGB
- MicA:? -HELP- Micron Abatement? Thru the Y3 software using Hadoop?
- RDA: Real-Time Defect Analysis
- R2R: Run-to-Run Control: control strategy applied to successive runs of production
- E3: Enterprise Equipment Engineering, Software
- Y3: Yield Cube Program: Statistical analysis?

- REML: Restricted Maximum Likelihood: not an estimate of max likelihood for all info, but instead uses likelihood function from data sans nuisance parameters

-HELP- 7yjKh@&3gA

10/8 Pip Pip Hooray

Tuesday, January 21, 2020 11:05 AM

Notes 2019.10.08

Solved `pip install` problem!
- https://github.com/pypa/pip/issues/6261
- The Windows equivalent of rm -rf [path/to/del] is rmdir /S [path/to/del]

Installing packages and starting to work thru the OES excel files
- numpy, done, just confirming pip is up to date
- pandas
- matplotlib
- datetime
- time

10/15 DataFramez for Dayz

Tuesday, January 21, 2020 11:03 AM

Notes 2019.10.15

Today:
Completed python script for taking in data by specific columns
TODO - Clean up the retrieval of data by collecting OES column info first, then using those specific columns to be called in overall data retrieval.
Next up is identifying the critical steps to perform derivations on
- I'll do this by comparing the metro info for a process with the dry etch OES data. Steps included below.
TODO - Rearrange the data allocation for each OES file to be within retrieval loop
TODO - Comment current scripts for better code parsing and organization

OES: 5030-22 STI INTEGRATED DRY ETCH
Metro: 1230-22 STI INTEGRATED DRY STRIP THK = 22STIINT.txt?? -HELP-: Only includes 1220-22
Parameter:
5030-22 STI NITRIDE DRY ETCH/THK_OXIDE2_NCHK

Metro: 1230-53 SPACER NITRIDE DRY STRIP THK = 53Spacer.txt, includes correct stepname
OES: 5030-53 SPACER NITRIDE DRY ETCH
Parameter:
5030-53 SPACER NITRIDE DRY ETCH/THK_NITRIDE_PASS

OES: 3500-GH CONTACT OXIDE DRY ETCH
Metro: 1230-GH CONTACT PRE METAL CMP THK - GHContactOx1 & 2.txt
Parameter:
5200-GH CONTACT METAL CMP/THK_OXIDE_PASS

Since the STI Integrated Dry Etch seem to be mismatched, I'll work on the GH Contact process first

Think the next step would be to write a mini script seeing if LotIDs matched between metro .txts and oes .csvs

10/17 UniqueOES

Tuesday, January 21, 2020 11:02 AM

Notes 2019.10.15

Objective: Create code snippet that takes the LotID column from metro txt files and match them up to the LotID column in the oes csv files.

-Completed!-

Barriers: Understanding of Data Frames and how to use them with if loops

From Chris Turner:

On the OES analysis: Remember, you will want to plot 1st and 2nd derivative, find the extrema and then the time elapsed at the extrema.

OES: 5030-22 STI INTEGRATED DRY ETCH
Metro: 1230-22 STI INTEGRATED DRY STRIP THK = 22STIINT.txt?? -HELP-: Only includes 1220-22
Parameter:
5030-22 STI NITRIDE DRY ETCH/THK_OXIDE2_NCHK

Metro: 1230-53 SPACER NITRIDE DRY STRIP THK = 53Spacer.txt, includes correct stepname
OES: 5030-53 SPACER NITRIDE DRY ETCH
Parameter:
5030-53 SPACER NITRIDE DRY ETCH/THK_NITRIDE_PASS

OES: 3500-GH CONTACT OXIDE DRY ETCH
Metro: 1230-GH CONTACT PRE METAL CMP THK - GHContactOx1 & 2.txt
Parameter:
5200-GH CONTACT METAL CMP/THK_OXIDE_PASS

Notes:

Okay, so I think I have some code that's working:
I created a UniqueNames array of the oes LotIDs (relatively small)
Next, compare those to available metro list of lot IDs
So far everything is turning up 'False' which is slightly disheartening
But perhaps my code is flawed, will test with by altering oes to contain a known lotID in metro
And... shoot, problem's on my end. I'm going to dig into what exactly these are comparing.
Woooo, figured it out!
Use numpy.isin to compare the UniqueOES lotIDs to the a list of the metro lotIDs
Next step was concatenating the metro multiples into one --Done!
Now to correlate the True values in 'new' to the oes files and discard the rest, also identify the step it's being performed at

Notes on corresponding Metro and OES wafers:
Metro: GHContactOx2.txt

lotid	waferscribe	stepname	parameter_name
sample_date	ext_mv		
3932323.003	WCELL085ESC4	5200-GH CONTACT METAL CMP	TIME_STEP6_WVALP
2019-07-08 16:41:51	29.058333333333301595757802715525031089782714843 75		

OES: 3500-GH CONTACT OXIDE DRY ETCH_10.csv >>> WCELL085ESC4.xlsx
- I'm noticing that there's multiple wafers w/in a lot

- One would see that TIME_STEP6 is referenced in the Metro parameter name, though the Step_IDs in OES are a little trickier to pick out.
- TimeStamp and sample_date don't align either, TimeStamp = 7/5/19 AM, while sample_date = 7/8/19 PM.
- No glaringly obvious association with ext_mv so digging into that may get tedious

Notes on correlating the True values in 'new' to the oes files:
Contain NO corresponding metro data: (maybe should add in code to discard these completely if new contains ONLY False) TODO
- 3500-GH CONTACT OXIDE DRY ETCH_1.csv
- 3500-GH CONTACT OXIDE DRY ETCH_2.csv

Objective: read_csv only certain rows that contain specified value within a specific column
Ideas: Possibly use na_filter: "Detect missing value markers (empty strings and the value of na_values). In data without any NAs, passing na_filter=False can improve the performance of reading a large file."
End point: I am able to create a dict of lots for 'filename', but they get written over each iteration in oes\

I implemented concatenate to accumulate 'data' throughout the loop, but it seems like the length of lots and UniqueOES are mismatched? Maybe they should be?
Investigate further Tuesday^^^

10/22 Dask is Wack

Tuesday, January 21, 2020 10:55 AM

Notes 2019.10.22

Follow-up with Chris: Confirm STI INTEGRATED has matching LOT IDs between Metro and OES
-Completed!-

Because STI INTEGRATED contains corresponding metro LOTs:
Ran LotIDCompare script for STI INTEGRATED and ran into issue concatenating 'data'
Can create the 'lots' dict only within loops, so accessing the whole data is gonna be tricky

Continuing from Thursday: For GH CONTACT, it seems that when you concatenate the data as a whole,
there's repeated lots and they get combo'd in UniqueOES. Going to proceed, but may be cool to go back and verify this somehow

Looking into the concatenation disparity between GH CONTACT and STI INTEGRATED
- Perhaps using dask vs. pandas? Check it out!
  - Nah, dask is wack.
- Maybe using dict.update? Check it out!
  - Didn't really do the due dilligence, but I found something interesting!...
- STI INTEGRATED has 1 & 2 duplicated, I believe
  - Can't check as sharepoint isn't working? Follow-up on this on Thursday (TODO)

Tuesday Left-Off:
STII is going through the glob loop, but the enumerate to create lots[i] is breaking, Porque?

10/24 MemoryError

Tuesday, January 21, 2020 10:55 AM

Notes 2019.10.24

Tuesday Left-Off:

STII is going through the glob loop, but the enumerate to create lots[i] is breaking, Porque?

Robert Patraw's back! Wants a powerpoint presentation of my progress in VM thus far

See OES_VM_Presentation.pptx for more deets

Additionally pressed me for timeframe of completing automation script, by end of internship?

WOOF

Objective: Make LotIDCompare.py work for STI & GH & EVERYTHING

Barrier: Loop creating lots[i]

Ideas: Let's keep {data}, but use DataFrame.drop to get rid of non-metro OES

Dang, really keep running into that MemoryError

Maybe explore memory error? That seems soooo tedious...

Maybe drop these non-metro rows within the glob loop?

10/29 STI & GH Works

Tuesday, January 21, 2020 10:55 AM

Notes 2019.10.29

Thursday Objective: Make LotIDCompare.py work for STI & GH & EVERYTHING

Quick Aside:

- Got a message from a PCVD guy regarding validating VM models for them. He mentioned he was referred to me from his boss, Josh Martin, from discussing this with Patraw. Requested an email follow-up looping Patraw in.
- Robert feels ready for the VM meeting with Gatzemeier and other higherups with my mini-presentation

  Yay!

Revist Previous Objective:

- Going to upload python scripts to private GitHub along with the matching lotID sample I have to help run some script and get input from Mike & Eric
- Will also copy over some frequent errors to a text document for more info

Objective: Check that GH CONTACT and STI INTEGRATED both work with LotIDCompare.py script

Notes: They work!

Final length of 'data' after dropping non-matching metro

GH CONTACT: 240774

STI INTEGRATED: 1759337

Woah, that's a long set!

Onto new objective!

Objective: Incorporate DataRetrieve to LotIDCompare

To revist on Thursday!

10/31 Combo Script

Tuesday, January 21, 2020 10:55 AM

Notes 2019.10.31

Objective: Incorporate DataRetrieve to LotIDCompare

Quick Aside:

- Got an email from Robert regarding some more steps to... correlate?
  I wanted to clarify whether I should follow up with Adrian
  It sounds like I should follow up with Adrian regarding identifying the step where I perform derivative and extrema break-down

HUH:

- So I'm running the previously working LotIDCompare script and I'm getting the memory allocation errors again. WHAT GIVES?!
- I went ahead and trimmed down the columns to the carbon ones and got it to work, but still WHAT GIVES?!

Continuing on to combine DataRetrieve and LotIDCompare
Completed 'Time in Seconds'
Completed plotting the OES trace per wafer

To-Do: Write up an email to Adrian/Chris/Robert giving deets on where I'm at, questions I need answers to and a demo of what prints out.
Done!

Ooh, ooh, ooh!
Step 6 IS the phase change step! If the one I've zoned on IS correct, which I think it is.

Also!
'ext_mv' IS the VM parameter I need, or more that it contains in. The values with the trailing decimals correspond to the parameters I'm measuring! Sweet!

Objective: Find 1st Derivative, 2nd Derivative, Extrema
Barrier: Need to smooth that baby out! I'm able to plot it but man is it messy.

11/5 Smoooothing

Tuesday, January 21, 2020 10:55 AM

Notes 2019.11.5

Objective: Find 1st Derivative, 2nd Derivative, Extrema
Barrier: Need to smooth that baby out! I'm able to plot it but man is it messy.
Notes: Looking into scipy's smoothing techniques
Found maybe an interpolation technique that's working but not sooooper smoothing

HUGE aside:

- From Adrian, Chris is looking for less automation on my part and more traditional model building a la Eric
- From Adrian: "I'm working on dumping every etch step, every oes signal and then running a change point to return the extrema time stamp, next step would be once we have 32 spc points to pull from snowflake and auto model"
- Meeting is setup for next week, 11/12 for OES-VM Planning and Next Steps. Should maybe prepare laptop to run during meeting? Yes.

11/7 Derivatives & Extrema

Tuesday, January 21, 2020 10:55 AM

Notes 2019.11.07

Objective: Find 1st Derivative, 2nd Derivative, Extrema
Barrier: Smoothing that bad boy to a reasonable extent
Notes: Currently playing with the degree of spline fit, k, and the smoothing condition, s.
s-value has to be laaarge, while k-value, I'm really not sure
Going to start exploring the extrema and see what I can find
Using 'find_peaks' though scipy to get it done, mkay, I think I have 'em
Eric says:
Sends inflection points' time stamps to data table
Matches up with respective wafer's metro
Plots parameter v. etch time
Subtracted t @ inflection point from trace endpoint

Metro: 22STIINT
OES: 5030-22 STI INTEGRATED DRY ETCH_2
Wafer: 5763-02
LotID: 3955763.003
Parameter Name: DEPTH_TOTAL_TRENCH_BOX01
sample data: ext_mv:
2019-08-10 10:18:46 335.39341666666700803034473210573196411132812 5
2019-08-10 10:18:46 335.47449999999997771737980656325817108154296875

- These time stamps don't match up at all, so I dunno what Eric was able to match up
Step 6: 8/10/2019 9:27:35 AM - 9:27:54 AM

- Maybe it's because the STI's don't match up? The Metro info was supposed to be the 1230, but I have the 1220. Let's look at the GH Contact Oxides and see what I can connect?

Left-off:
Still working on adapting the ProcessData.py script to the GH CONTACT stuff. Also made a script for checking out the whole kaboodle of traces, aptly named. ProcessData.py is still the main script, just using GH_Contact_EXT to explore tweaking the 'tck' interpolation by lowering the smoothing s-value.

Metro: GHContactOx2
OES: 3500-GH CONTACT OXIDE DRY ETCH_1
Wafer: 2323-17
LotID: 3932323.003
Parameter Name: TIME_STEP6_WVALP
sample data: ext_mv:
2019-07-08 16:41:51 29.05833333333330159575780271552503108978271484375

- AGAIN, these time stamps don't match up at all
Step 4: 7/5/2019 9:03:19 AM - 9:04:04 AM

Whomp. Let's see what the meeting directs me to.

11/12 WAFERSCRIBE

Tuesday, January 21, 2020 10:55 AM

Notes 2019.11.12

Left-off:

Still working on adapting the ProcessData.py script to the GH CONTACT stuff. Also made a script for checking out the whole kaboodle of traces, aptly named. ProcessData.py is still the main script, just using GH_Contact_EXT to explore tweaking the 'tck' interpolation by lowering the smoothing s-value.

Objective: Cleaning up script and creating outputs to show in meeting
Notes: Starting with WholeKaboodle.py - Done!
Next GH_Contact_EXT.py - Done!
Finally ProcessData.py - Needs a lot of repairing, I didn't really put it back together after trying to get it to fit GH_Contact AND STI_Integrated.

Objective: Plot Parameter v. Etch Time
Barriers: For a given ext_mv parameter for a Lot, how does that value get assigned to an OES trace for a Wafer within that Lot?
Notes:
- AH! It's WAFERSCRIBE, christ, Neale, get it together.
- So now I'm needing to narrown down to metro-matching lotid, but also waferscribe, THEN match those OES critical times with those Metro parameters and plot 'em.
- Now I'm pondering how to go about singling out the metro with ext_mv with decimals thru x%1 == 0, or something like that. It'd be nicer if I could make the metro data frame only have those and THEN filter down the LotIDs and WaferScribes. I think that'll be my objective for Thursday

11/14 Trim Metro

Tuesday, January 21, 2020 10:55 AM

Notes 2019.11.14

Left-off:

Now I'm pondering how to go about singling out the metro with ext_mv with decimals thru x%1 == 0, or something like that. It'd be nicer if I could make the metro data frame only have those and THEN filter down the LotIDs and WaferScribes. I think that'll be my objective for Thursday

Objective: Trim down metro dataframe to only include the relevant ext_mv

Notes: Think I've got it, I pretty much used the same trimming down script within the Lot and Wafer loops. Which makes me think I'm going thru too many extra steps in (TODO) identifying the unique wafers/lots. Maybe I should check that out someday.

So I think I'm ready to start the linear regression, correlating the ext_mv to either etch time or critical time. Something tells me etch time may be smarter? Also, I think I should start chatting with process owners regarding the steps I should flag. I dunno why I'm dreading that, but maybe I could move forward with a rough linear regression and have the step be changeable.

Next Step!

Objective: Plot Parameter v. Etch Time

Notes: Checking another step, prolly not what I'm looking for. Based off Eric's etch time stuff, step 6 is prolly the one I want and should pursue, at least for the etch time route.

Man, step 12 is messsyyyyy, s-value is past 1e10, trying 1e15 and expecting smooth AF, hahahahahaha, oh baby, so smooth. Let's try 1e12, nah, for step 12 an s-value of 1e11 is best.

Okay, let's go back to step 6, with s-value of 1e9, and start evaluating etch time
Man, I dunno why I'm so hesitant to start the regression but I keep veering between GH-Contact or STI-Integrated.

Okay, here we go.
Wait, I should really write this email to Stan to get some more insights into these processes and which step/trace I should be honing in on. I can create the scaffolding for it, but in the end, I'm only going to get good results from getting good info.

Working on creating an etch_time DataFrame that appends the waferscribe value and the x_int[peak[0]] values, which I can then combine into the wafer_param_time DataFrame, hopefully.

Ooh, possibly worked, just cleaning up code to print out results, really need to pee though so will prolly leave it for Tuesday

11/26 Etch v Param

Tuesday, January 21, 2020 10:54 AM

Notes 2019.11.26 (9:30-15:30)

Left-off:

Working on creating an etch_time DataFrame that appends the waferscribe value and the x_int[peak[0]] values, which I can then combine into the wafer_param_time DataFrame, hopefully.

Ooh, possibly worked, just cleaning up code to print out results

Back and Better(ish) than ever!

Starting with replying to Stan

Done!

Reply to Eric, clean up MSE code

--------Post lunch, dang, can't access codelab

Changed password, hint: When_WHERE

Done!

Follow-up with Debbie(R41T/22 STI Integrated) & Michael(R41T/GH Contact)

Done!

Working on where I left-off a couple weeks ago...

Objective: Create merged DataFrame of etch_time & wafer_param_time

Notes: Completed forming etch_time & wafer_param_time, but I realize one etch_time is a dict and wafer_param_time is a DataFrame

Turned etch_time into a Dataframe and successfully created a combo of the two

12/3 Weird TSteps

Tuesday, January 21, 2020 10:54 AM

Notes 2019.12.03

Left-off:

Plotted scatter plot of peaks v ext_mv. Not looking super linear, there's a couple outliers, but I don't think those are the real issue, I think the real issue is in the trace chosen.

Emails: Reply from Stan Johnson RE: R41T Process-GH Contact etch asking "why this level was picked to use for a potential VM OES model. It sounds like GH module is still in a lot of flux. Actually kind of wondering how we ended up focusing on EM levels for this project"

- Replied with Robert CC'd in there as he could best respond to the reasoning behind our focus

From Yong RE: R41T Process-GH Contact etch, step4 and OESIB2 are the step and trace I should use for developing VM. Also: "Please explain me what you would like to do then I can explain you little better."

- Kinda already explained what I'm doing.

From Dave: Step 6 of the R41T 22 etch is the nitride etch. The inflection points probably correspond to where the nitride and pad oxide are clearing before we start to etch the silicon. There may be some correlation to cd for this step. I am wondering what your step 12 trace looks like as this is the silicon etch. You may not see and infections as this is a stop in film step. Step 2 is the darc etch and step 4 is the carbon etch. These are CD sensitive steps and you may see a correlation between CD and the traces form these steps.

- Displayed step 12, the noisy one,

Objective: Create updated powerpoint on progress thus far hopefully with some good R-squared values. Due Thursday.

What the hell is going on, I returned to GH_Contact_EXT and things are WEIRD.
My time steps are WAY off:
print(x, UniqueWafers[x], Tsteps, t_sec[x])

---

55 CX8GQ372SEE0 1057103 88.409
1057105 88.309
1057106 88.209
1057107 88.109
1057108 88.009
...
1057635 40.109
1057636 40.009
1057637 39.909
1057638 39.809
1057639 39.709
Length: 488, dtype: float64 1056928 105.008
1056929 104.309
1056930 104.209
1056931 104.109
1056932 104.009
...
1058072 0.309

1058073 0.309
1058074 0.000
1058075 0.000
1058076 0.000
Length: 1149, dtype: float64

WHY, I'm really pretty clueless why this happened, I'm going into the original loop and see where things are messing up
So apparently in the wholekaboodle the t_sec are reversed, checking for 22 STI -- It's NOT reversed! What's going on.....
Leaving for class, dang

12/5 Update PPT

Tuesday, January 21, 2020 10:50 AM

Notes 2019.12.05 (8:30-16:30)

Left-off:I returned to GH_Contact_EXT and things are WEIRD. My time steps are WAY off.
WHY, I'm really pretty clueless why this happened, I'm going into the original loop and see where things are messing up. So apparently in the wholekaboodle the t_sec are reversed, checking for 22 STI -- It's NOT reversed! What's going on.....
Notes: Okay, so I think I got a fix for that with a try/except bit

Objective: Create updated powerpoint on progress thus far hopefully with some good R-squared
values. Due Thursday.
Notes: Okay, so back to my Left-off of weeks ago. Super cool.
So I'd been exploring 22 STI, but started wanting to apply it to GH Contact
Thinking I should go back to the emails and try and get some conceptual information on it before I go messing around.
For GH Contact
- Step 4 & IB2 are what I'm wanting to check out
For STI Integrated
- Step 2 is darc etch (Dielectric Anti-Reflective Coat. A nitrogen silicon film that prevents pattern distortion at Photo due to reflective light.)
  - Nothing of note in plot
- Step 4 is carbon etch
  - Nothing of note in plot, weird because this one should be where I start

- Step 6 is nitride etch
  - This is the one with traces C516, CN387, CO483
- Step 12 is silicon etch
  - This is the one with the osciallations and traces H656, C516, CN387, CO483
- Other steps with activity
  - Step 10 w/ BCl272
  - Step 14 w/ C516

To-Do: Robert wants a script update & a broader picture included in the presentation, done! Will check in
with him in a moment to make sure he's happy with the presentation

From graph notes:
Wanting 5 spot, getting confused with 2/3 spots. Let's differentiate! * Is the target peak
x_int[peaks]: [123.94309091 125.08848485*]
x_int[peaks]-x_int[0]: [1.71809091 2.86348485*]
x_int[-1]-x_int[peaks]: [17.18090909 16.03551515*]
y_int[peaks]: [160953.59263673 153810.95107909*]
first[peaks]: [-11227.86355897 -32128.22417487*]
second[peaks]: [1073146.05399261 420010.83397917*]

x_int[peaks]: [123.85032323 124.99656566 125.76072727*]
x_int[peaks]-x_int[0]: [1.52832323 2.67456566 3.43872727*]

x_int[-1]-x_int[peaks]: [17.38467677 16.23843434 15.47427273*]
y_int[peaks]: [170457.80454311 182111.34848402 129663.87037373*]
first[peaks]: [-24478.06516378 -156513.47546079 -17553.8827407*]
second[peaks]: [496546.13993332 1536531.69251392 498675.1118907*]

x_int[peaks]: [123.23181818 125.32037374* 126.45958586 127.40892929 130.44682828
140.13013131]
x_int[peaks]-x_int[0]: [1.70881818 3.79737374* 4.93658586 5.88592929 8.92382828
18.60713131]
x_int[-1]-x_int[peaks]: [17.08818182 14.99962626* 13.86041414 12.91107071 9.87317172
0.18986869]
y_int[peaks]: [183461.80449857 143113.91449272* 113756.18369157 95120.27902624
90252.72666266 75907.55575723]
first[peaks]: [-16192.37911138 -32866.86593423
* -30565.95043883 -10428.34578591 -12563.18029048 -7193.39109019]
second[peaks]: [2187905.16218964 773399.17077193* 523434.18979376 456249.9940037
434594.6169837   858221.55592851]

x_int[peaks]: [124.32536364 125.46560606*]
x_int[peaks]-x_int[0]: [1.71036364 2.85060606*]
x_int[-1]-x_int[peaks]: [17.10363636 15.96339394*]
y_int[peaks]: [158265.23160144 156018.45978287*]
first[peaks]: [-31957.76140062 -43077.58437136*]
second[peaks]: [1448339.800229   708243.49302117*]

Back to Peaks! I was able to work through it and got to a consistent place with our peak...
(this is a lie, not really consistent if the R-squared value is terrible)
... identification. But the R-squared value is leaving a bit to be desired.

1/7 Smoothing?

Tuesday, January 7, 2020 2:13 PM

Notes 2020.01.07(9:15-4:15)

Left-off:See 20191205 powerpoint presentation to see where currently at

Notes: Currently, cathcing up on emails, got me some stock options, chyeah!
In attempting to narrow down to a single transition point, the smoothing technique yields
inconsistent results, meaning that marking a specific transition point isn't consistent.
So...
Options - Redo smoothing so that more consistent peak marking occurs
- Find a better way identify the desired transition point

Taking stock...
- ProcessData.py: Chugs thru 22STI, plots C516 traces for step 6 and plots the
resulting linear regression attempt with an R-squared of 0.05652 in range[15.0,16.4], these etch times are derived from:
peaks = peaks[first[peaks]<0]
peaks = peaks[y_int[peaks]<2.6e5]
peaks = peaks[x_int[peaks]-x_int[0]>2.5]
Which are derived from the stats taken in notebook page(-3 & -4).
- 22STI-Step6-OES_C516.py: Pretty much the same thing, I must havge made this so
that I can continue tweaking ProcessData.py but not lose progress I've made.
- 22_STI_Integrated_EXT.py: 22STI, plots Wafer Scribe and the identified peaks of
the 22 wafers, which also get written to output with first example here:
x_int[peaks]: [121.38243434 121.76015152 122.51558586 122.89330303 123.64873737 125.53732323 139.13514141]
x_int[peaks]-x_int[0]: [0.75543434 1.13315152 1.88858586 2.26630303 3.02173737 4.91032323 18.50814141]
x_int[-1]-x_int[peaks]: [17.94156566 17.56384848 16.80841414 16.43069697 15.67526263 13.78667677 0.18885859]
y_int[peaks]: [200846.0497037 238945.52870371 245773.0319303 285583.05467476 195061.59424573 126392.23092397 94576.40270849]
first[peaks]: [512713.76697787 -51733.04181196 288221.01926548 -472245.34105602 -90043.6365338 -20101.43993834 108022.30426542]
second[peaks]: [2848736.26091821 470760.3492216 7805042.00465622 2674553.79502153 2045434.27212937 797559.92165941 2308444.27689132]
These metrics were taken and recorded on that previously mentioned notebook page
- WholeKaboodle.py: Currently, processing 22STI, but can work for GH. Plots the
full steps and traces of a single wafer with color coordinated steps, done for
all trimmed down wafers.
- GH_Contact_EXT.py: Processes GH, and plots OESIB2 traces for all wafers with
peaks. From this first plot, I can already see that I'm going to need to figure
out a way to clean up and outlier in this one. Gross.
- Scratch.py: My best work yet.

Okay, so I'll need to head out to lunch soon, but for now, I'll set up ProcessData.py to
pick up on and play with smoothing, or play with peak marking.

Regarding smoothing!
Currently cmoothing with scipy.interpolate.splrep. Okay, I'm playing around some more
with smoothing, yielding consistent peaks and then triming down peaks. Seems that all the
peaks I'd want can get really trimmed down by 'peaks = peaks[first[peaks]<0]'. Nice...

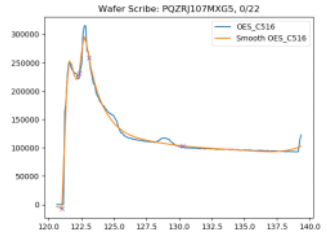
Trying this out and checking out the resulting R-squared value...
Oof. R-squared: 0.00596534
Dang, so not quite. Digging into Eric's procedure would probably be a lot more helpful
than my flailing.
Hm... So I got ahold of Eric's smoothing/differentiating script and he creates spline
coefficients for X&Y with a s = 5 (I believe, JMP documentation is predictably sparse)
Okay, so JMP documentation state that this lambda (=5 in Eric's script) value is a tuning
parameter in the spline formula.
Trying out k = 3 and s = 5 with a first[peaks] must be greater than zero.
Kinda messy, there's inconsistencies in Wafers: 4, 12, 14/22.

Looking into why these ones have weird peaks.
Wafer Scribe: DC1SQ073SEB0, 4/22 -- No peaks at the 1,2,3 spots. Checking 2nd Derivative, never
reaches a maxima, going to explore tweaking s values some more...

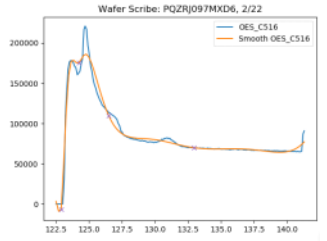
Maaaaaan, smoothed with k = 3, s = 1e10 and
peaks = peaks[first[peaks]<0],peaks[x_int[peaks]-x_int[0]>2.0]
Buuuuuut... R-squared: 0.01061598704116058, and it's a negative slope.
WOOF.

Leaving off: EITHER continue tweaking smoothing and peak identification, OR

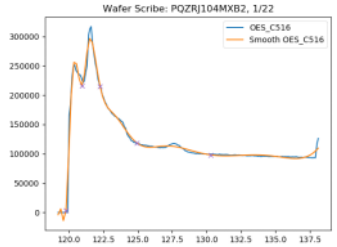
WaferScribe	#	x_int[peaks]	x_int[peaks]-x_int[0]	x_int[-1]-x_int[peaks]	y_int[peaks]	first[peaks]	second[peaks]
DJ2AQ003SEA2	21/22	123.14194949	1.51894949	17.27805051	176577.3568483	23302.78023961	6.55665998e+04
DJ1SR010SEA2	20/22	124.13532323	1.52032323	17.29367677	170065.35018332	9147.0079219	1.85121181e+05
DJ1SR408SEE1	19/22	123.53051515	1.51151515	17.19348485	167273.69238959	23229.47738883	3.65957390e+05
DJ1SR405SEG1	18/22	122.84370707	1.51870707	17.27529293	177864.15242363	21780.38400767	6.81721626e+04
DH8QG300SED0	17/22	122.44408081	1.52008081	17.29091919	192062.13351038	41061.92495441	3.28285424e+05
					Always positive!		



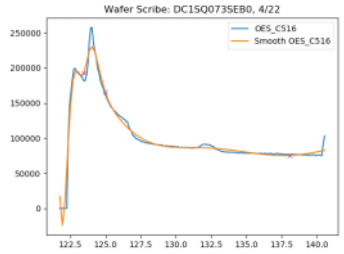
Splrep: k = 5, s = 1e10



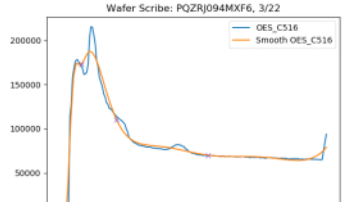
Screen clipping taken: 1/7/2020 2:19 PM

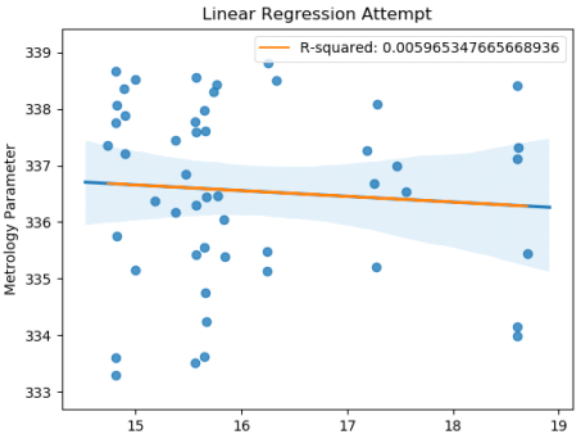


Screen clipping taken: 1/7/2020 2:20 PM



Screen clipping taken: 1/7/2020 2:20 PM

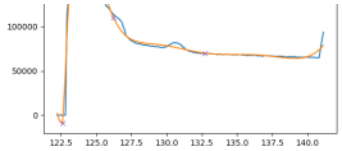




Oof, not exactly what I was hoping to see...

peaks = peaks[first[peaks]<0],peaks[x_int[peaks]-x_int[0]>2.0]
Buuuuuut... R-squared: 0.01061598704116058, and it's a negative slope.
WOOF.

Leaving off: EITHER continue tweaking smoothing and peak identification, OR pursue Eric's
scripts to get a better idea of how he develops models.



Screen clipping taken: 1/7/2020 2:20 PM

1/8 Robert Chat

Tuesday, January 21, 2020 10:51 AM

Notes 2020.01.08(11:15-)

Left-off:EITHER continue tweaking smoothing and peak identification, OR pursue Eric's scripts to get a better idea of how he develops models.

Let's do some more smoothing and peak identification, then I'll start harassing people.

Notes: Problem wafers to look into further...
Wafer Scribe: DC1SQ267SEE3, 27/50
x_int[peaks]:
[119.06508081]
y_int[peaks]:
[129380.49368721]

Wafer Scribe: DA8LP169SEF0, 43/50
x_int[peaks]:
[121.79376768 128.85441414]
y_int[peaks]:
[273434.57787084 106018.6036469]

Okay, so weird discrepancy going on. UniqueWafers in Scratch.py (taken from 22_STI_Integrated_EXT) is not matching UniqueWafers in ProcessData.py.
Oof. Scratch/22_STI_Integrated_EXT were only taking in one csv file. Fixed it! Back to exploring...

So luckily the peak value for etch time is taking in just the first of x-int[peaks] so we can move past Wafer#43 and focus on why Wafer#27 is so weird.

Okay! Heading out for lunch now, but I'll want to check out Wafer#27. I want to try adjusting s=1e9 and see if we get the same peak point we're hoping to see. I also want to see if the peak trimming parameters I have in are part of the problem. Not sure which to do first, altering the s-value may mess up more and create more issues, but investigating the peaks may be relatively quick. Or at the very least, easier to walk away from.

Okay, so I ended up doing peak investigating and switched the transition point marker I'd been going for. Now I'm wanting to hit that dip. Went ahead and pursued that and got a familiar scatter with aligning points around every 0.10 etch time. I know I've explored those aligning segments before with no feedback, but investigating the first one (in a plot with four aligning segments), yielded a much better looking plot. R-squared=0.404, which isn't amazing, but much better than the trash I've been working with.

In etch time range [16.98,17.02] we have an R-squared of 0.4043830712604732
including wafers: 0,1,6,7,15,19,22,35/50
In etch time range [17.07,17.12] we have an R-squared of 0.01668878588271478
including wafers: 4,5,9,10,12,13,18,21,24,25,27,28,30,31,32,34,36,37,38,40,41,45,49/50
In etch time range [16.98,17.02] we have an R-squared of 0.003251037830976933
including wafers: 3,11,14,23,26,39,43,46,47/50

Okay, I think chatting with Robert about my direction will be best. The first alignment looked

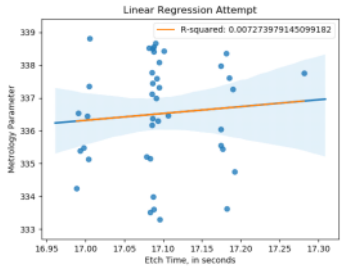
great, but the next couple were pretty dismal. We'll see if he's able to give me some insights and direction that feels more confident.

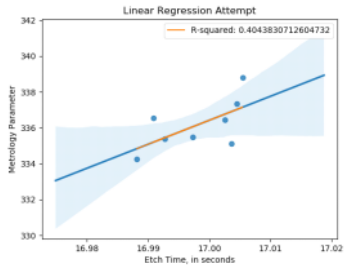
...Okay.

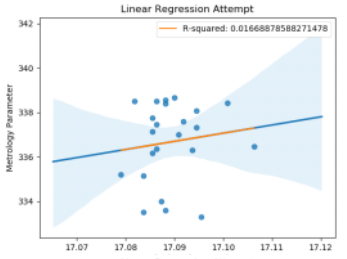
1/13 Emails, Updates, & Linear Regression

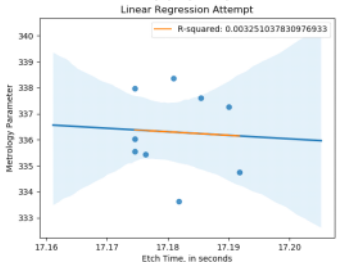
Monday, January 13, 2020 8:29 AM

These are the Lin Regs for full etch time (end point - transition point)

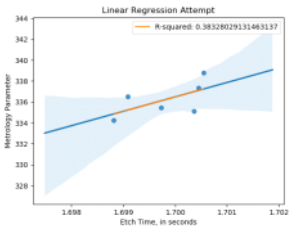


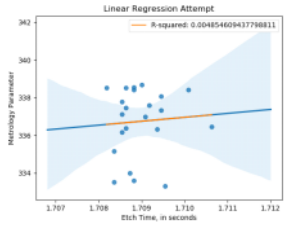


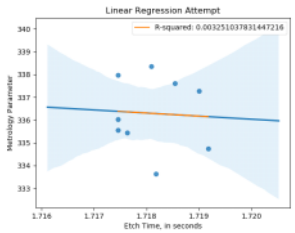




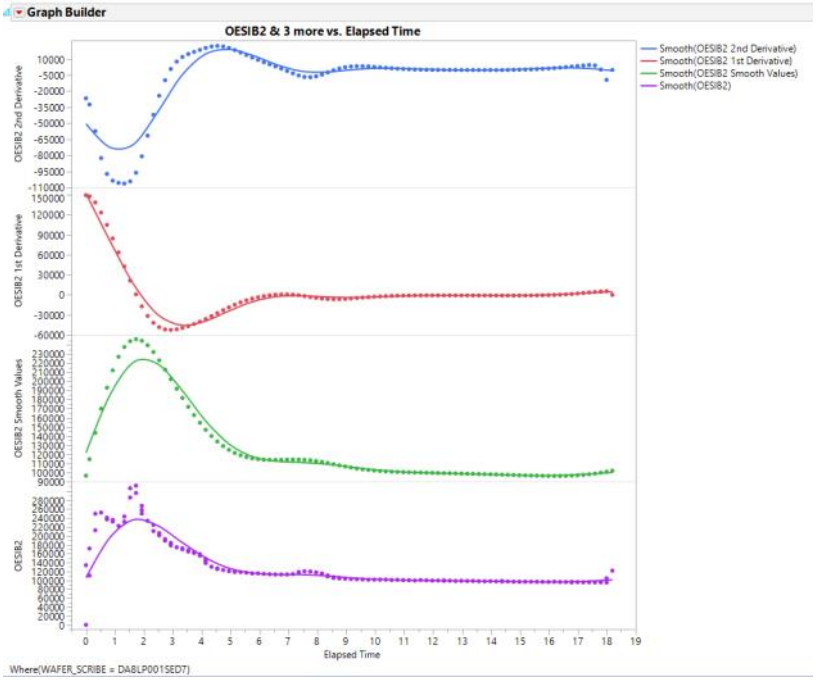
These are the Lin Regs for time to transition (transition point - start point)

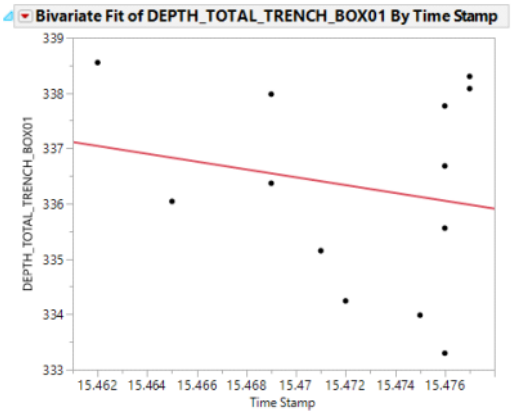






So here's my results from Eric's procedure





Linear Fit

DEPTH_TOTAL_TRENCH_BOX01 = 1431.6942 - 70.79619*Time Stamp

Summary of Fit

RSquare	0.038234
RSquare Adj	-0.0492
Root Mean Square Error	1.820249
Mean of Response	336.3083
Observations (or Sum Wgts)	13

Lack Of Fit

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	1	1.448883	1.44888	0.4373
Error	11	36.446358	3.31331	Prob > F
C. Total	12	37.895240		0.5220

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	1431.6942	1656.461	0.86	0.4059
Time Stamp	-70.79619	107.0592	-0.66	0.5220

Notes 2020.01.13(8:15-4:15)

Left-off: Okay, I think chatting with Robert about my direction will be best. The first alignment looked great, but the next couple were pretty dismal. We'll see if he's able to give me some insights and direction that feels more confident.

---Email to Robert---

With the new semester finally starting for me, my schedule will slightly shift from Tuesday/Thursdays to Monday/Wednesdays.

Moving onto the VM update, I have a script that:

1. Takes full OES and Metro data files
2. Parses through and trims down the OES data to contain only the matching Metro wafers
3. Identify extrema through smoothing and finding the first and second derivatives
4. Identify a transition point (currently more of a manual identification until I get the hang of it in other processes)
5. Plot etch time (end point - transition point) versus metrology parameter values and produce an R-squared value to measure the fit.

As noted in my recent slides I'd sent to Robert, the Etch time v. Parameter plots aren't yielding any apparent trend, so I've set out to tweak the smoothing techniques and transition point identification. A common linear regression plot I've been coming across is this one:

Where we can see the section of alignments forming across the etch time. Exploring these alignments further yields a mixed bag of results:

In the first, there's a hopeful R-squared value to pursue, but then the next couple alignments throw that hope away.

I'm going to continue playing around with the smoothing and peak identification techniques, but I think

it would also be prudent of me to run these transition points through Eric's (the previous intern working on this project) OES VM JMP scripts. In Eric's end of the year presentation he'd mentioned encountering a process where the etch time versus parameter technique didn't apply. If this process (R41T, 22 STI Integrated) happens to be one of those, I'll be able to move on to a new process and hopefully have better luck with that.

My next steps this month will be following up with Robert and Shannon to retrieve Eric's JMP scripts to run the 22 STI Integrated data through, follow up with the Stan and the process owner(s) of 22 STI and confirm transition point identification within this step, and continue to play with the smoothing technique and peak identification to make way for semi-automating transition point identification.

Specifically for Chris and Adrian, does this linear regression aligning look familiar at all? I'm thinking there may be some rounding possibly going on in the end point time stamp, but I haven't fully looked into it yet.

Moving on: Playing around with Etch-time, plotted parameter v. transition point, clearly didn't factor in the varying process length times. Now trying out (transition-start time) trimming the data but I suspect I'm going to see the alignment issue. Yep, I'm going to investigate the alignments, but really it should be the same results I've seen before in end point-transition point. And they are.

Spoke with Robert:
Keep playing around with smoothing techniques and bettering those.
Excellent, meeting with Eric this semester will help with that, too.

Follow-up with Robert & Shannon:
Delete Erroneous Data.jsl
"Open the JSL script that calculates and exports the dry etch time stamp"

Objective: More uniform smoothing technique
Notes: Playing around on 22_STI_Integrated_EXT.py
Actually, I'mma switch this to Scratch.py, I don't wanna mess up precious previous versions

Interlude! I've received all of Eric's JSL files, going to try to recreate his procedure with 22 STI. Ha, so he smooths the crap outta these bad boys, which won't work super well with STI Integrated because there's some significant features that can't just be smoothed over. Also, in his procedure he states that his procedure can only be applied to LAM tools, and tools that sample at a frequency of 10Hz.

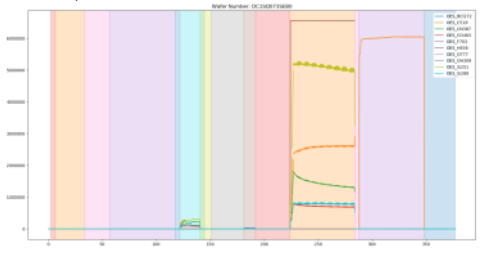
Went back to adjusting the smoothing technique. Applying the Savgol_filter from scipy. Pretty well adjusted, now onto the derivatives.

To-Do for next day: Email Dominic Seo about 22 STI

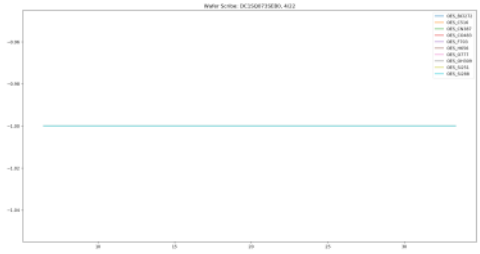
1/14 Process Owner Convo

Tuesday, January 14, 2020 11:35 AM

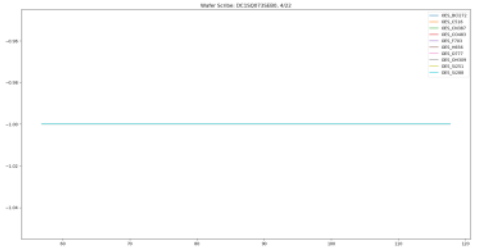
W/RT 22 STI:
(Sent to Dave Keller, Debbie Dando, Robert and Stan)
Whole Kaboodle yields:



Step 2, a Darc Etch [Dielectric (Deposited) Anti Reflective Coating] which has CD sensitive steps that may have a correlation between CD and the traces.
{Personally, I haven't seen any of that behavior within those steps and would need clarification from him regarding that.}

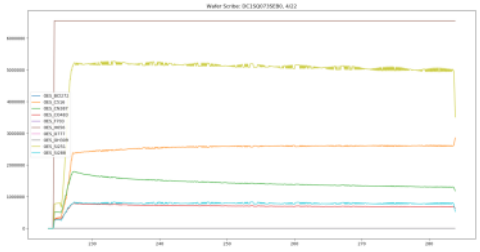


Step 4, a Carbon Etch, also a CD sensitive step that would have a correlation between CD and the traces.
{Again, I haven't seen any of that behavior within those steps and would need clarification from him regarding that.}



Step 6, a Nitride Etch has inflection points where the nitride and pad oxide are clearing before starting to etch the silicon, with a possible correlation to CD here.
{This is the step I've been investigating all along. I think it would be smart for me to get some insight on the trace names with respect to this}

Step 12, a Silicon Etch, suggested that no inflections would be seen as these are a stop in film step.
{This is step has a lot of oscillating going on and an interesting plateau from OES_H656}



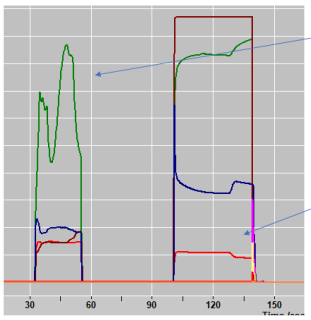
Step 10 has some interesting activity with the OES_BI272 trace

Reply!

1. what recipe are you looking at? Is it 44422IT1A6?

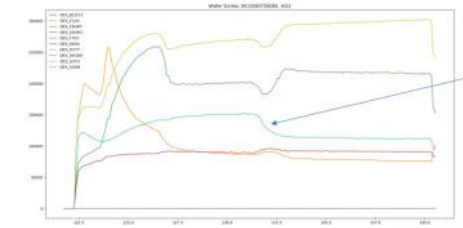
2. Here's steps 2 & 4, for the DARC and Carbon etches and so far the wafers I've been examining don't have any activity during those steps. Is that common?

You should see the traces change for step 2 and step 4. The traces should show the DARC and Carbon clearing. You must not be looking at the right wavelengths. For DARC step 2 you should look at 387 nm (Si line) for Carbon step 4 you should look at 520 nm (CO line). The DARC trace should look the green trace, and the carbon trace should look like the red trace. The photo CD controls the final cd but fluctuations in the DARC and carbon etch could change the final CD.



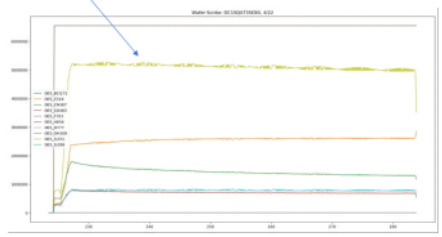
3. Here's the nitride etch step for this same wafer where the intensities C516, CN387, CO483, Si251, & Si288 are present. You'd previously mentioned that the inflections were due to the nitride and oxide pad clearing and I was curious if you'd be able to elaborate some more on those features and which intensity contributes to them? The naming scheme has given me some insight, but I'd be interested how their profiles relate to each other.

Nitride is Si3N4 when it etches with CF4 based chemistry the by products are SiF4 and CN. So you will see CN387 and Si251, & Si288 rise when the film starts to etch and then fall when the film is clear.



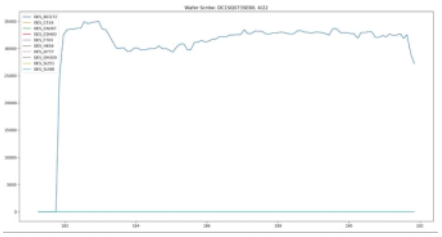
4. Here's the silicon etch step, you'd expressed curiosity about. I was wanting to know which intensity would be the best to explore possible CD relationships with?

Step 12 is the silicon etch step. I am not sure you will be able to se anything from this step. The Si etch step is the silicon substrate etching. We stop in the Silicon so there is no film that is clearing so the OES traces will not change as we etch deeper. The oscillation you see here is probley due to the pulsing of the plasma. We pulse the bias power for the Si step, meaning we tun it on and off which may make the plasma intensity go up and down which will make the OES trace fluctuate up and down.



5. Lastly, here's step 10, which doesn't always have activity among the wafers I've been investigating, but in the case of this wafer, there's a single intensity present. I was curious what occurs during this step and why it's only present in a few wafers?

Step 10 is a short break thru step. After the DARC, carbon, and nitride etch we run a short o2 flash to clean up polymer and then a breakthrough step which cleans the silicon surface before the silicon etch starts. Not sure why you see this on some wafers but not others.



Notes 2020.01.14(8:45-)

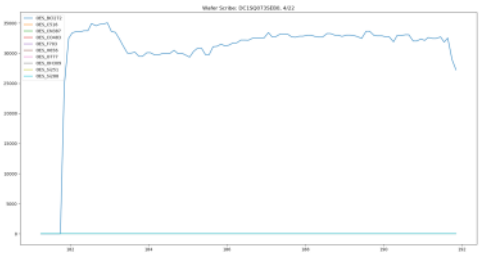
Left-off: Email Dominic Seo about 22 STI
Savgol Filter getting refined and check out what the LinReg looks like

----Now with Sanjay, with Brady!----

Following up the follow-up email from Dave, where he notes that R41T 22 STI has:
Step 2, a Darc Etch [Dielectric (Deposited) Anti Reflective Coating] which has CD sensitive steps that may have a correlation between CD and the traces.
{Personally, I haven't seen any of that behavior within those steps and would need clarification from him regarding that.}
Step 4, a Carbon Etch, also a CD sensitive step that would have a correlation between CD and the traces.
{Again, I haven't seen any of that behavior within those steps and would need clarification from him regarding that.}
Step 6, a Nitride Etch has inflection points where the nitride and pad oxide are clearing before starting to etch the silicon, with a possible correlation to CD here.
{This is the step I've been investigating all along. I think it would be smart for me to get some insight on the trace names with respect to this}
Step 12, a Silicon Etch, suggested that no inflections would be seen as these are a stop in film step.
{This is step has a lot of oscillating going on and an interesting plateau from OES_H656}
Step 10 has some interesting activity with the OES_BI272 trace

Done! Going to get rid of Lot_ID
Done! That was easy! Maybe I'll take on switching to Process_Elapsed_Time

Making headway! But there's a bunch of snow, going to check with Brady about driving home. Should go home and be safe, back tomorrow!



OES_H656)

Step 10 has some interesting activity with the OES_BI272 trace

Done! Going to get rid of Lot_ID
Done! That was easy! Maybe I'll take on switching to Process_Elapsed_Time

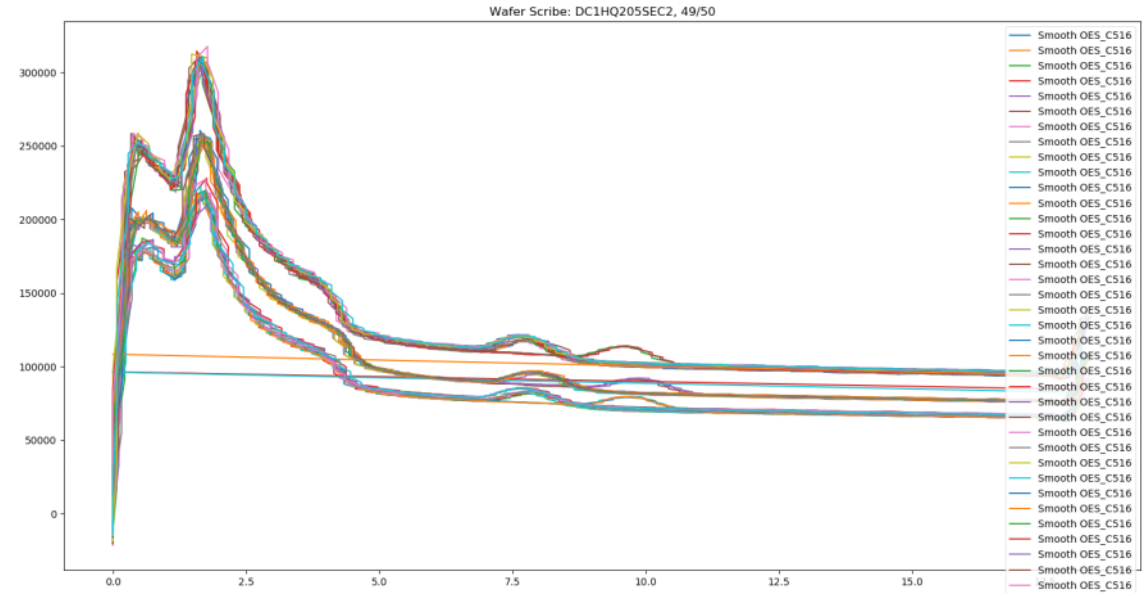
Making headway! But there's a bunch of snow, going to check with Brady about driving home.
Should go home and be safe, back tomorrow!

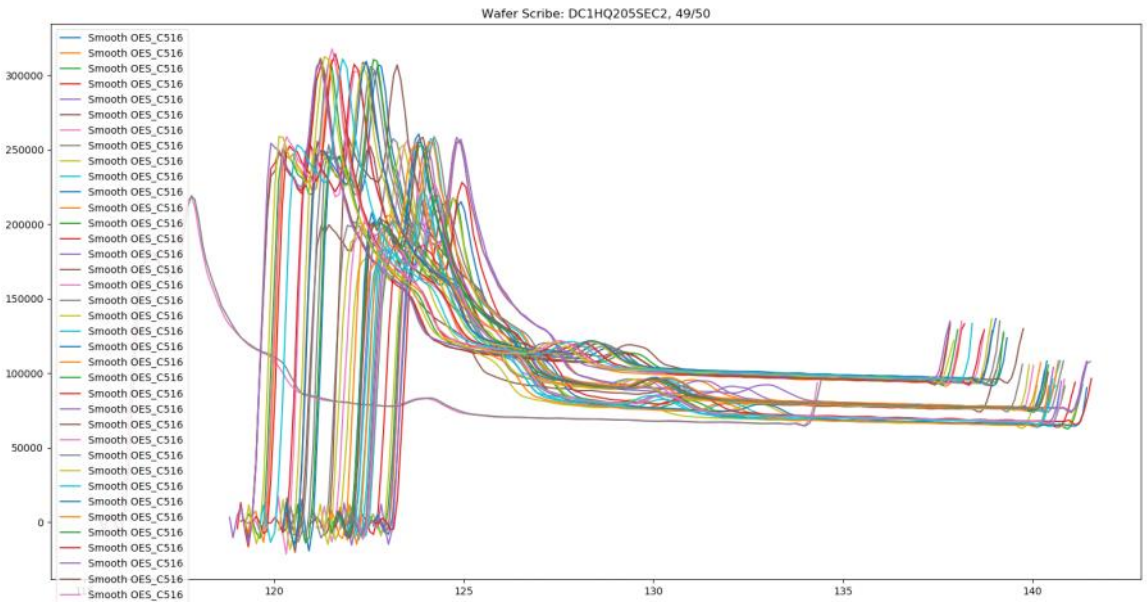
1/15 New Timesteps

Wednesday, January 15, 2020 2:13 PM

Okay, I think I fixed up the script so that ProcessElapsedTime is the new time frame unit for the traces, yields a much more blocky trace, and also didn't quite work with the interpolate.splrep and splev, so I checked if things worked with the savgol_filter. Savgol works but should be tweaked a bit. Though I am getting more peaks throughout the traces, which is always helpful to have more options to pick from.

Huh, so here's the trace of all the wafers using ElapsedTime , going to check back what the cumulative plot looked with for RunTime





Wooo, so the interpolate isn't great still, but makes for some groovy cumulative traces (k = 5, s = 1e11):

Notes 2020.01.15(11:30-16)

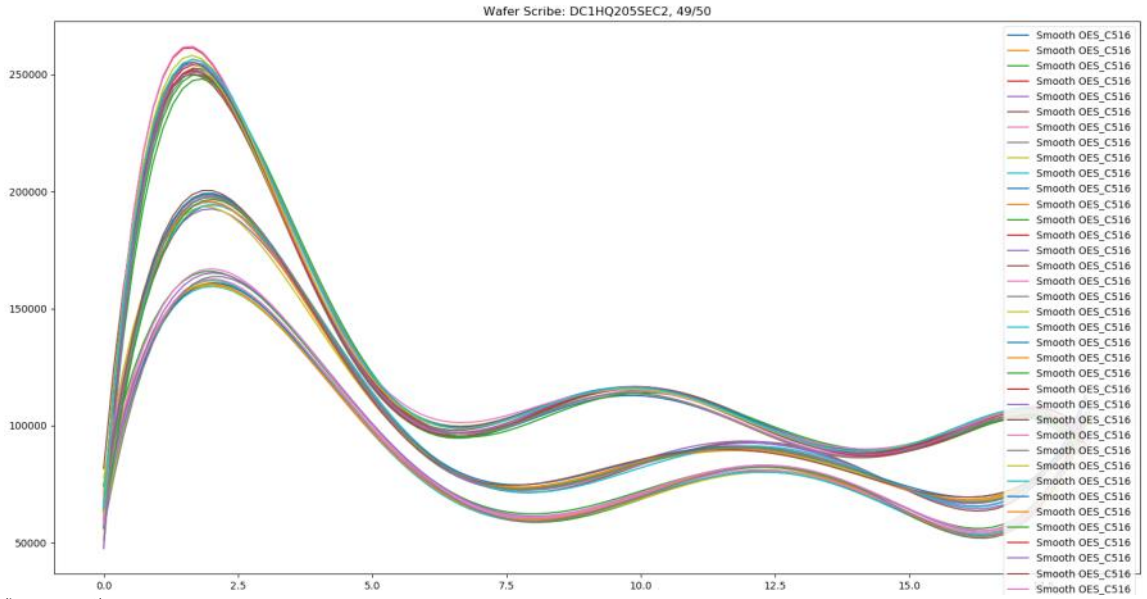
Left-off:Taking on switching to Process_Elapsed_Time

Got a reply from Dave regarding the weirdness for my 22 STI. He's referring me to Tino Martinez
- and Shaun Thompson, I went ahead and shot out a follow-up email and offered any
- additional information needed to help diagnose this trace.

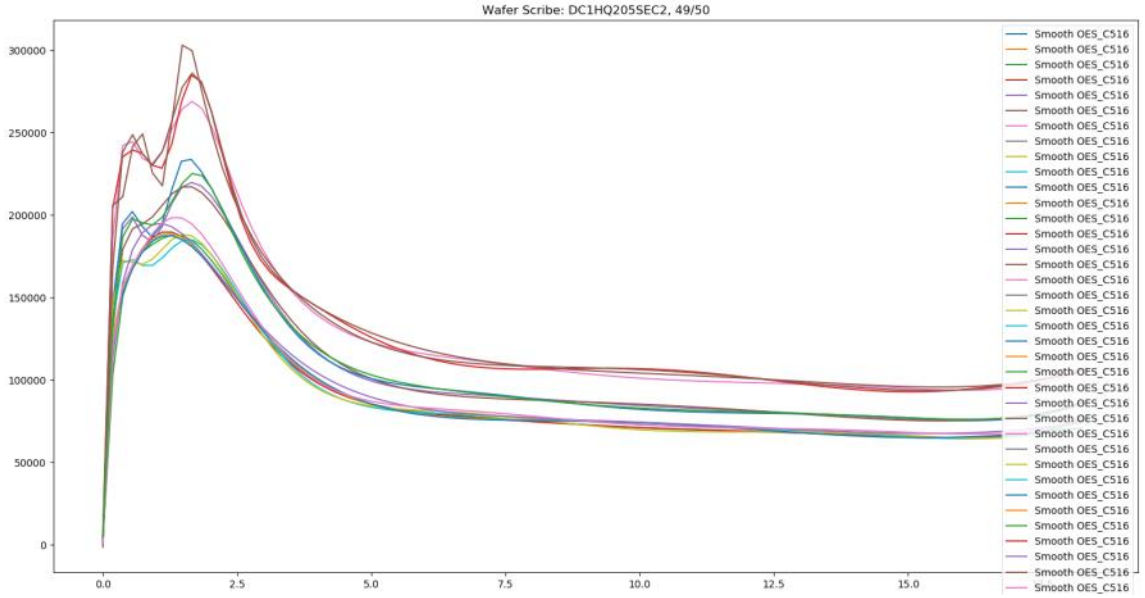
Lot's of communication on the missing data from 22 STI
- Shaun asked how data was retrieved
- Robert looped in Adrian/Chris to respond to those questions
- Dave clarifying which steps I'd want to check out
- Shaun asked Adrian some direct questions and answers Dave's inquiries
- Chris added Stan (again)
- Dave voiced appreciation
- Adrian replied and clarified what wavelength matches which EyeD

Okay, I think I fixed up the script so that ProcessElapsedTime is the new time frame unit for the
- traces, yields a much more blocky trace, and also didn't quite work with the interpolate.splrep and splev, so I checked if things worked with the savgol_filter.
- Savgol works but should be tweaked a bit. Though I am getting more peaks throughout the
- traces, which is always helpful to have more options to pick from.

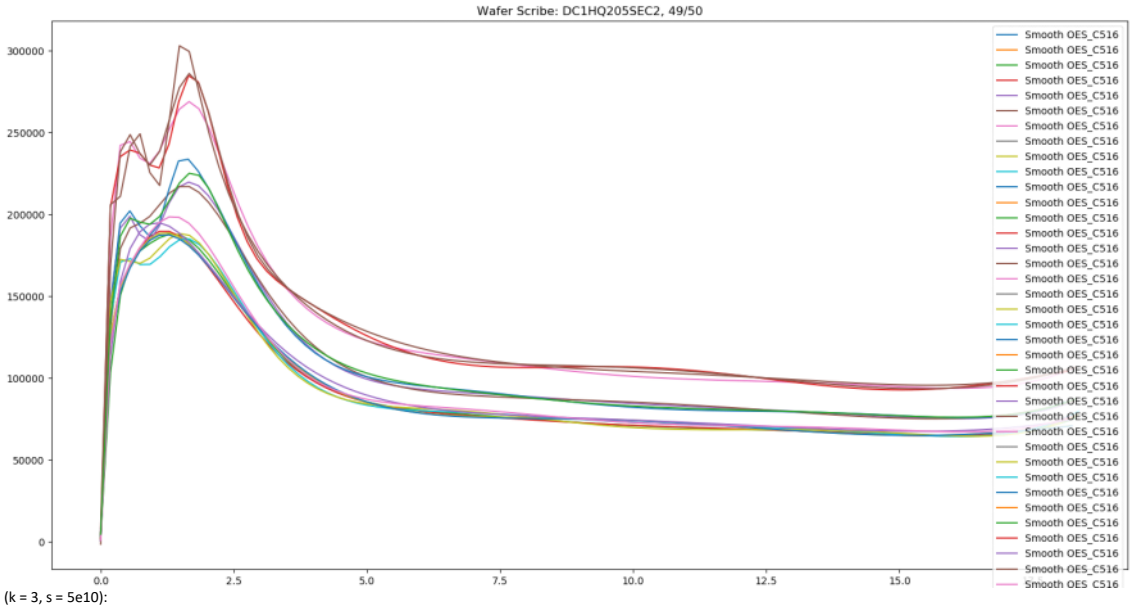
Playing around with the cumulative traces from switching to ElapsedTime. Pretty cool looking
- plots, next up on the list is tweaking the Savgol filter to make some smoother plots. If
- I recall, I need to figure out how long the new Tsteps is and work back from there for
- the window_length and play around with the polyorder.



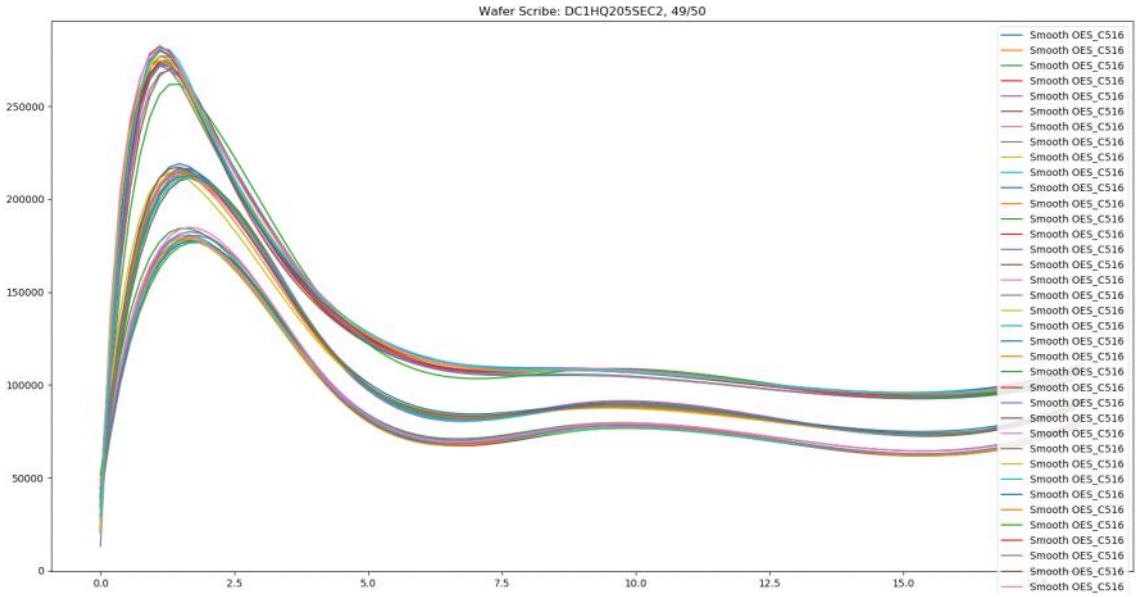
(k = 5, s = 1e10):



(k = 3, s = 1e10) Lookin' identical:



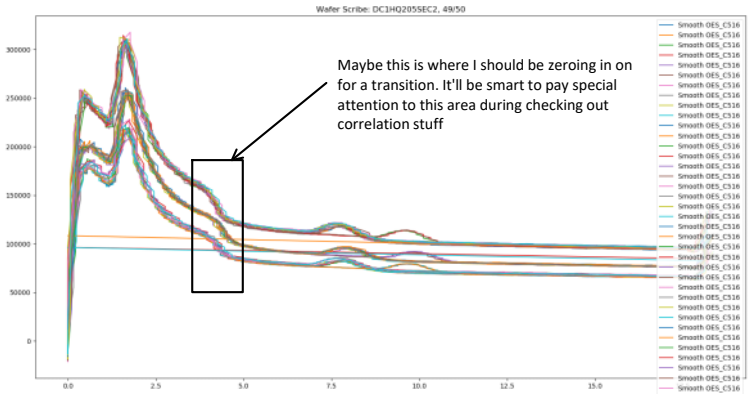
(k = 3, s = 5e10):



Screen clipping taken: 1/15/2020 3:14 PM

1/21 Git, OneNote, New Sched

Tuesday, January 21, 2020 10:30 AM



Left-off:

Playing around with the cumulative traces from switching to ElapsedTime. Pretty cool looking plots, next up on the list is tweaking the Savgol filter to make some smoother plots. If I recall, I need to figure out how long the new Tsteps is and work back from there for the window_length and play around with the polyorder.

Git Bash:

It's a no-go, I'm unable to download it for windows. Maybe I'll see if there's a pip install option?
Also a no-go, but I am upgrading pip for the heck of it

GitHub:

Uploading current python scripts, going to clean them up a bit for clarity, start back up again with GH_Contact_EXT.py.
Done!
Now I should probably include some output? Lemme look through the notes and see what we're working with already.

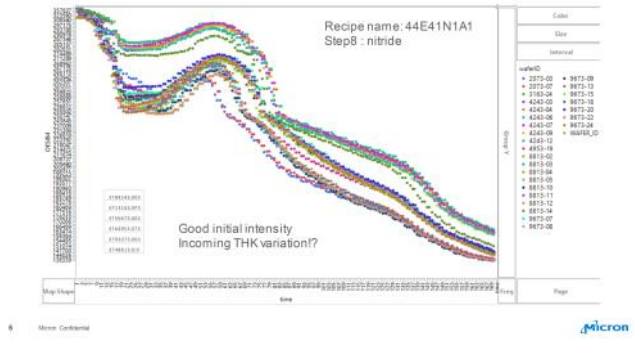
22 STI Update:

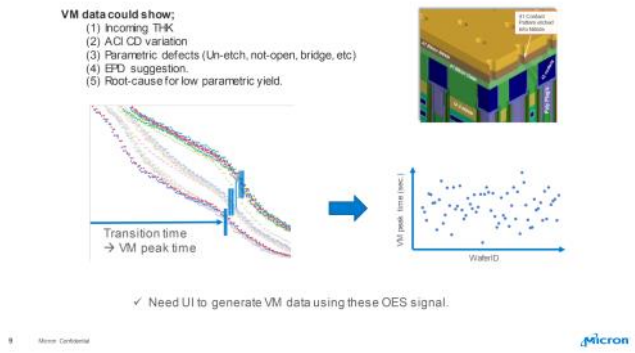
Waiting on Adrian (PCS) for the completed data, Shaun (Dry Etch Equipment Eng) modified the data retrieval algorithms, Robert said he was going to message him and get a follow up. Nothing needed from my end.

22 STI = Amat tool
GH Contact = Lam tool

Dominic Seo's VM presentation:

Stumbled upon Dominic's presentation, where he's applying VM to the nitride step (6):





1/22 SavGol

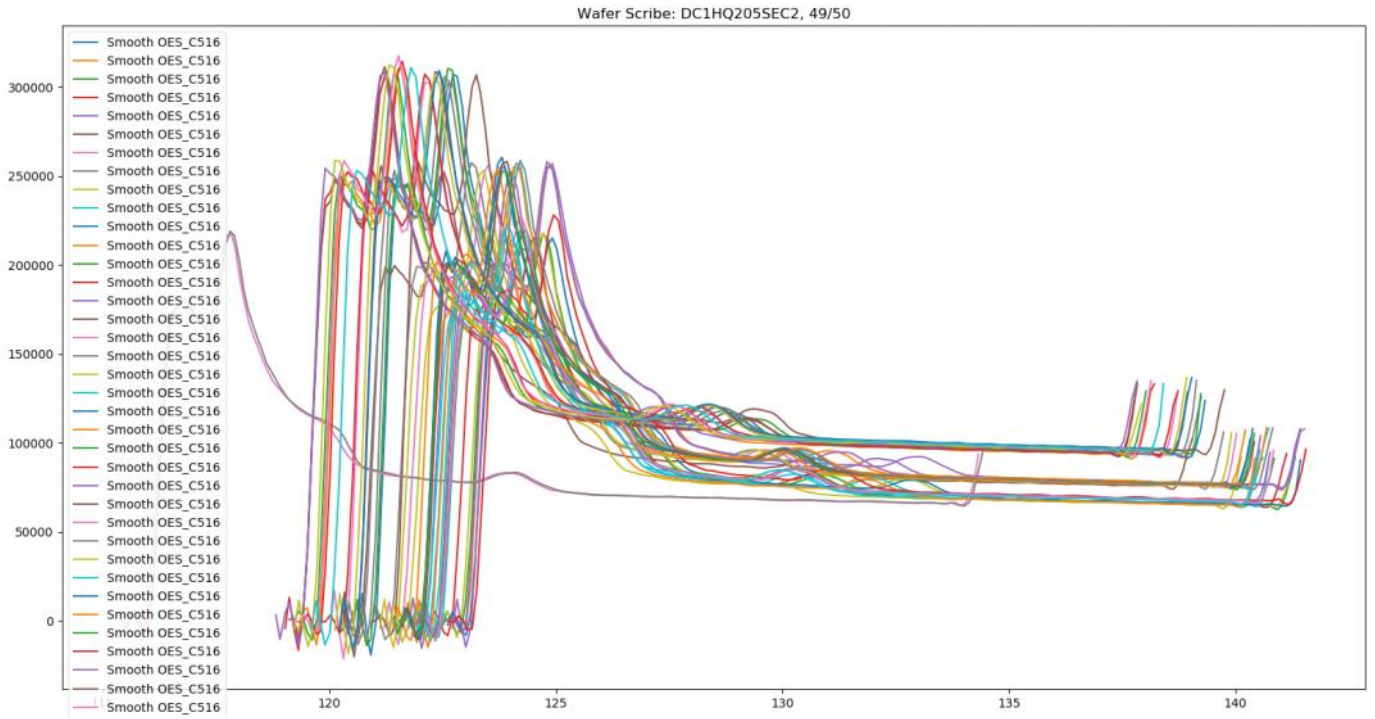
Wednesday, January 22, 2020 11:30 AM

Left-off:

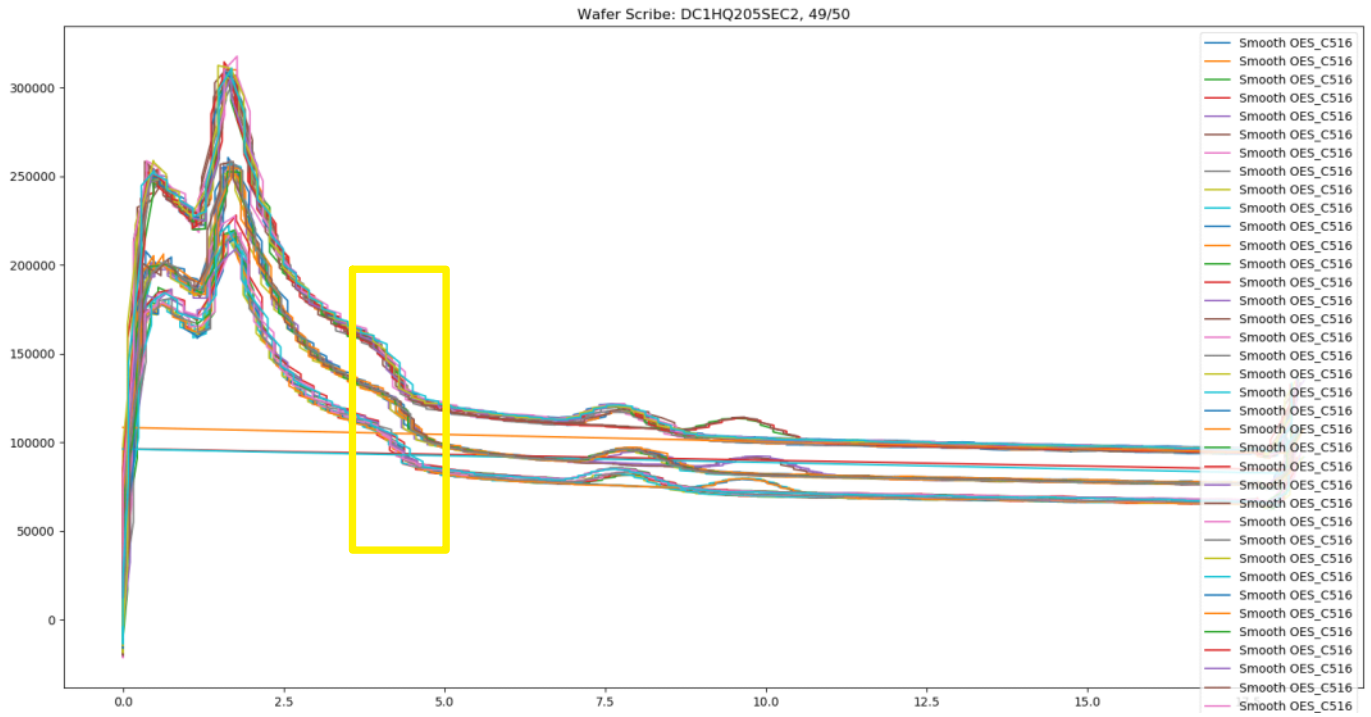
Nowhere in particular, though tweaking Savgol seems to be where I should have picked up on yesterday.

Savgol:

Running SavGolSmooth-runtime.py yields, for a window-length of 11 and polyorder of 5:

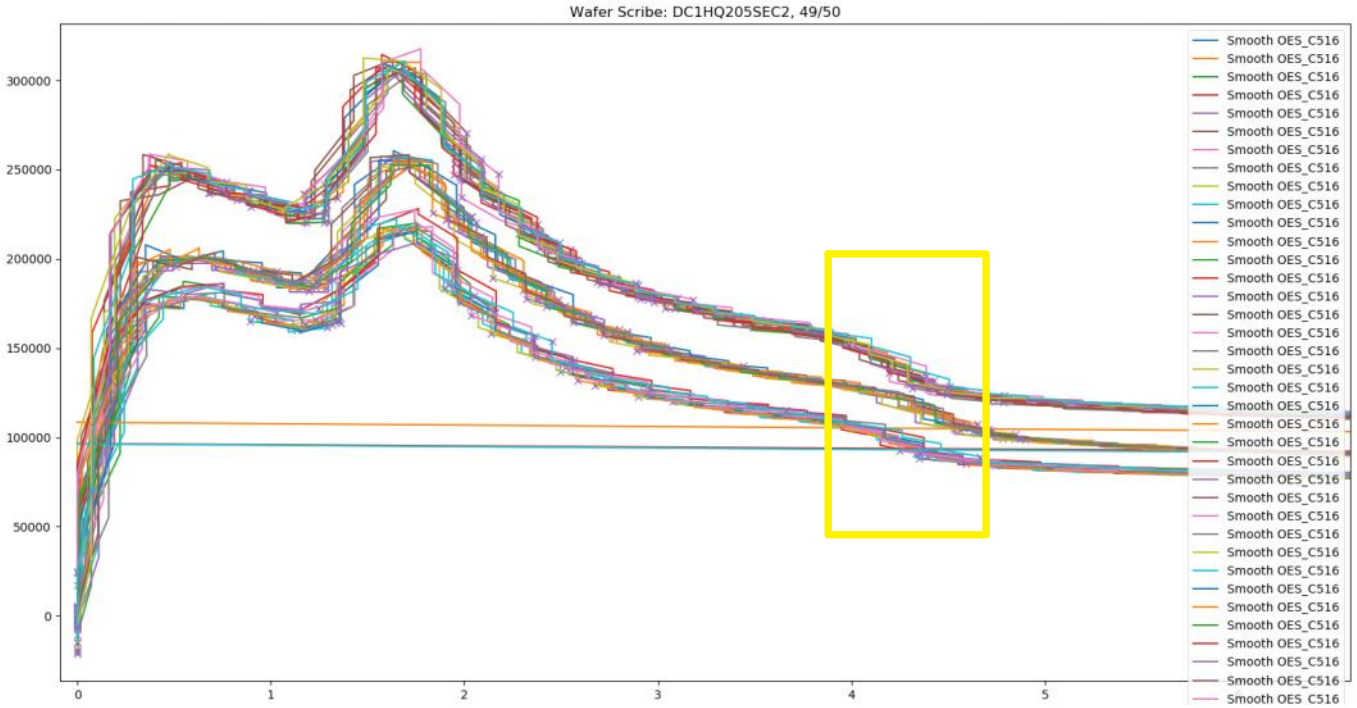


Running SavGolSmooth-elapsedtime.py yields, for a window-length of 11 and polyorder of 5:

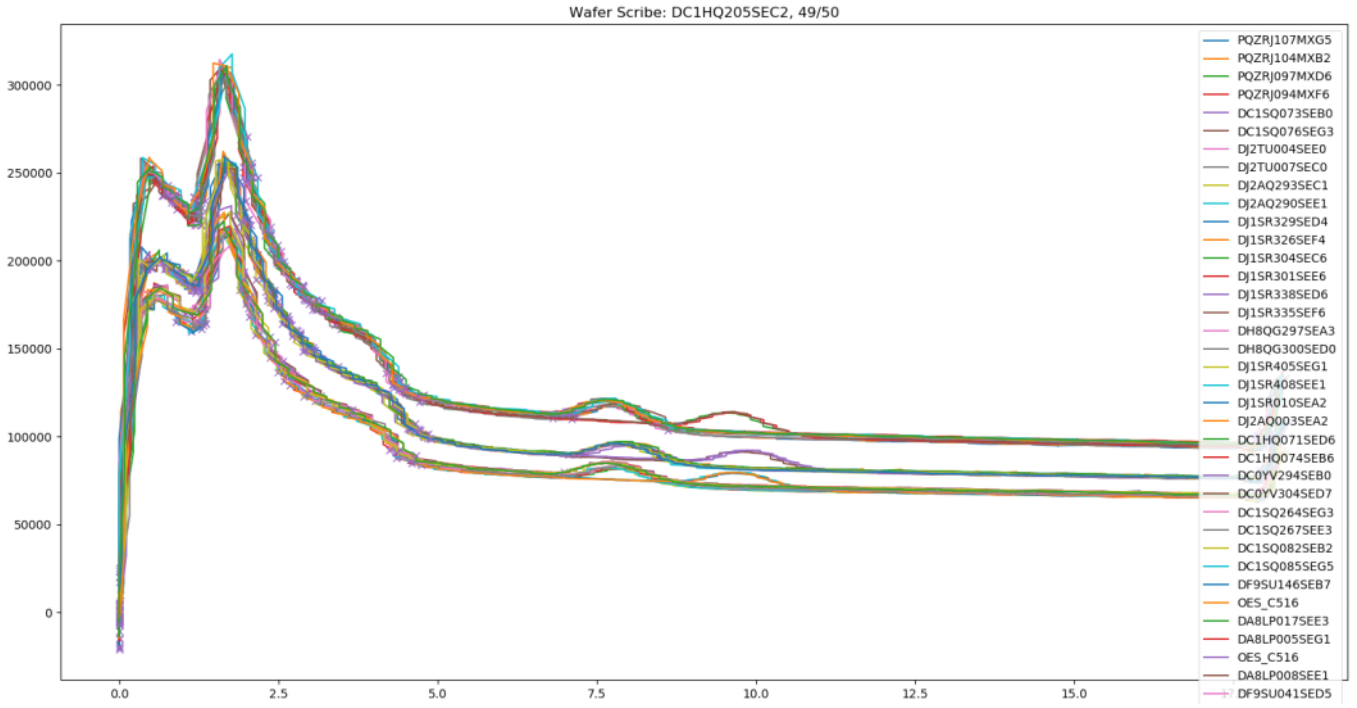


So I'm thinking I'm wanting to use the elapsed time more to be consistent with other model building techniques

within Micron.
I'm also wanting to check out what's wrong with those orange and blue straight lines and why they're so weird. Also, I'm wanting to see where the peaks are lying and see about getting peaks to include this transition point, which I think is where Dominic identifies a transition point correlated to the etch point.
Okay, so here's where peaks are at:



It's good to know we're at least identifying some peaks in here. Going to work on identifying the outlier traces.
Oh! I think what's happening is that for a given step, the trace time steps go back to zero.
Fixed it:



Okay, so I've been working on trimming down peaks, and the notation is different this time because it's a dataframe. I'll have to keep troubleshooting this.

1/27 Tweaking SavGol

Monday, January 27, 2020 8:00 AM

Following-up with Robert/Chris/Adrian on getting another process that will include that carbon etch step that we're wanting to perform the model building on.

Left-off:

Okay, so I've been working on trimming down peaks, and the notation is different this time because it's a dataframe. I'll have to keep troubleshooting this.

So, revisiting the data frame notation issue. I want to be able to constrain the peak values by saying "peaks = peaks[Tsteps[peaks]>4]". This was the lingo I used previously, but now the x & y values are DataFrames rather than numpy arrays.

ProcessData.py Input:

```
print(first[peaks])
print(type(first[peaks]))
print([first[peaks]>0])
print(type([first[peaks]>0]))
print(peaks[first[peaks]>0])
print(type(peaks[first[peaks]>0]))
```

ProcessData.py Output:

```
[ 247801.91500589   88756.81889823 -116230.32187122]
<class 'numpy.ndarray'>
[array([ True,  True, False])]
<class 'list'>
[2 9]
<class 'numpy.ndarray'>
```

SavGolSmooth-elapsedtime.py Input:

```
print(Tsteps.values[peaks])
print(type(Tsteps.values[peaks]))
print(Tsteps.values[peaks]>4)
print(type(Tsteps.values[peaks]>4))
print(peaks(Tsteps.values[peaks]>4))
```

SavGolSmooth-elapsedtime.py Output:

```
[[ 0.   ]
 [ 0.674]
 [ 1.277]
 [ 2.081]
 [ 2.282]
 [ 2.483]
 [ 2.885]
 [ 3.488]
 [ 4.493]
 [ 7.106]
 [18.152]]
<class 'numpy.ndarray'>
[[False]
 [False]
 [False]
 [False]
 [False]
 [False]
 [False]
 [False]
 [ True]
 [ True]
 [ True]]
<class 'numpy.ndarray'>
Traceback (most recent call last):
```

```
  File "c:/Users/nellyson/Documents/Projects/201910Metro-OES/SavGolSmooth-elapsedtime.py", line
106, in <module>
    print(peaks(Tsteps.values[peaks]>4))
TypeError: 'numpy.ndarray' object is not callable
```

Okay, looking into TypeError

Done! It looks messy, but that's a problem for a different day

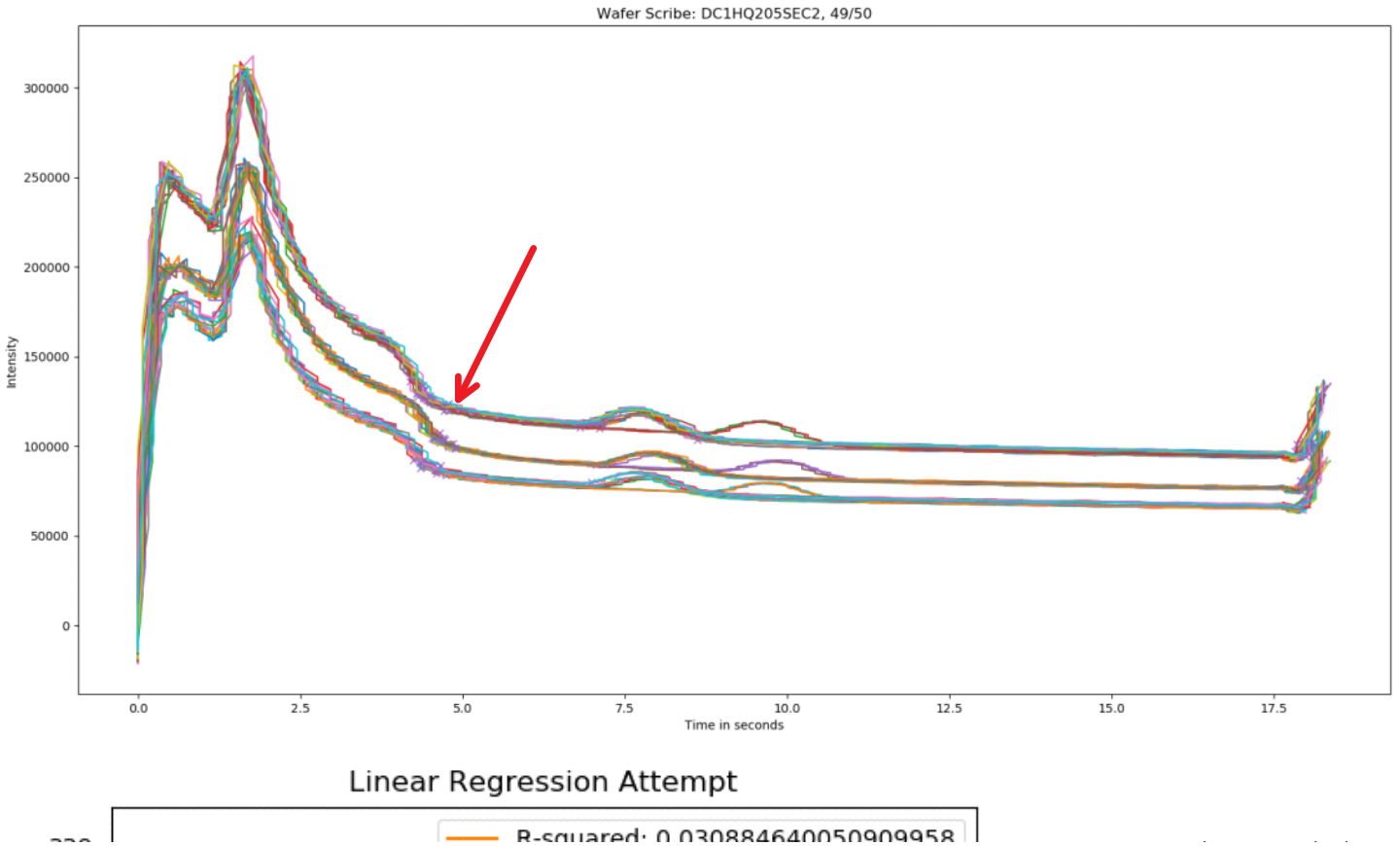
SavGolSmooth-elapsedtime.py Input:

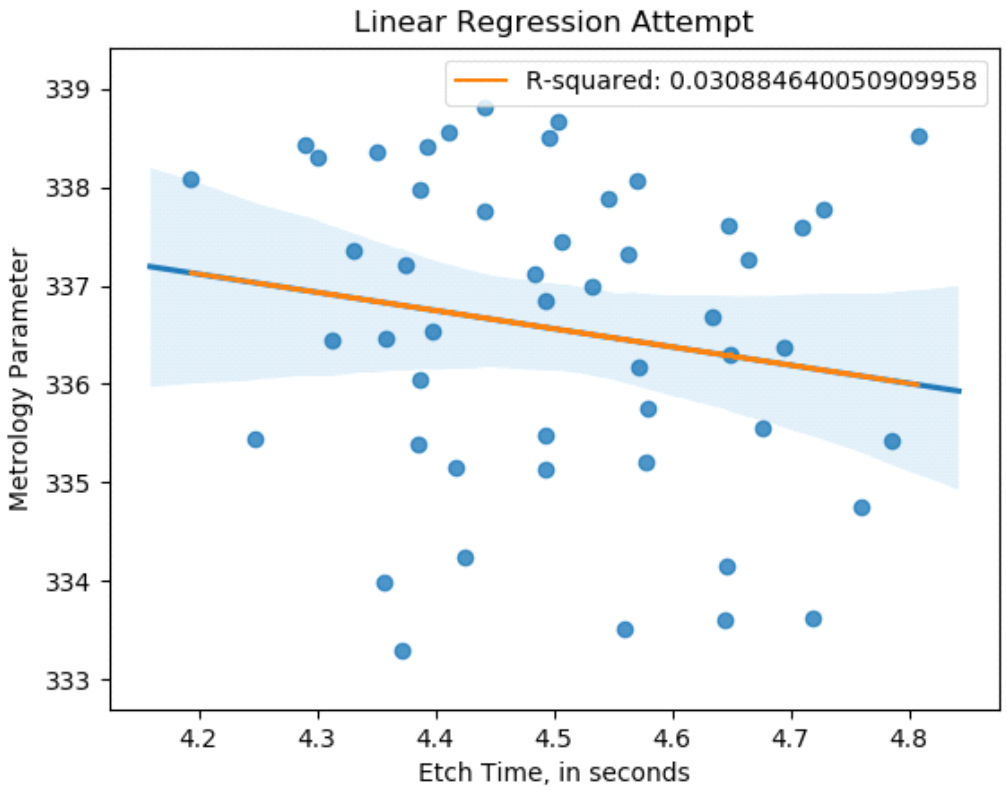
```
print(Tsteps.values[peaks])
print(type(Tsteps.values[peaks]))
print(np.concatenate(Tsteps.values[peaks], axis=None))
print(type(np.concatenate(Tsteps.values[peaks], axis=None)))
print(np.concatenate(Tsteps.values[peaks], axis=None)>4)
print(type(np.concatenate(Tsteps.values[peaks], axis=None)>4))
print(peaks[np.concatenate(Tsteps.values[peaks], axis=None)>4])
```

SavGolSmooth-elapsedtime.py Output:

```
[[0.   ]
 [0.889]
 [1.29 ]
 [2.094]
 [2.295]
 [2.496]
 [2.898]
 [3.099]
 [4.506]]
<class 'numpy.ndarray'>
[0.    0.889 1.29  2.094 2.295 2.496 2.898 3.099 4.506]
<class 'numpy.ndarray'>
[False False False False False False False False  True]
<class 'numpy.ndarray'>
[51]
```

Now I'm going to see what the linear regression looks like for the SavGol elapsed time script.

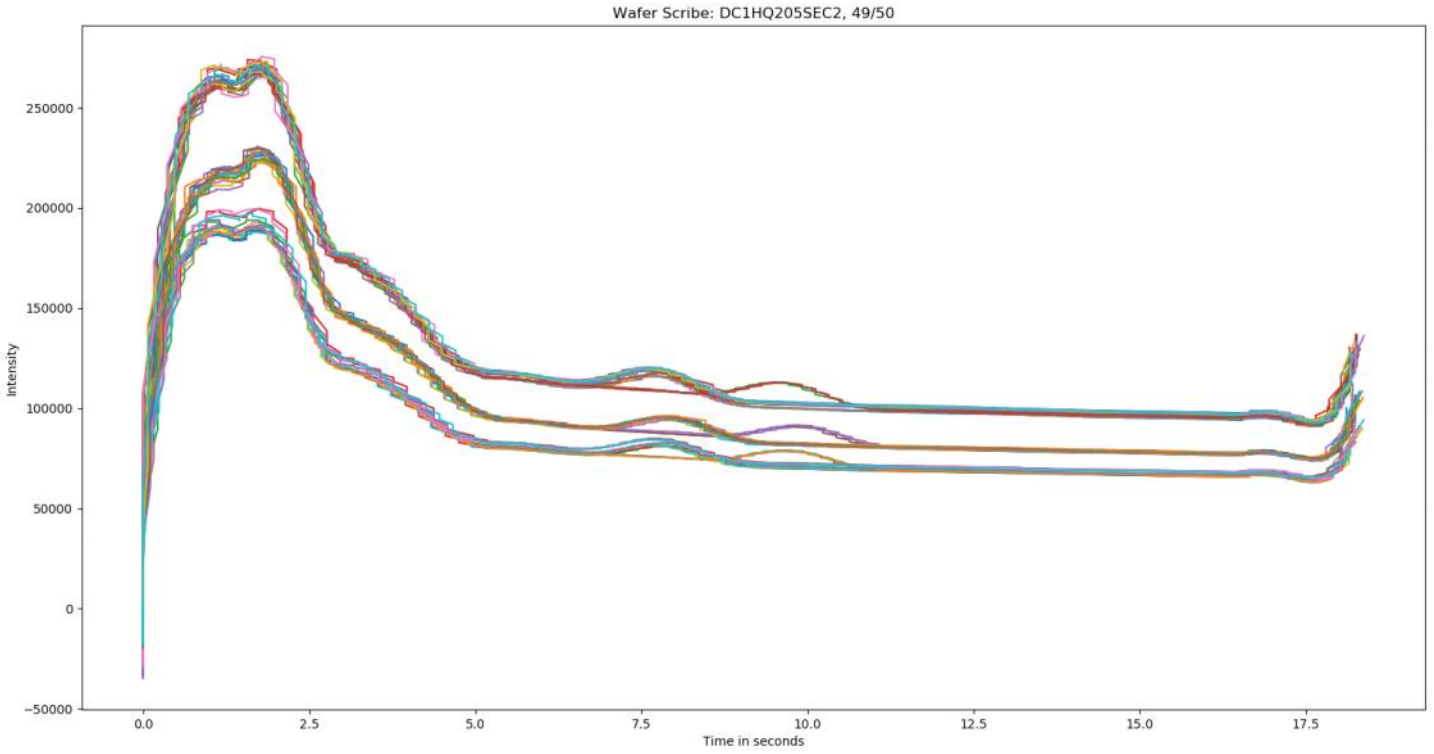




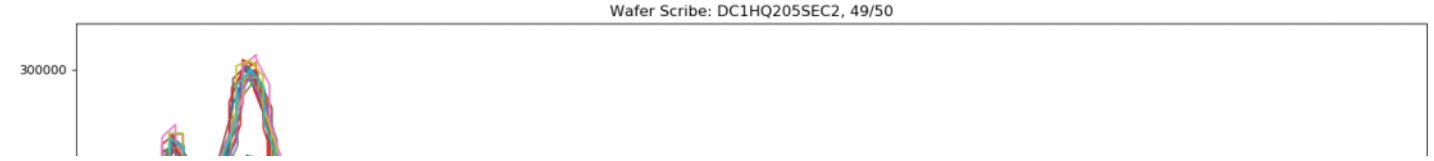
So, not a super cool R-squared. This would be some prime content to ask Dominic about. In this one, the peaks that I use are identified by the arrow. At least we don't have the alignment issue here.

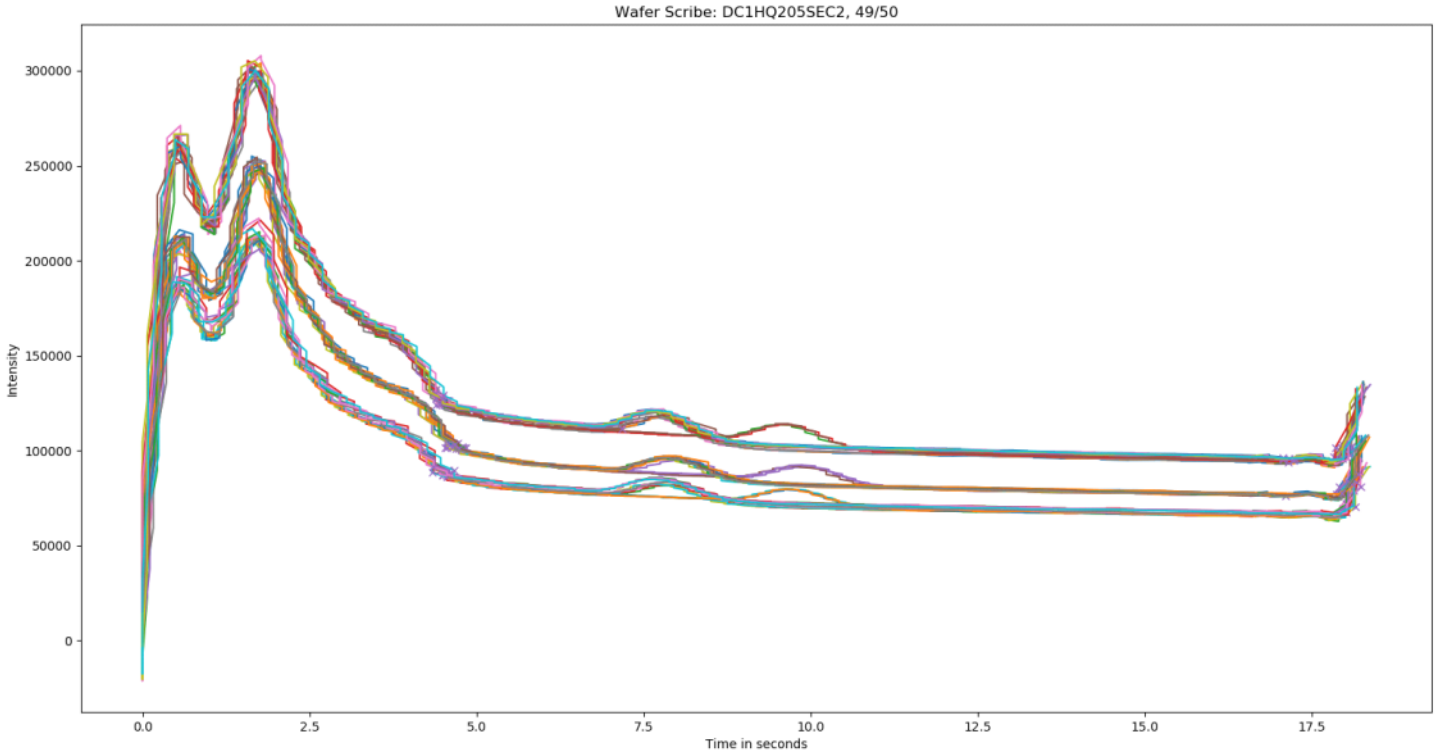
Playing around with smoothing:

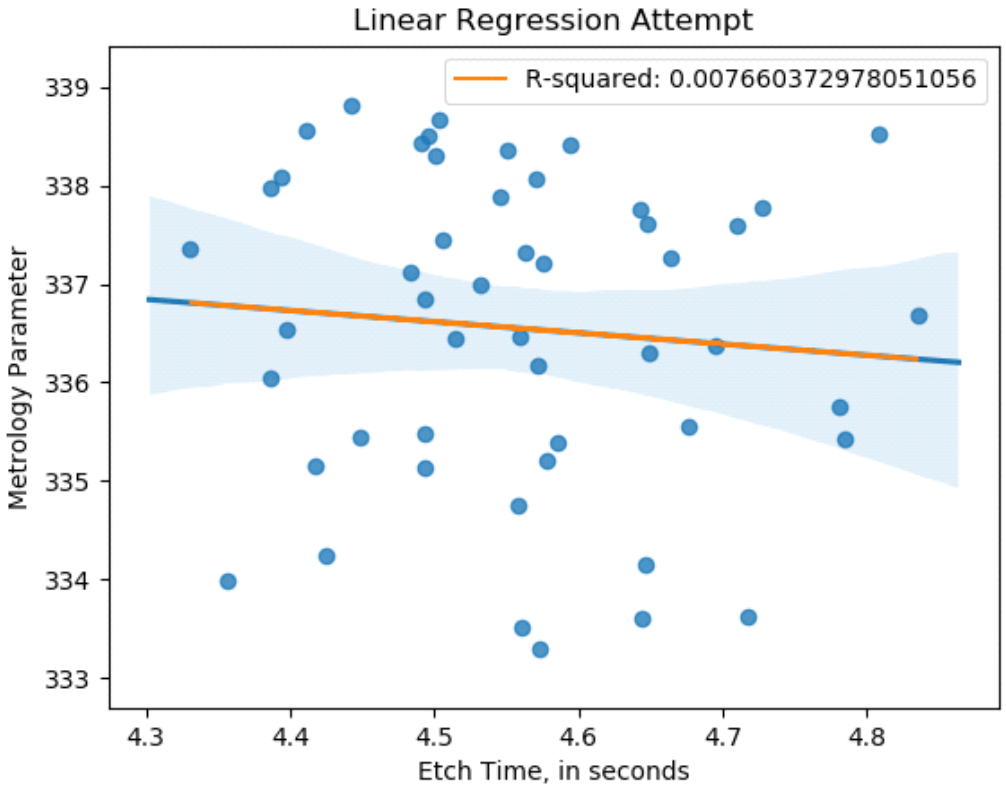
```
WinLen = 31
POrder = 10
```

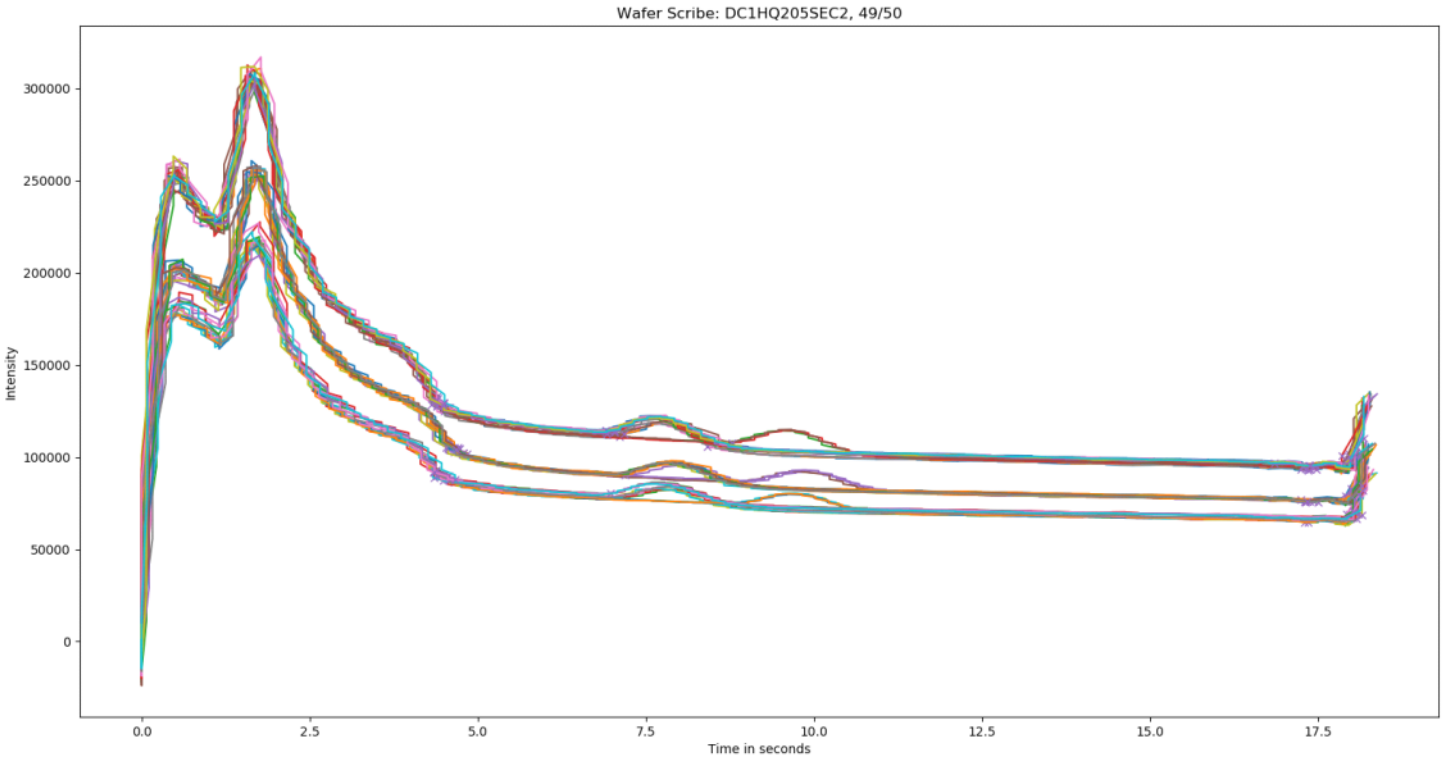


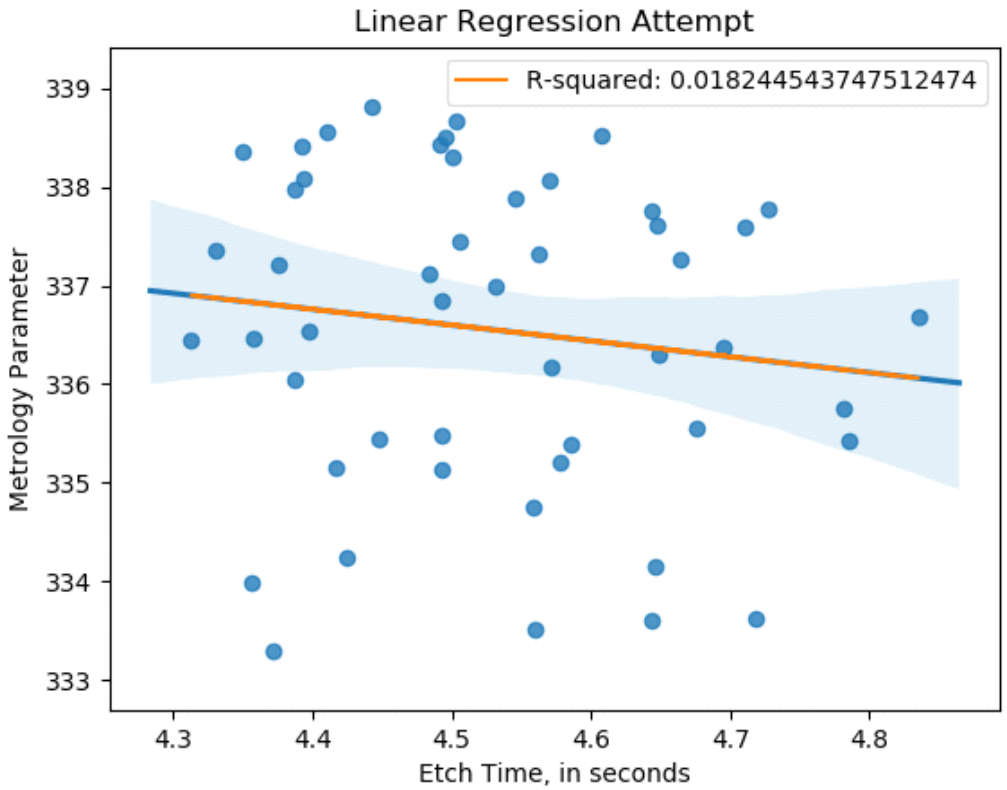

```
WinLen = 31
POrder = 10
```




Wafer Scribe: DC1HQ205SEC2, 49/50
Intensity
300000
250000
200000
150000
100000
50000
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Time in seconds


Linear Regression Attempt
R-squared: 0.007660372978051056
Metrology Parameter
339
338
337
336
335
334
333
4.3
4.4
4.5
4.6
4.7
4.8
Etch Time, in seconds





Let's call the window length M, and N is the order of the polynomial. One important thing to realize is that you use a polynomial of order N to approximate M data points. This means that you use M points to compute N+1 polynomial coefficients (an Nth order polynomial has N+1 coefficients). So if N+1=M then you do not smooth at all but you just use the polynomial to interpolate the data points. In your example this is almost the case, i.e. there is hardly any smoothing. Generally, N is chosen considerably smaller than M to achieve some smoothing (and also for numerical stability). The smaller N compared to M, the more smoothing you will get. What remains is the choice of M, but as far as I know there are no general guidelines. In practical examples I've seen polynomial orders of 4 or 5 with much larger values for M. With all this in mind it remains to try several combinations of M and N and choose the one that fits best your application.

From <https://dsp.stackexchange.com/questions/15643/savitzky-golay-filter-parameters>

HOLD EVERYTHING.
Jupyter notebook is RUNNING on my WINDOWS PC at MICRON. WAT. HOW, WHY, WHO. I'm very excited and will work on transitioning from Visual Studio Code to JupyterLab. Oh man, this is working so well now and taking so much less time. Frustrating that it took this long, but glad I'm here now.

I bet this has to do with the recent updates I've been doing here. Maybe. Hopefully it doesn't change.

Also, gotta start working on correlation stuff! Will be much easier to do it on a Notebook.l

1/28 JUPYTER

Tuesday, January 28, 2020 10:30 AM

Left-off:

> Work on transitioning from Visual Studio Code to JupyterLab. Also, gotta start working on correlation stuff! Will be much easier to do it on a Notebook.

Fixing the automatic browser that Jupyter opens. Hopefully will work once restart.

So far I've got SavGolSmoorth-elapsedtime and ProcessData to Jupyter. I've been working on getting interactive capabilities working, but no dice. Maybe things just need a restart as well? Unlikely, but can't hurt.
Oooh, but I can just drag a plot picture from Jupyter to the OneNote page.

Got to meet with Roy, another Metrology Intern working on WIS-based VM. Talking with him is nice, he's just getting started and I'll do my best to be a good resource for getting onboarded onto that project.

Email from Robert:

> I've been asked to look at VM options correlating 3010-44 METCON PHOTO or 5030-44 METCON INSITU DRY ETCH to 1210-44 METCON INSITU DRY STRIP 2 SEM CD and 1210-44 METCON INSITU DRY STRIP 2 SEM CD 2.
> The SLEDL flow is in flux and VM in this area would be highly valued. In particular, tying something to 1210-44 METCON INSITU DRY STRIP 2 SEM CD 2 (which is the BSE Bottom CD).
> Thoughts?

I'm down for any of this, just gimme some more data.

I think tomorrow I should go ahead and shoot an email to Dominic Seo and see what he thinks of my progress and specifically where I'm at in the Nitride step and see what he notices and where I could go.

1/29 Correlation Taster

Wednesday, January 29, 2020 9:30 AM

From Stan:

> I hope to have another process or two identified for you by the end of the week.
> SJ

Left-off:

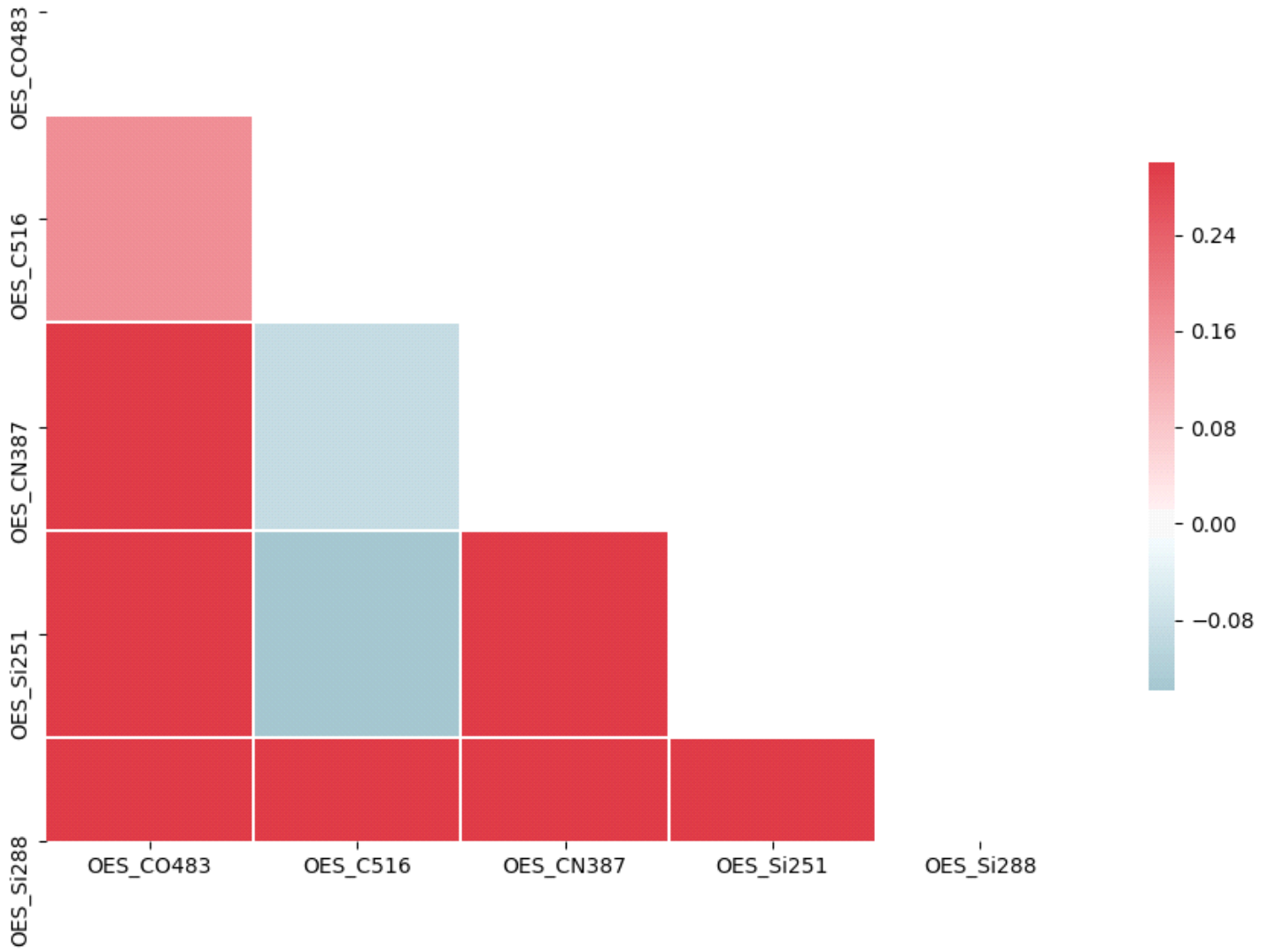
> I think tomorrow I should go ahead and shoot an email to Dominic Seo and see what he thinks of my progress and specifically where I'm at in the Nitride step and see what he notices and where I could go.

Questions I asked Dominic:

> What features in a process identify it as a good candidate for virtual metrology?
> Are there any processes that are **not** good candidates for VM, and why?
> Regarding this specific step & process, how did you identify the transition step to use?

Been running into weird Memory Allocation errors. Maybe this should be a longer conversation with Eric, if I keep running into this with no apparent solution yet.

Here's a cool correlation plot though probably not what Eric's talking about



	OES_CO483	OES_C516	OES_CN387	OES_Si251	OES_Si288
OES_CO483	1.000000	0.167694	0.853307	0.917994	0.813568
OES_C516	0.167694	1.000000	-0.086855	-0.138526	0.346425
OES_CN387	0.853307	-0.086855	1.000000	0.945755	0.663284
OES_Si251	0.917994	-0.138526	0.945755	1.000000	0.644521
OES_Si288	0.813568	0.346425	0.663284	0.644521	1.000000

2/4 Getting Caught Up

Tuesday, February 4, 2020 10:30 AM

Reply from Dominic:

What features in a process identify it as a good candidate for virtual metrology? Any material having THK data is working. Nitride working better normally by seeing clear transition in the OES.

Are there any processes that are **not** good candidates for VM, and why? If open ratio is small (ex. BEOL), OES signal is more noisy than normal.

Regarding this specific step & process, how did you identify the transition step to use? Once you are sure you are correlating same material for both OES and THK, normally end of the OES the time when final material is removed.

[Additional comments and concerns]

There is no step/materials that do not working as long as we are looking at right OES and THK.

Are you correlating the transition time of the OES of the material that measuring THK?

THK range is little narrow(336A +-3A) that makes good correlation.

Why over etch time is long after looking at the transition? Wonder 4.5 sec is the time when nitride is near completely gone? Need to check with Dry etch engineer.

Etch rate is different chamber to chamber and changing over time, especially after wet PM.

Reply to Stan & Zach! (Yay!):

Hey Zach, I just followed up with Adrian, Chris and Robert regarding retrieving the metro data and OES trace data for the process. Generally, I'm interested in the carbon etch process and which wavelength will be more closely related to the THK data that metro measures. I'll have more questions once I have the trace in front of me, but for now any general information on that process will be very helpful.

Recap from Eric:

Why am I filtering this data? Why am I using that correlation technique?

Let's get more informed.

Let's comment through our script explain what's going in (what data structure in what data type), coming out, why I "trim down" and what technique/method is being applied here? Where will it be used later? Justify it!

Essentially we have a 1-D array of OES data correlated to a scalar, float metro value.

How do we get to here?

We can zoom in on a couple of specific cases w/ easy to see info

Limit the cognitive load on 'how' step

Each step *could* be generalized… then we could get specific

Read! About relationships for 1D array to scalar-float

Artificial Neural Network

Matrix Multiplication

Mitchell, Mike, Eric Paper on ID-ing equilibration

$$[Model \quad \dots \quad m_n] \begin{bmatrix} OES \\ \vdots \lambda \\ o_n \end{bmatrix} = [Metro\ Scalar]$$

Where we can "train" the model, or have multiple layers

Orrrr… maybe $f\left(m_x, o_y\right) = [Scalar]$

What x or f(x) correspond to metro? Can we throw those away?

What's the reasoning to reach a solution?

The larger a model is the more training it requires

Need constraints:

At least 2 wafers

Need ability to probe between OES-Metro for collection

Correlates to a wavelength and a set of wafer metros

$\lambda_{1A}, \lambda_{1B}, \lambda_{1C} \rightarrow m_A, m_B, m_C$

So 22 STI is the Dry Strip Scatter with the parameter Depth_Total_Trench_Box1
Lotid: 3938223.003
Waferscribe: DA8LP017SEE3
Stepname: 1220-22 STI INTEGRATED DRY STRIP SCATTER
parameter_name: DEPTH_TOTAL_TRENCH_BOX01
sample_date: 2019-07-11 20:51:59
ext_mv: 336.307500000000004547473508864641189575195312 5

22STI has no THK, therefore, the model building with this one isn't correlated.

2/5 Getting Organized

Wednesday, February 5, 2020 10:30 AM

Starting with the Jupyter Notebooks

Not going to get into this now, but a cool Jupyter Widget could do WONDERS, mostly for me, but dang, these look super cool!

Monitor broke, slightly handicapped.

Went down a Memory Allocation rabbit hole in my stalling while waiting for a monitor fix

https://stackoverflow.com/questions/552744/how-do-i-profile-memory-usage-in-python
https://docs.python.org/3/library/tracemalloc.html