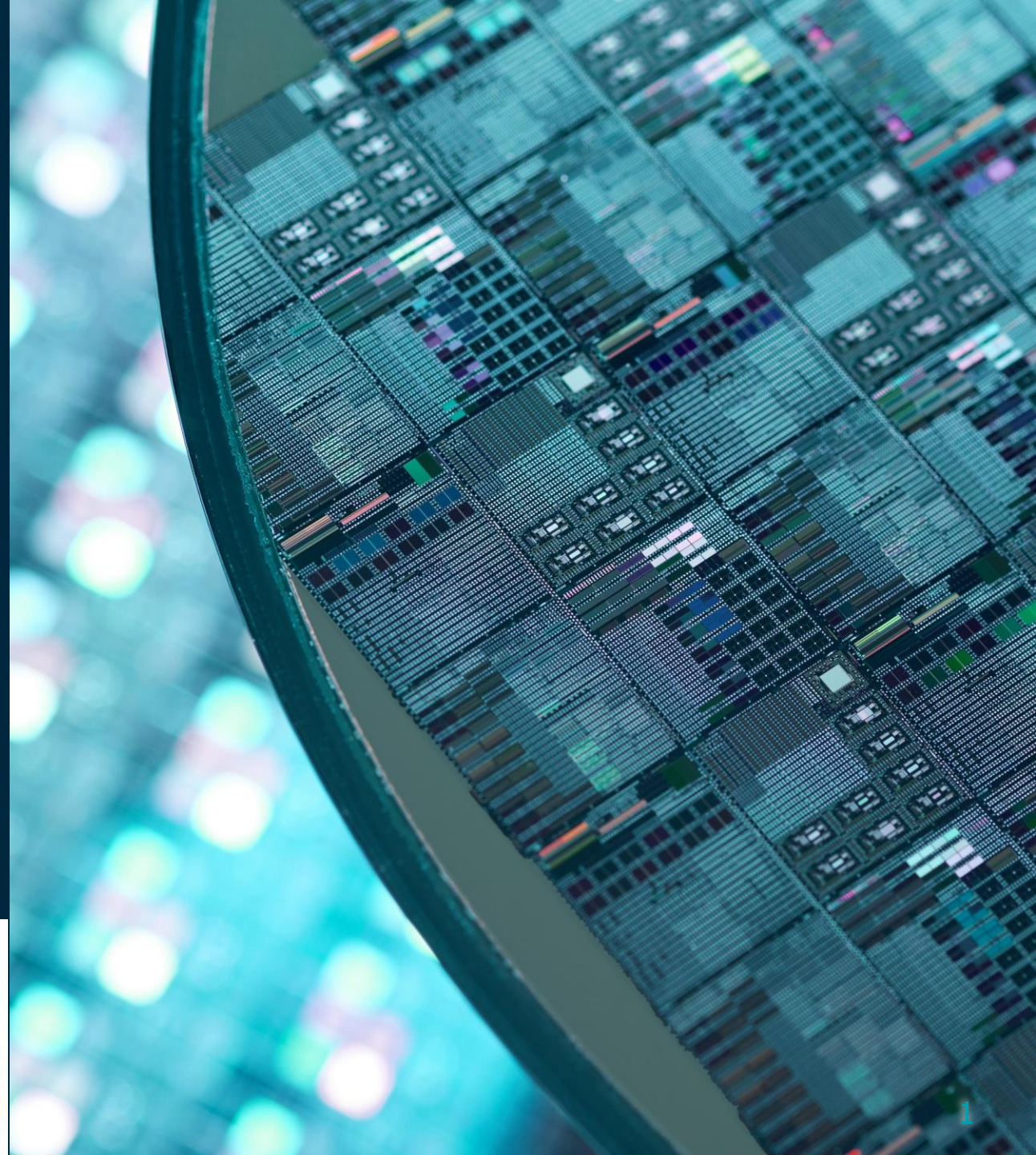




Dynamic height adjustment using Vector Network Analyzer based contact sensing using FormFactor WinCal XE 4.9™ and Velox 3.4 - THMA6

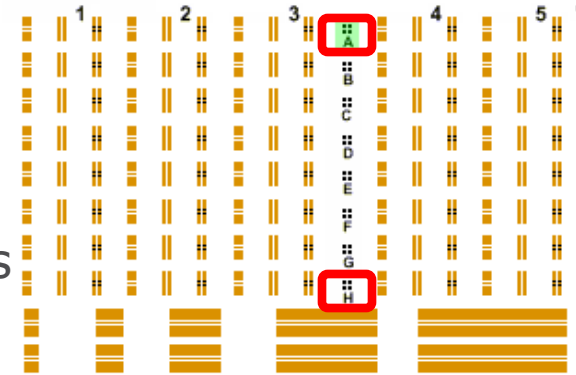
Gavin Fisher
James Hibbert
Thanks also to Pranav Shrivastava

The logo for the International Microwave Symposium (IMS), featuring two overlapping speech bubbles and the letters 'IMS' in a bold, black font.	<p>2023 IEEE MTT-S INTERNATIONAL MICROWAVE SYMPOSIUM</p>	<p>11-16 JUNE CONVENTION CENTER San Diego, California</p>	The logo for the San Diego 2023 event, featuring a sun icon, the text 'The Coolest Ideas Under the Sun', a vintage car with a surfboard on the roof, and 'SAN DIEGO 2023' in a stylized font.
The logo for the Institute of Electrical and Electronics Engineers (IEEE), featuring a diamond shape with a lightning bolt and the letters 'IEEE'.	The logo for the Microwave Theory and Technology Society (MTT-S), featuring a stylized 'M' and 'T' and the text 'IEEE MICROWAVE THEORY & TECHNOLOGY SOCIETY'.		

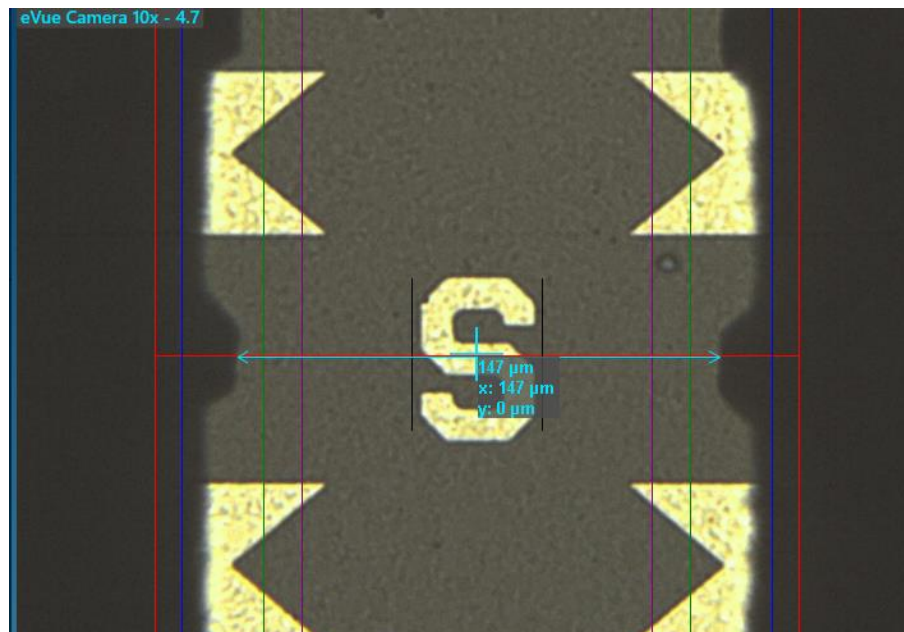


The problem – Contaminants can affect planarity

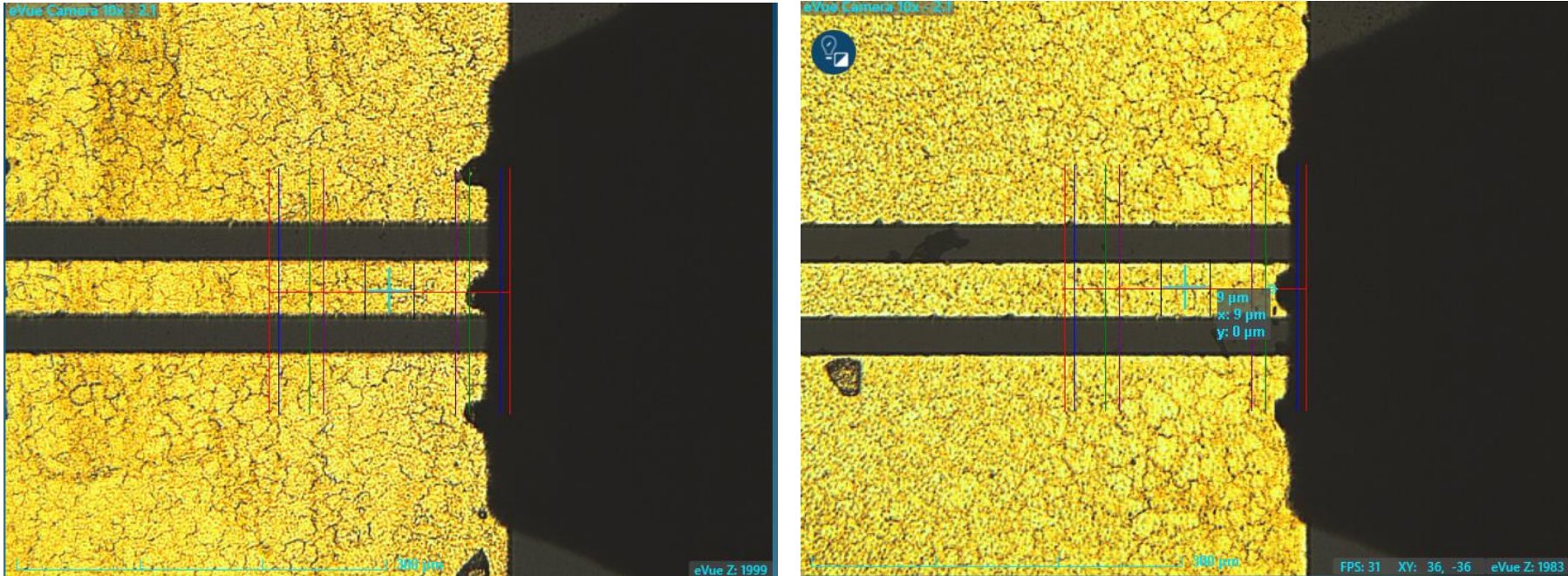
- In WinCal XE the probe geometry is set at a single reference location in terms of XYZ
- During calibration, the system steps using iss co-ordinates assuming the planarity is perfect
- Contaminants under the substrate can cause planarity to change, resulting in more or less overtravel
- Overtravel variation affects probe final position at the standards away from the reference



- Augmented alignment
Green lines set to be 130 μm – probe geometry set to this spacing at alignment Mark A
- Stage move to location H –
Less skate and probes now spaced to 140 μm



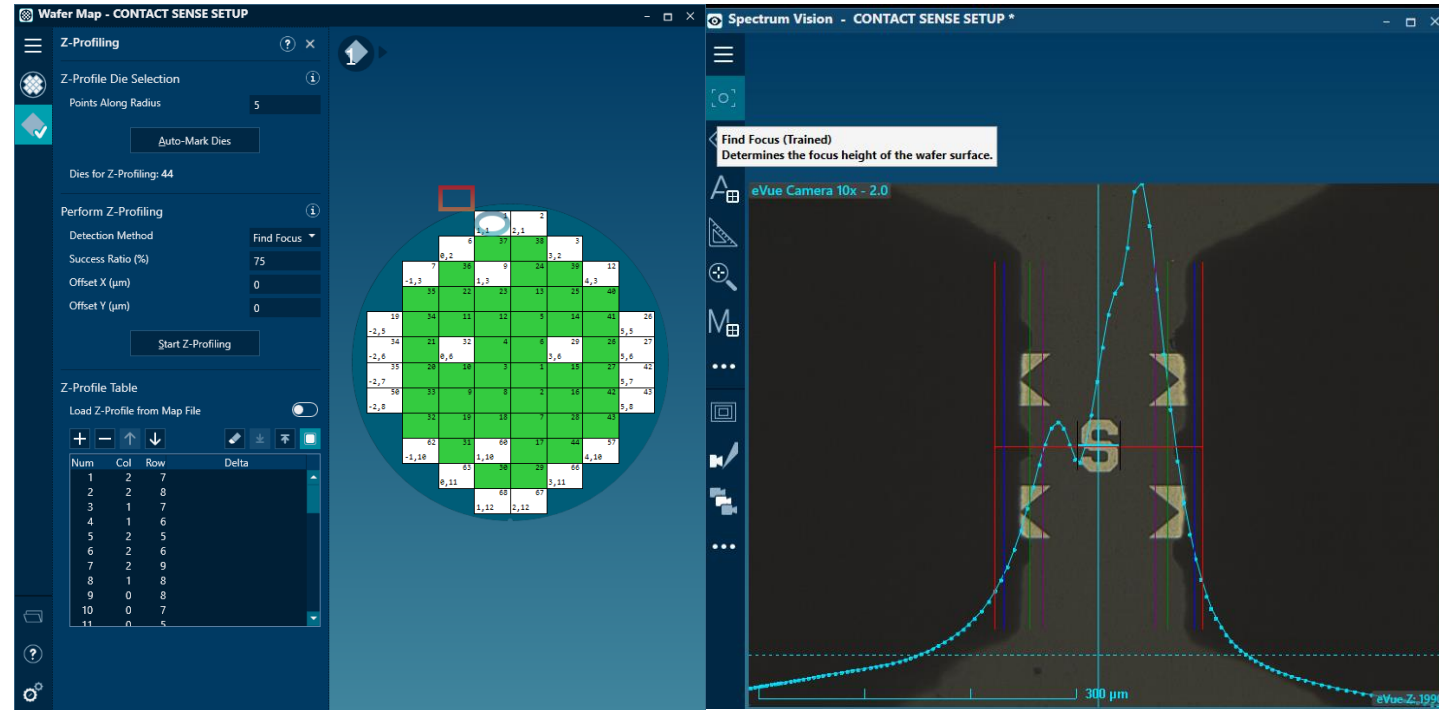
The problem – planarity of iss and positioner runout



- Final placement error seen here is 10 μm – A combination of small positioner planarity error and iss planarity error and small X offset
- Note height of scope Z reference was changed by 16 μm to get best focus

Alternative solutions to contact sensing?

- Velox supports Z profiles to the main wafer chuck but doesn't do this at the auxiliary chucks at present
- Lookup table for Z that could be applied at calibration time ,breaking into the automatic calibration routine but needs calculated initially
- Dynamic height adjustment could be done using the FindFocus algorithm and height adjusted this way but there is potential for failure with contaminants on iss itself.
- Find focus wouldn't detect probe changes as it stands without use of Vuetrack and this is more difficult to implement in calibration sequence

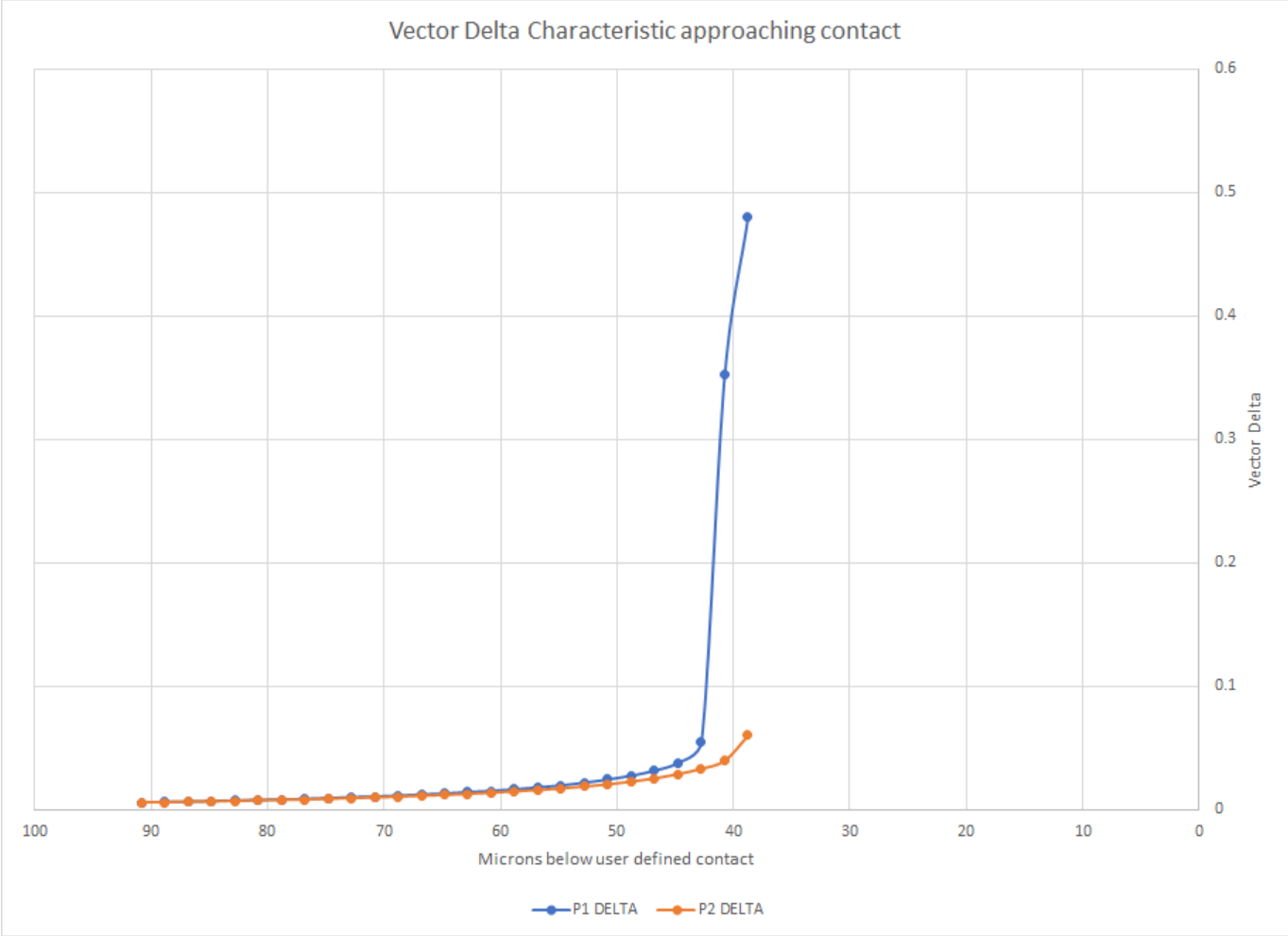


Contact sensing



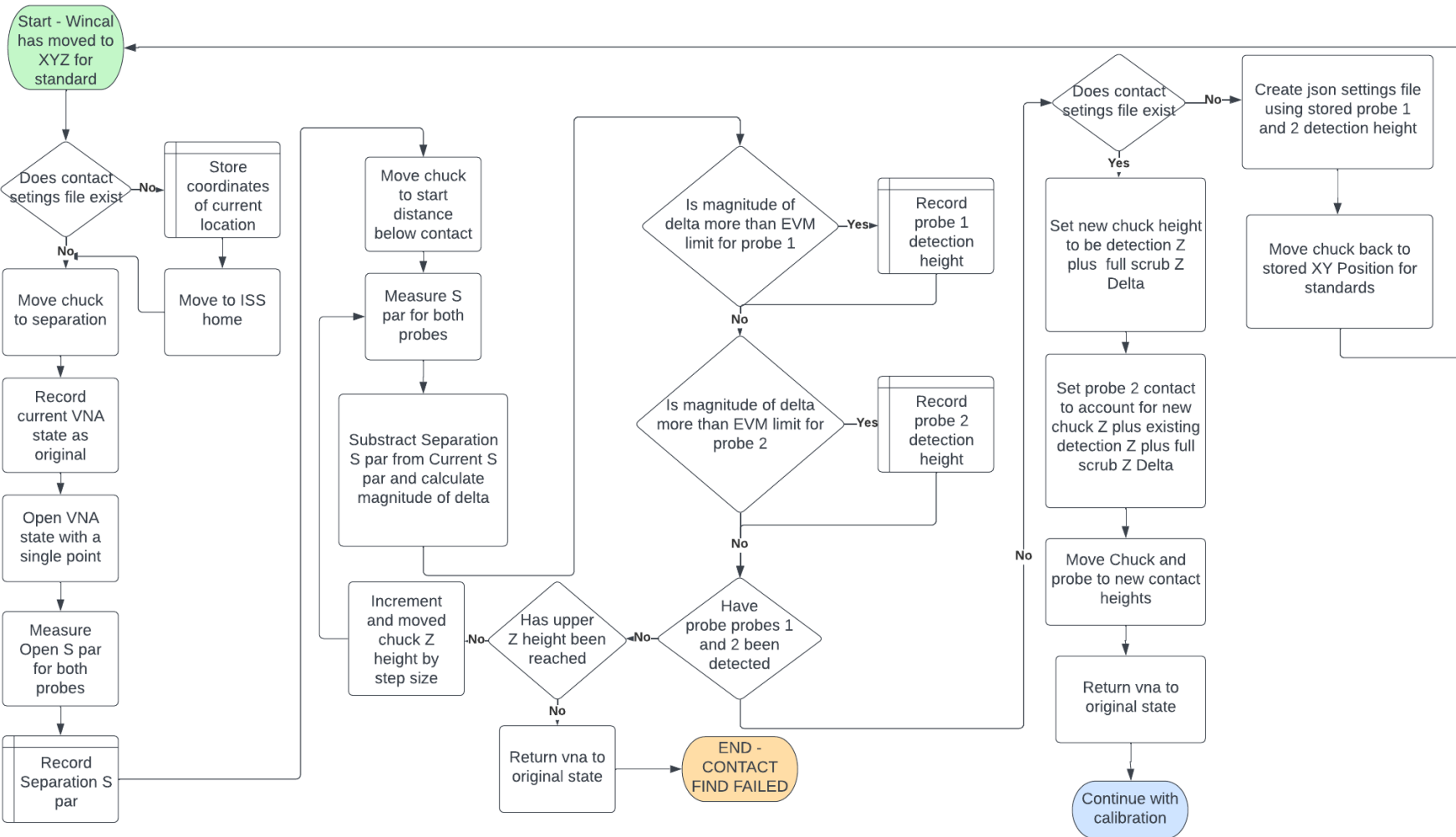
- Very repeatable and uses the measurement system itself
- Is dynamic and reflects the height of the probe at the actual time of calibration (as probes cool the height can change)
- Can be very quick when communicating directly with the vna via tcp (as we did)
- Drawback of direct approach is a driver is needed per instrument type additional to WinCal's own
- Can be compatible with Autonomous RF setups
- Is simple – probes need setup for the iss anyway....
- $\Delta \text{ Magnitude} = ((\text{Real_current} - \text{Real_Open})^2 + (\text{Imag_current} - \text{Imag_Open})^2)^{0.5}$

Contact detection Threshold



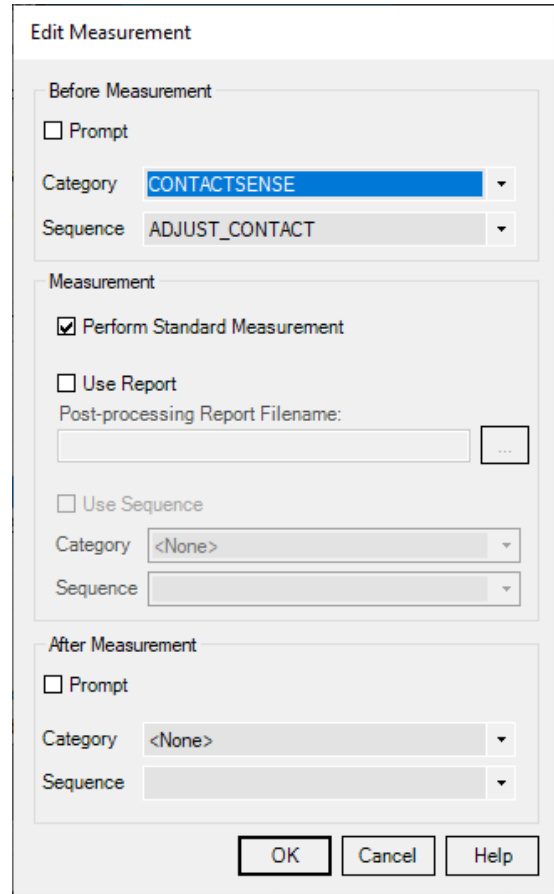
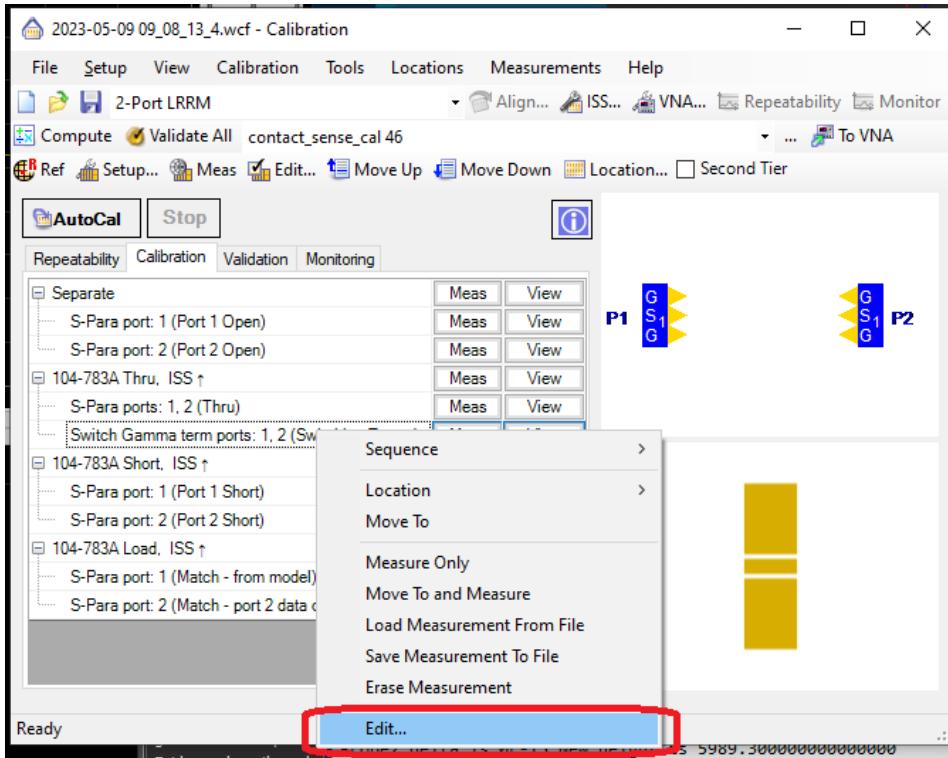
- Steps are in 2 um here
- Rate of change of Vector delta changes rapidly typically at 0.05 delta
- Here the stopping point is 0.05 – probes barely kissing pad

How does it work in general inside Python script?



This is a flowchart of the general contact sense python script logic. This script is run during the calibration process.

How to make python scripts work within WinCal



- Python direct approach could use WinCal as a slave, but preference is to sense in the normal calibration approach
- WinCal can invoke a “sequence” during the calibration as an activity carried out prior or post standard measurement
- Calibration sequence can in turn invoke a python script using DoScript command
- Each measurement can have a sequence run before and after and even use a specified report for process work

A few tricks regarding implementation

Edit Measurement

Before Measurement

Prompt

Category: CONTACTSENSE

Sequence: THRU_SNAP

Measurement

Perform Standard Measurement

Use Report

Post-processing Report Filename: []

Use Sequence

Category: <None>

Sequence: []

After Measurement

Prompt

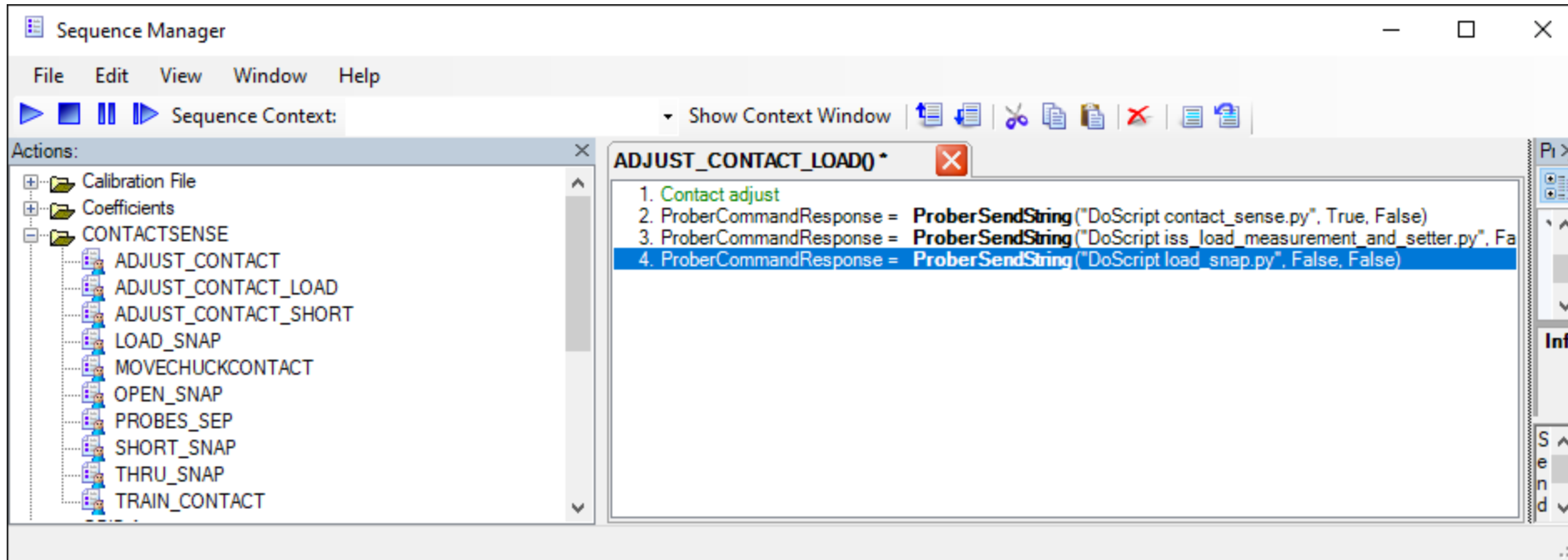
Category: CONTACTSENSE

Sequence: PROBES_SEP

OK Cancel Help

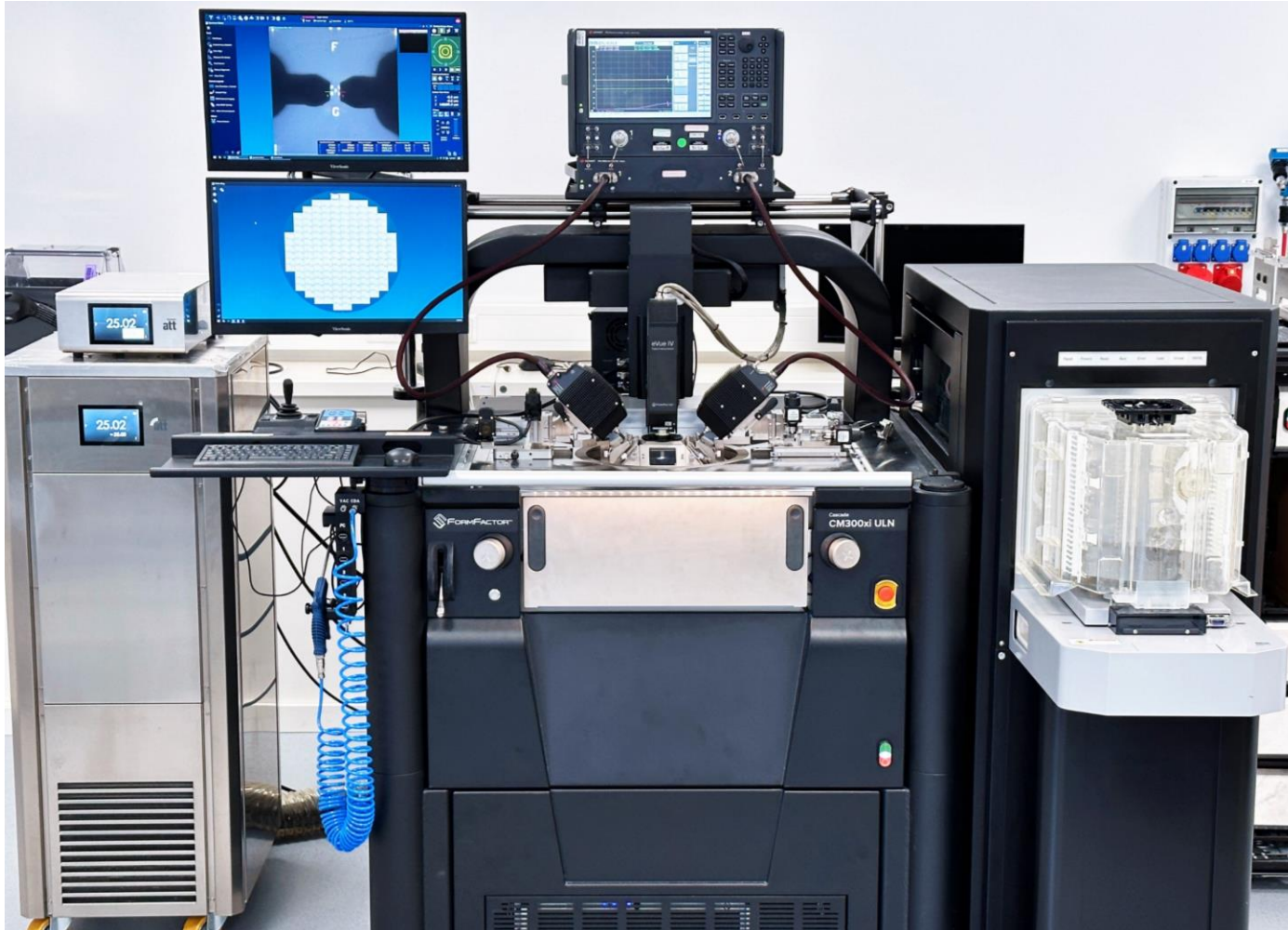
- By default, WinCal will perform all the movement activities prior to reaching the standard including movement to contact
- However, it will not bring positioners to contact if they were previously at separation before the movement
- Multiple touchdowns avoided to reduce pad wear
- For this reason, the ProbeSeperation sequence is done after the standard measurements
- In this case this is the second standard in the pair of reflect standards

Sequence manager being used to call Python scripts



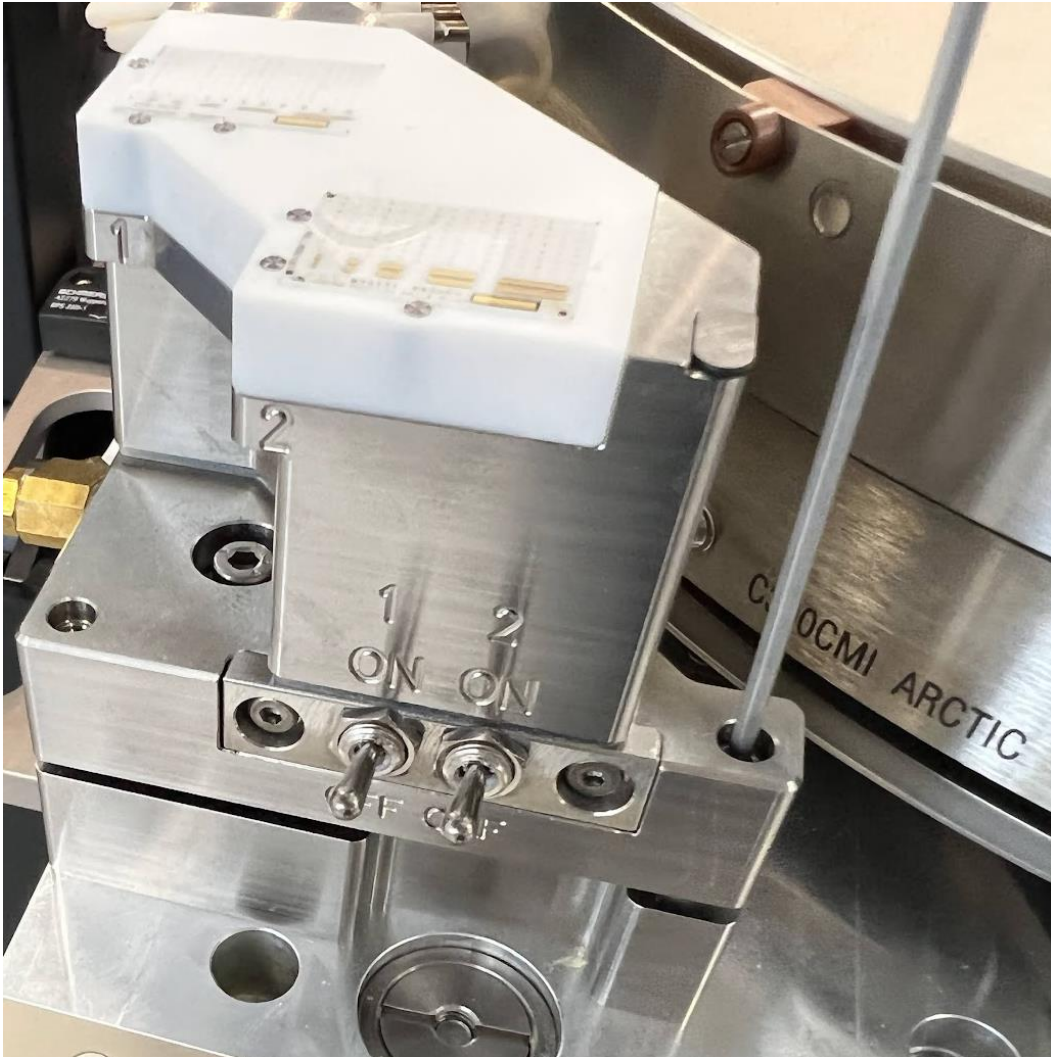
- The set of sequences is seen in the list

System used for evaluation



- WinCalXE™ 4.9
- Velox™ 3.4
- CM300xi ULN
- Keysight N5291A VNA
- I110-AM-GSG-100
- 104-783 ISS
- RPP504 Motorised positioners
- Remote Author was in UK
machine in Germany

Iss contamination simulation

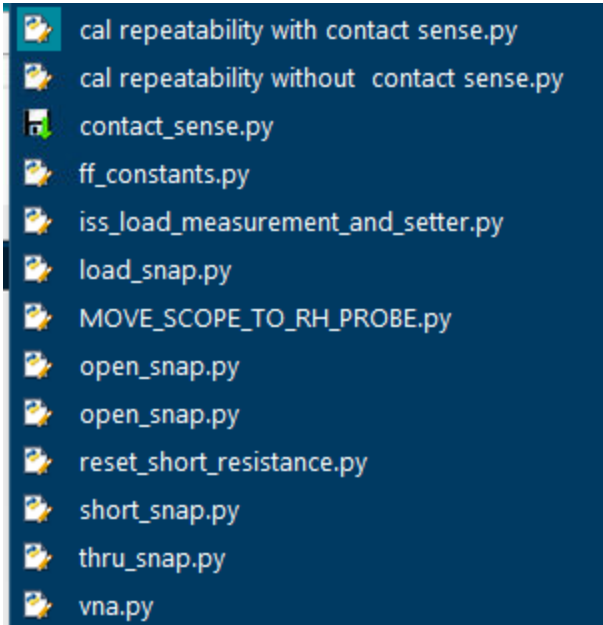


- ISS Chuck was deliberately mis-adjusted to simulate errors from contaminants or other sources
- This was done using mechanical adjustment of planarity screws

Testing methodology

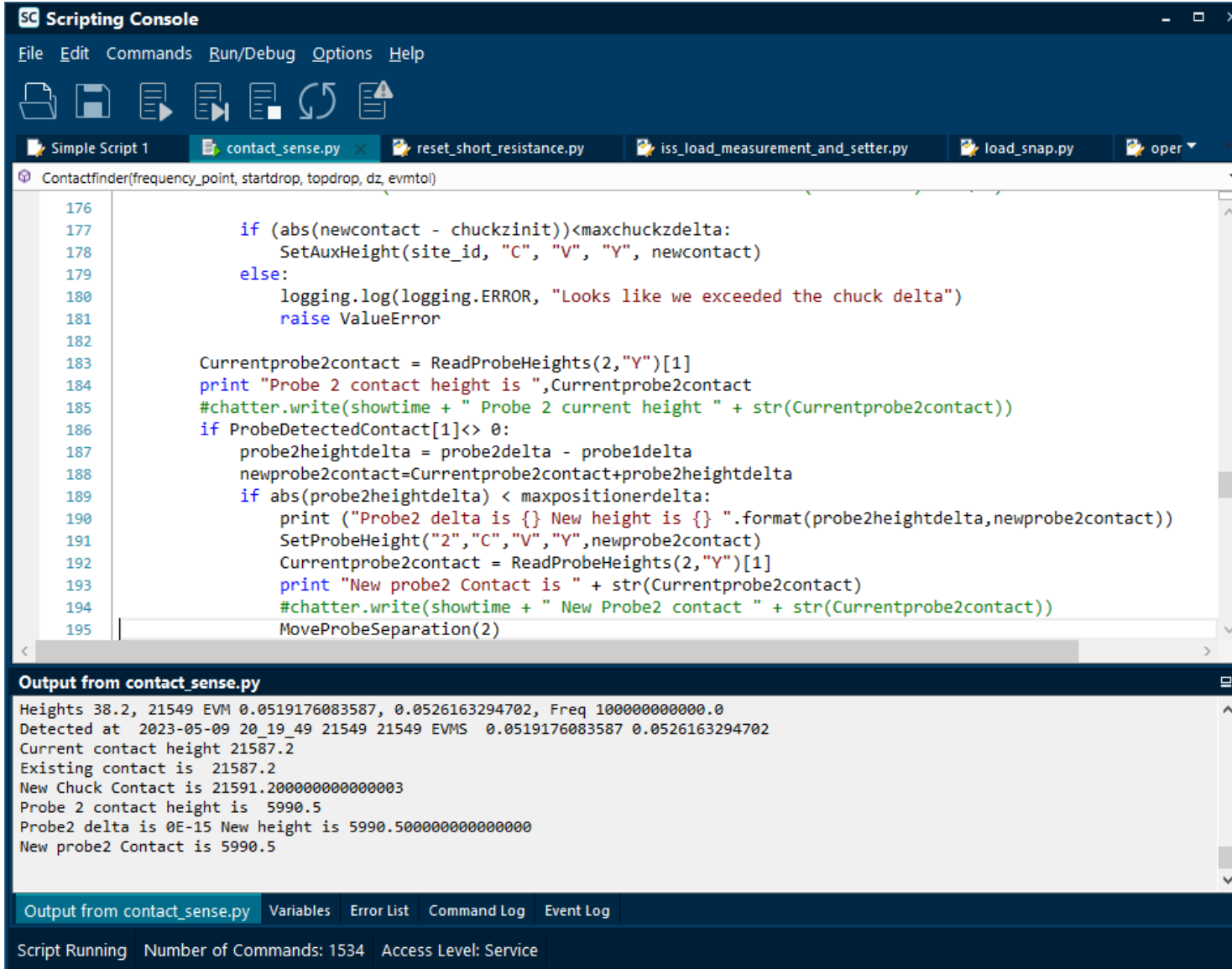
- Use a script to perform calibrations at 16 locations on 104-783
- Perform calibration set with load resistance compensation and contact correction
- Record error sets for all locations tested and photos of probe contact.
- Measure Open in air and same thru as used for the test
- Carry out calibrations on same iss and setup but only use load compensation and no contact correction
- Compare spread of error sets and Open measurements
- 25 um skate used
- Iss was a unit that had failed trim process but fine for our needs

Scripts used



- Main script is contact_sense, which either determines the required overdrive per probe set by the user from sensing or corrects the chuck and probe2 contact based on contact sensing of current standard
- VNA is crucial to communicate to the vna directly over socket
- FF_constants are the primary control variables
- Iss Load measurement and setter measure the short resistance and if on a load measures and compensates the load and sends to WinCal XE
- Cal_repeatability scripts are used to run the tests
- Reset_Short_Resistance removes the short data file to force measurement of a short rather than a load

Scripting console



The screenshot shows the Scripting Console interface with a menu bar (File, Edit, Commands, Run/Debug, Options, Help) and a toolbar with icons for file operations and execution. The main area displays a Python script with line numbers 176 to 195. The script defines a function `Contactfinder` and contains logic for handling probe contact heights, including error logging and state updates. Below the code, the 'Output from contact_sense.py' panel shows the execution results, including detected contact heights and probe delta values.

```
176 Contactfinder(frequency_point, startdrop, topdrop, dz, evmtol)
177     if (abs(newcontact - chuckzinit))<maxchuckzdelta:
178         SetAuxHeight(site_id, "C", "V", "Y", newcontact)
179     else:
180         logging.log(logging.ERROR, "Looks like we exceeded the chuck delta")
181         raise ValueError
182
183     Currentprobe2contact = ReadProbeHeights(2,"Y")[1]
184     print "Probe 2 contact height is ",Currentprobe2contact
185     #chatter.write(showtime + " Probe 2 current height " + str(Currentprobe2contact))
186     if ProbeDetectedContact[1]<> 0:
187         probe2heightdelta = probe2delta - probe1delta
188         newprobe2contact=Currentprobe2contact+probe2heightdelta
189         if abs(probe2heightdelta) < maxpositionerdelta:
190             print ("Probe2 delta is {} New height is {}".format(probe2heightdelta,newprobe2contact))
191             SetProbeHeight("2","C","V","Y",newprobe2contact)
192             Currentprobe2contact = ReadProbeHeights(2,"Y")[1]
193             print "New probe2 Contact is " + str(Currentprobe2contact)
194             #chatter.write(showtime + " New Probe2 contact " + str(Currentprobe2contact))
195             MoveProbeSeparation(2)
```

Output from contact_sense.py

```
Heights 38.2, 21549 EVM 0.0519176083587, 0.0526163294702, Freq 10000000000.0
Detected at 2023-05-09 20_19_49 21549 21549 EVMS 0.0519176083587 0.0526163294702
Current contact height 21587.2
Existing contact is 21587.2
New Chuck Contact is 21591.20000000000000003
Probe 2 contact height is 5990.5
Probe2 delta is 0E-15 New height is 5990.5000000000000000
New probe2 Contact is 5990.5
```

Output from contact_sense.py | Variables | Error List | Command Log | Event Log

Script Running | Number of Commands: 1534 | Access Level: Service

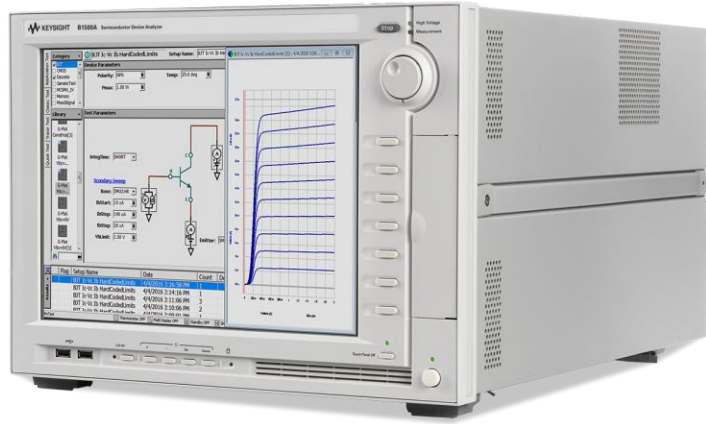
- The python scripts used were run via the scripting console but other approaches could also have been used
- Console is advantageous that the command choose and Intellisense for all the Velox commands makes coding fairly straightforward

Adjusting the Calibration group for measurement

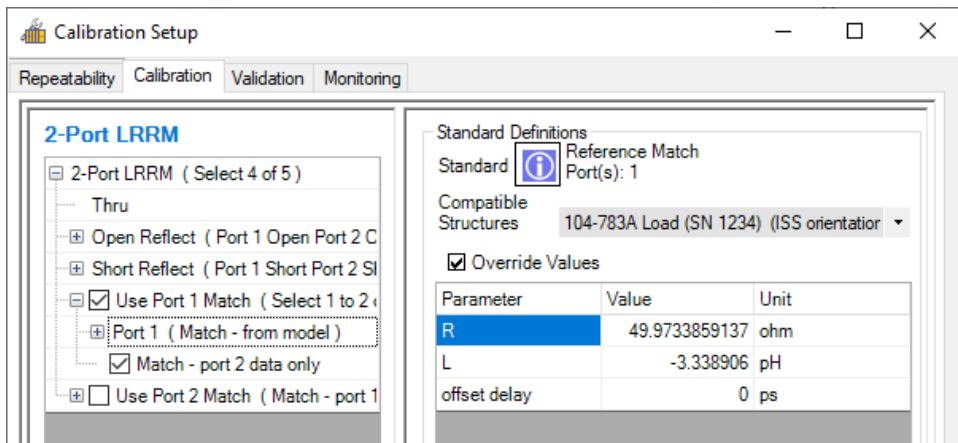


- During test procedure we automatically index the cal groups using following command `w.CalUseNextGoodGroupOnIss()`
- Groups shown were used
- We deliberately ran a run through all locations prior to capture of data – experience shows significant variation from very first contact to the next

Load resistance measurement and adjustment on the fly

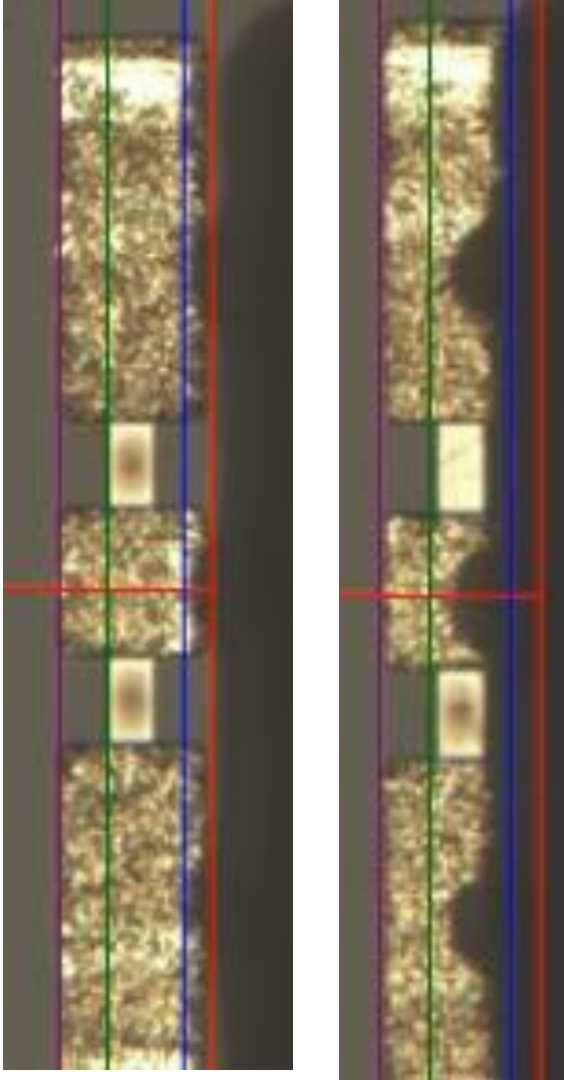


- ISS Map was not readily available so we needed to measure the load resistance – we used B1500 via N5291A instrument bias tees
- The ability to add sequences to calibration process made direct measurement and compensation appealing
- Short was measured at .1 volts and used as offset reference and written to file
- Load was measured a 1 volt bias and resistance of short subtracted
- Corrected load value is automatically measured and applied during calibration process
- `WinCalExecuteCommand("CalSetCoefficient, 1, Match - from model, Reference Match, R, {}".format(rload1_corrected))`
- One resistor appears to have failed in 5R location..



	1L	1R	2L	2R	4L	4R	5L	5R
A	50.12321	50.00223	48.33143	48.24285	47.79855	50.57794	47.00259	47.02301
D	50.20838	49.94828	50.14571	49.90843	50.19265	50.05771	50.2203	49.96821
E	50.24733	50.01891	50.12752	49.90194	50.20219	50.01985	50.19329	49.96933
H	50.28777	50.04244	50.12788	49.95298	50.25752	50.08616	50.1595	100.0728

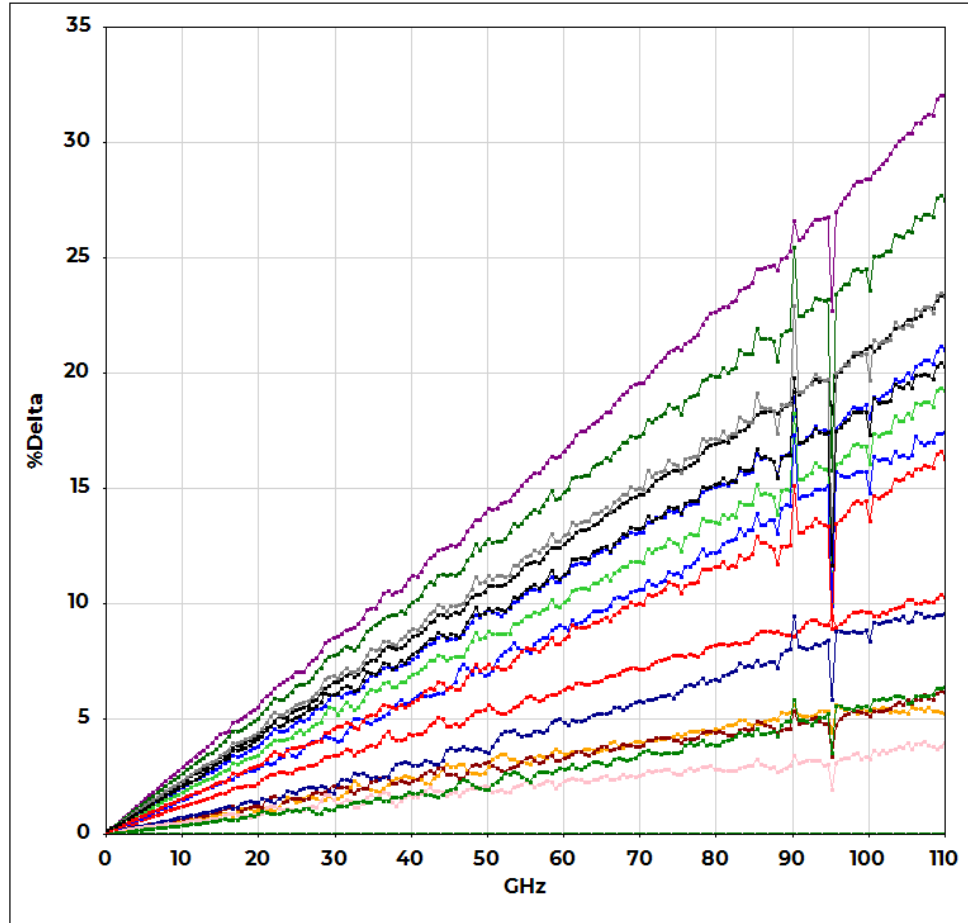
Resistor 5R Mistrim



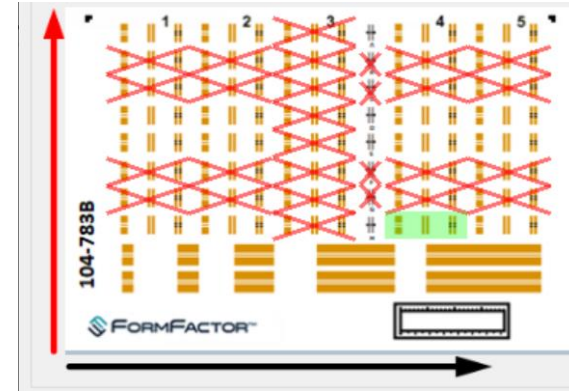
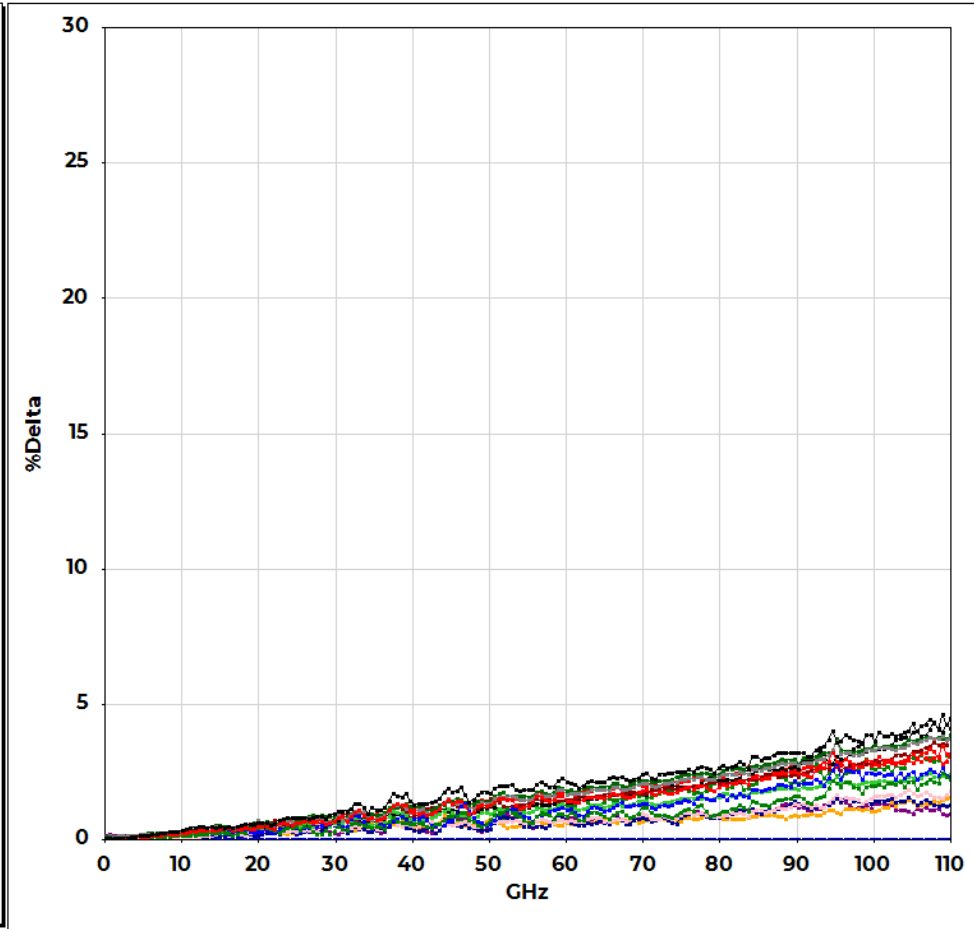
- This iss was not of saleable quality and was scrapped
- Location 5R was detected as 100 Ohms and it can be seen that the resistors are burnt asymmetrically indicating it is suspected an open circuit to upper resistor
- Resistor on the left is in normal trim state

Error set comparison – Left graph without sensing

No Z Sense but Load resistance compensated

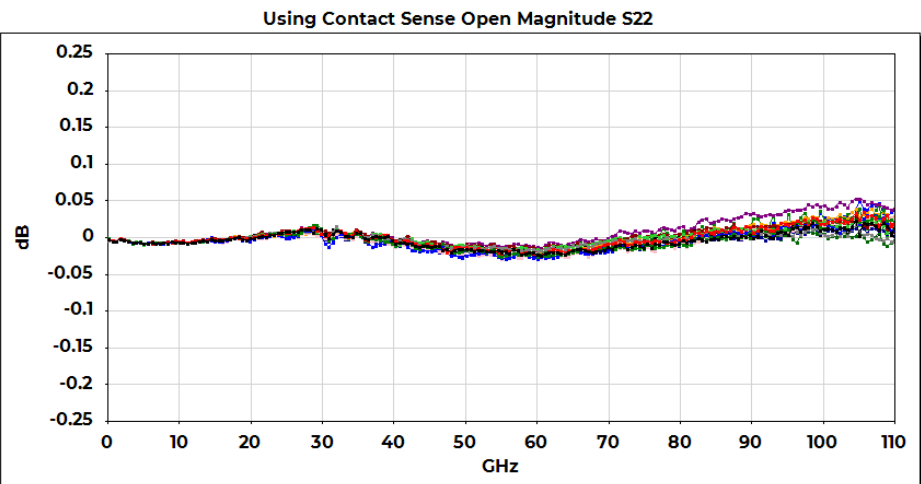
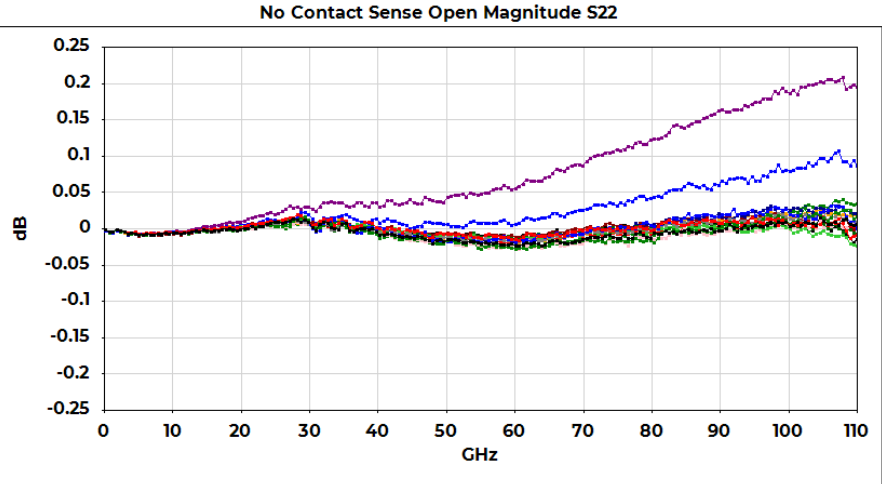
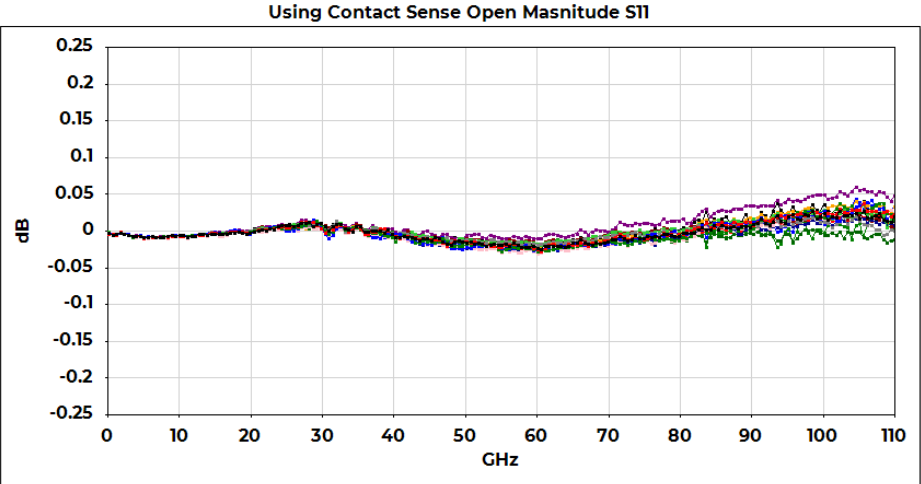
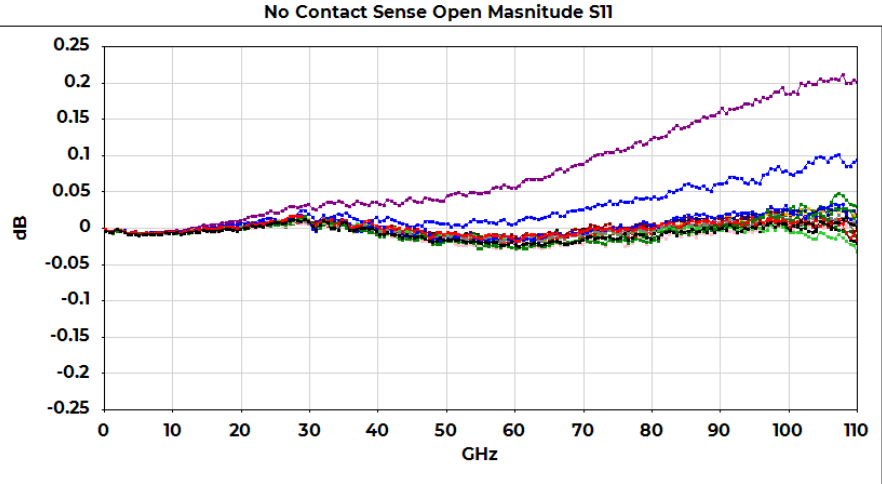


With Z Sense and Load resistance compensated



- Much tighter grouping when contact sensing is use
- All locations are the same
- Source of 95 GHz glitch during unsensed run not known

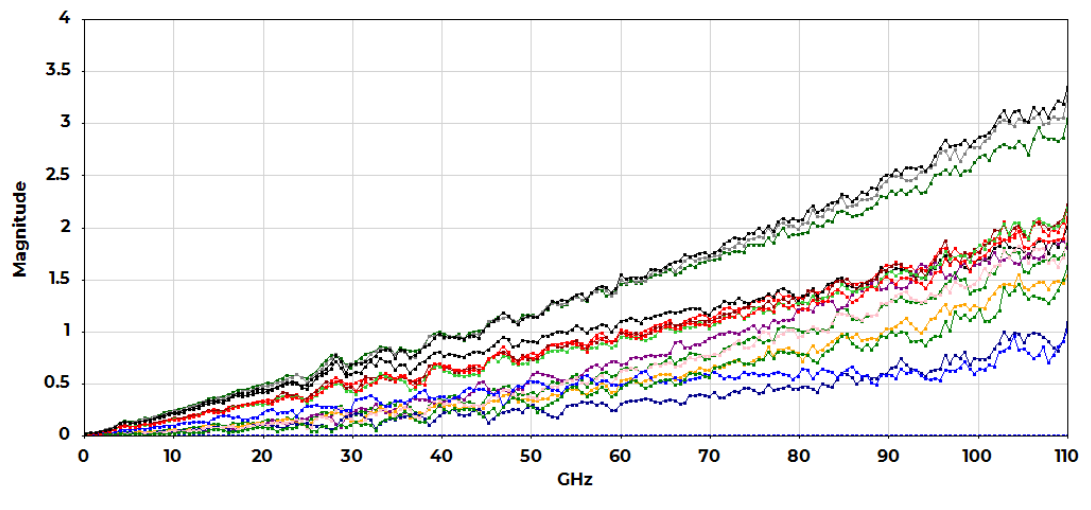
Post calibration Open Magnitude Variation



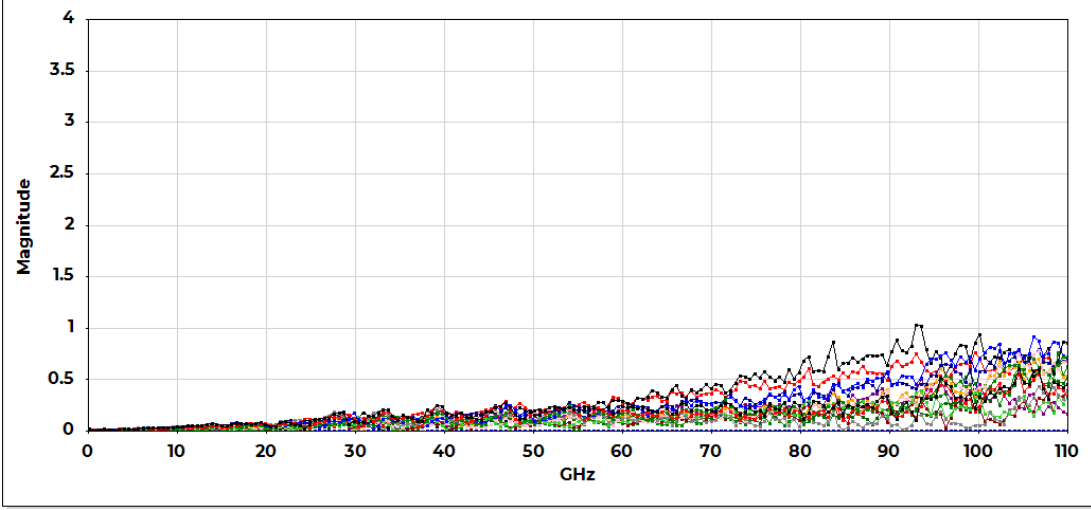
- Magnitudes fairly similar apart from outliers which may have had limited contact

Post calibration Open EVM Delta (x100) – This shows effects or varying capacitance

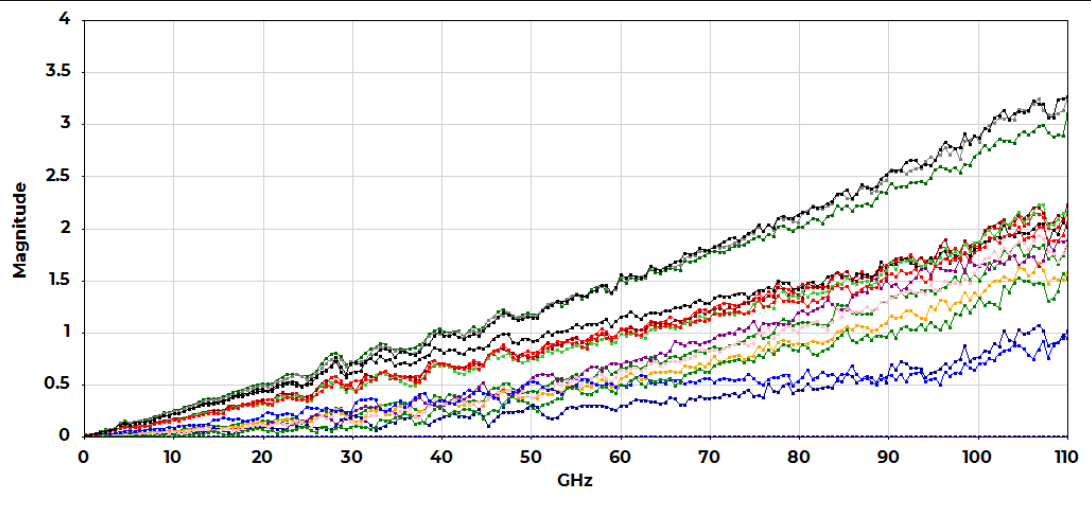
Regular Open EVM Port1



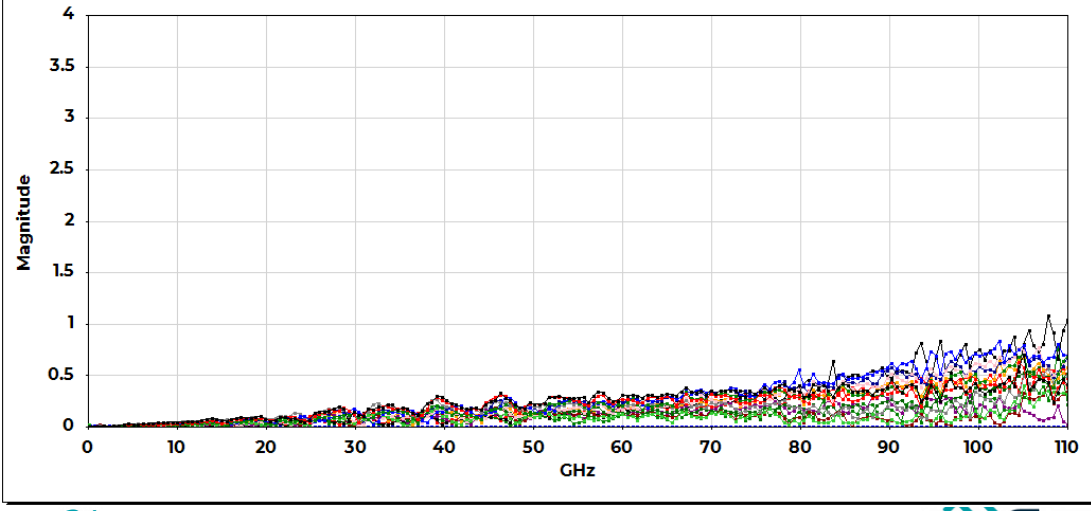
Z Sense Open EVM Port1



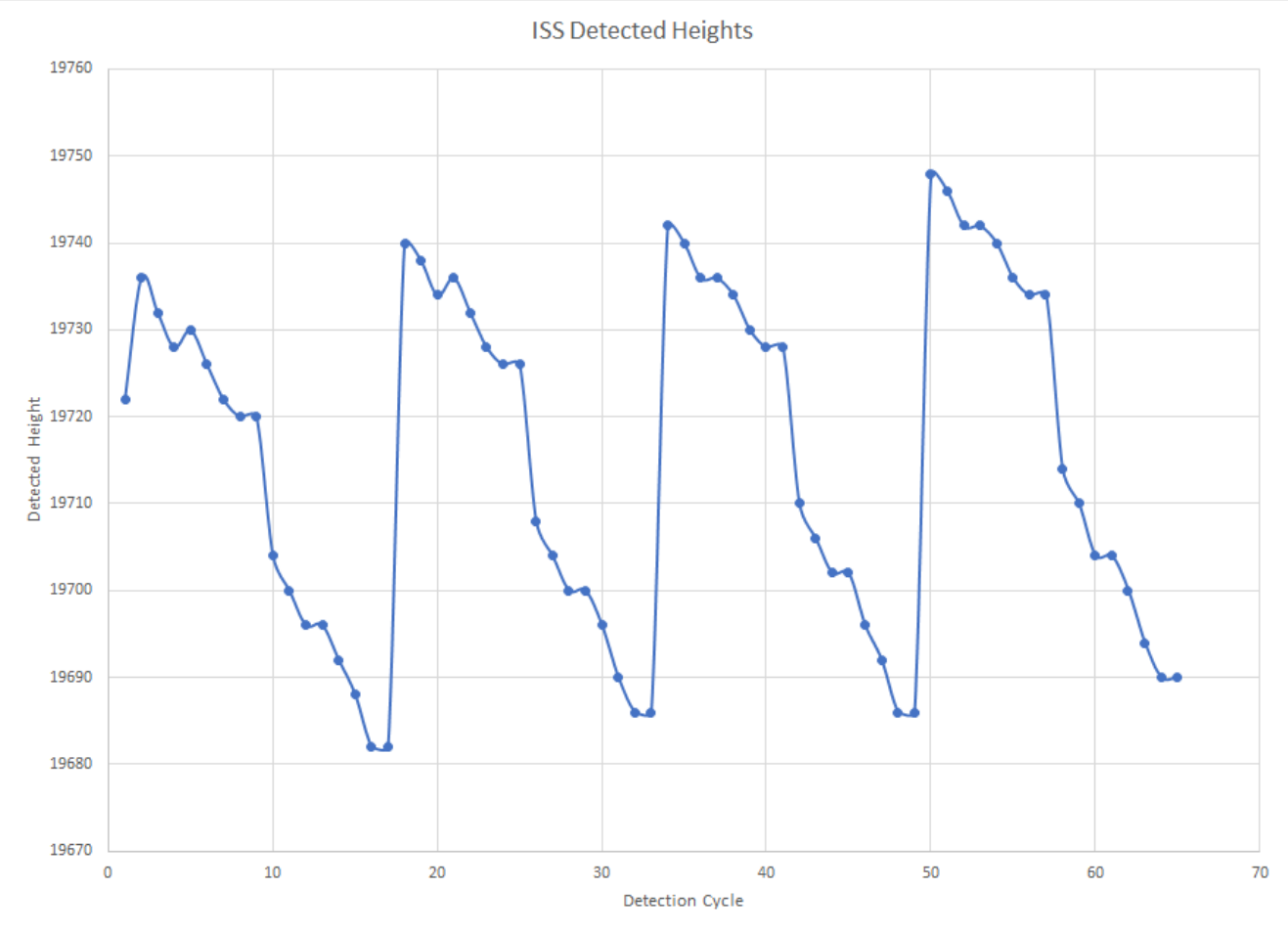
Regular Open EVM Port2



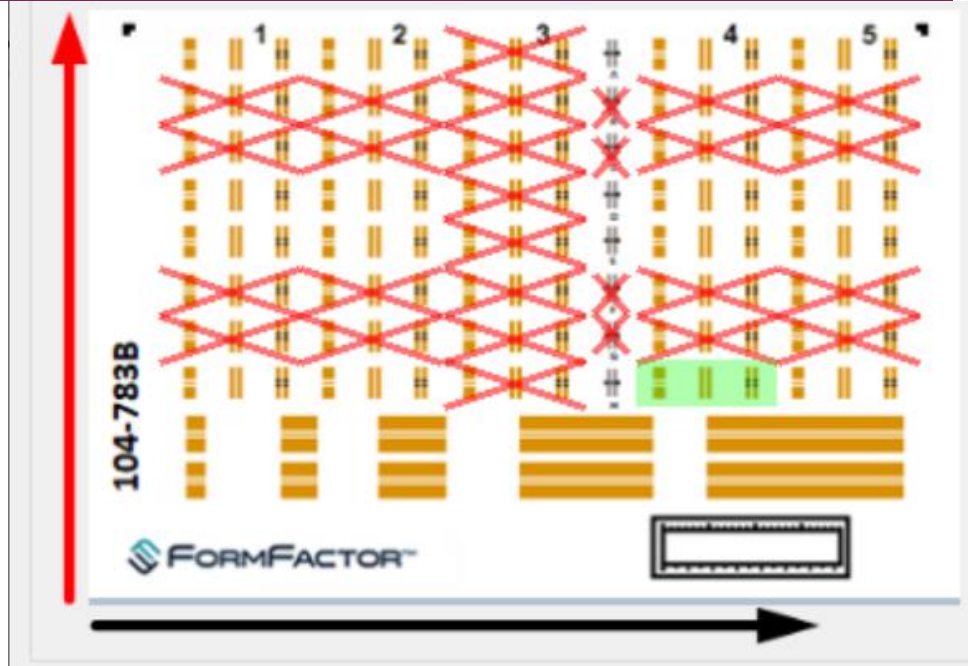
Z Sense Open EVM Port2



Contact height deltas across iss – Load to left of R-H is zero

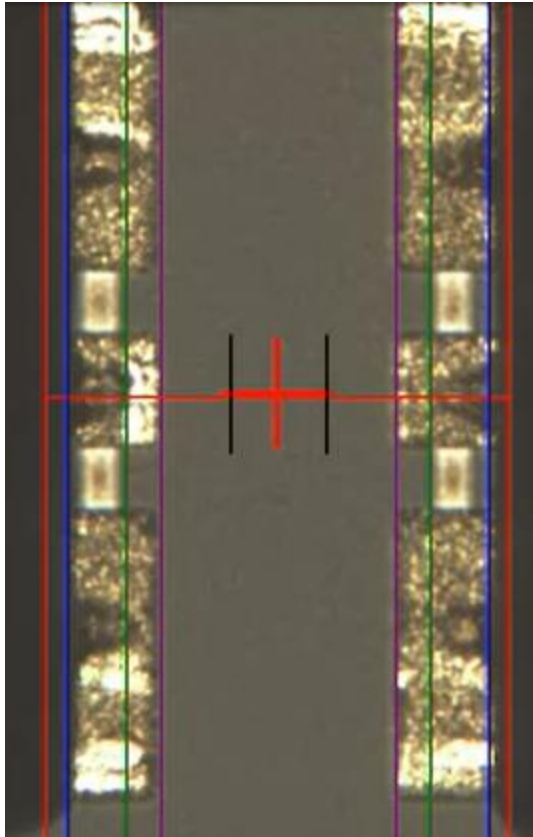


	1	2	3	R	4	5
A	14	4			-18	-30
B						
C						
D	18	10			-14	-26
E	20	12			-12	-26
F						
G						
H	26	18			-8	-22

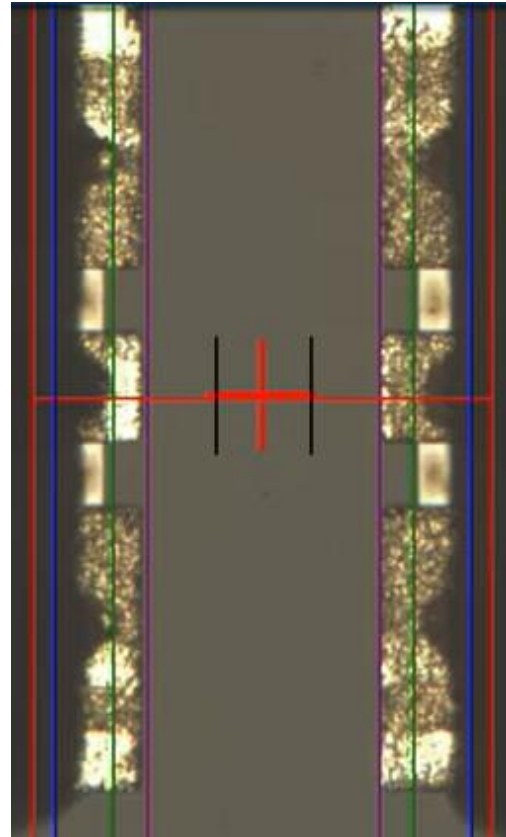


Comparing standard in the same location with and without sensing

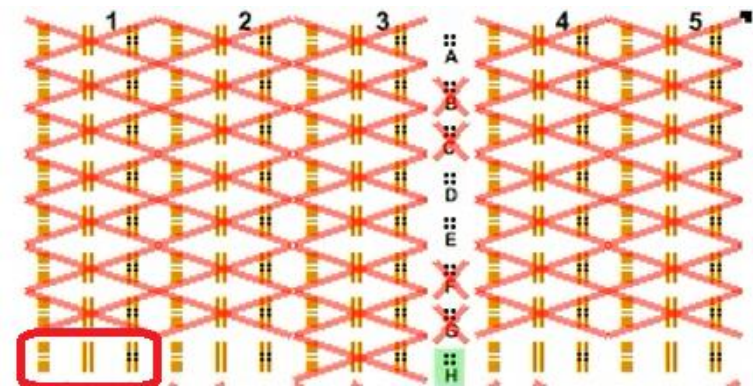
WITHOUT SENSE



USING SENSE



- Note relationship of dark leading edge of probe wrt blue lines (170 um apart)
- Tips clearly retracted from the green 130 um lines



Video of operation

The screenshot displays a complex software interface for a calibration system. At the top, a status bar shows 'Login: Service C:\data\RM224\RM224.spp', 'Chuck', 'Jog', 'Contact', and a temperature of '25.0 °C'. A 'STOP' button is visible in the top right corner.

The main interface is divided into several panels:

- basecal.wcf - Calibration:** A window with a menu bar (File, Setup, View, Calibration, Tools, Locations, Measurements, Help) and a toolbar. It contains a table for 'Repeatability Calibration Validation Monitoring' with columns for 'Separate', 'Meas', and 'Empty'. The table lists various calibration items like 'S-Para port: 1 (Port 1 Open)', '104-783A Thru, ISS', and '104-783A Short, ISS'. A 'P1' and 'P2' label is visible on the right side of the window.
- Spectru:** A window showing coordinate data for 'Chuck (Site 4) From Zero', 'Scope From Home', 'Positioner 1 From Zero', and 'Positioner 2 From Zero'. The data is presented in a table format with X, Y, Z coordinates and an angle.
- Control Center: Chuck:** A panel with 'Material Handling' and 'Predefined Positions' sections. It includes a 'Position From Home' section with X, Y, Z coordinates and a 'Z Setup' section with 'Contact' parameters (A: 100.0, S: 500.0, C: 19767.8).
- Scripting Console:** A window with a menu bar (File, Edit, Commands, Run/Debug, Options, Help) and a script editor. It shows a Python script for 'cal repeatability with contact sense.py' and an output window.
- TightVNC Viewer:** A window showing a graph with a y-axis ranging from -50 to 50 and a plot area.
- Joystick:** A panel with a 'CAL' button and directional controls.

Conclusion

- Contact Z adjustment via RF electrical sensing is a readily achievable enhancement to FormFactor probes equipped with WinCal XE 4.9 using existing software with the addition of some scripting and sequences
- Using a Z sensing approach, the error set variation is greatly improved
- Dynamic measurement of load resistance and on the fly compensation can be performed for customers who also have a DC measurement instrument
- Although we used motorised positioners also the approach could simply adjust contact using chuck variation. Motorised positioner addition useful to compensate potential positioner planarity variation also