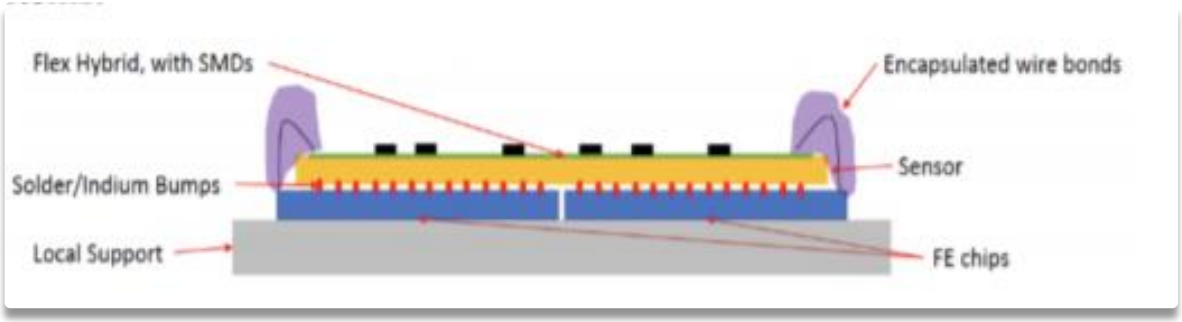
Placement of ITk Pixel Modules

PRESENTATION BY PATRICK BATES

SUPERVISOR: JOHN MATHESON

Introduction –Atlas ITk upgrade

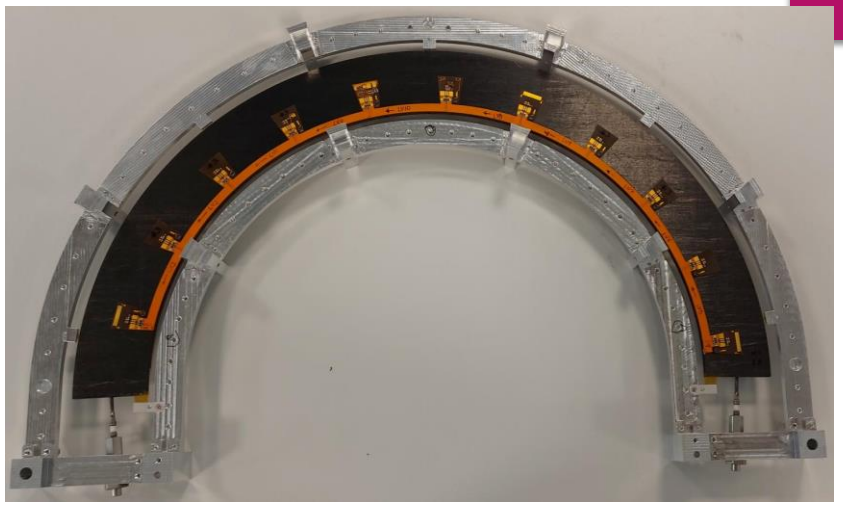


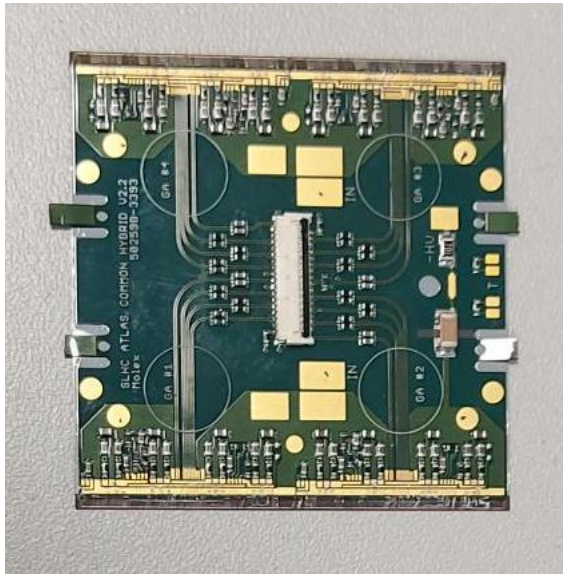
ATL-ITK-PROC-2020-002

- Atlas is currently undergoing an upgrade to replace the old tracking system with a new, all silicon, Inner Tracker (ITk)
- In the ITk there will be pixel endcaps, composed of pixel modules mounted on carbon fibre half rings
- The new Atlas ITk pixel modules consist of front-end chips bump bonded to a sensor, with a copper-Kapton hybrid glued to the sensor and then wire bonded to the front – end chips
- The pixel modules have dimensions of 4.0 x 4.1cm ^{2}, with 153,600 50×50μm^2 pixels

Mounting of the pixel modules

- The pixel modules are mounted on carbon fibre half rings, which are low mass, so should minimally affect the flight path of particles
- The modules are mounted to the half ring via a thermally-conducting glue
- Some modules are being mounted in the UK and others are being mounted in Italy so to share the work load
- Accurate placement of the modules is important as to achieve hermeticity and sufficient cooling





My Goals

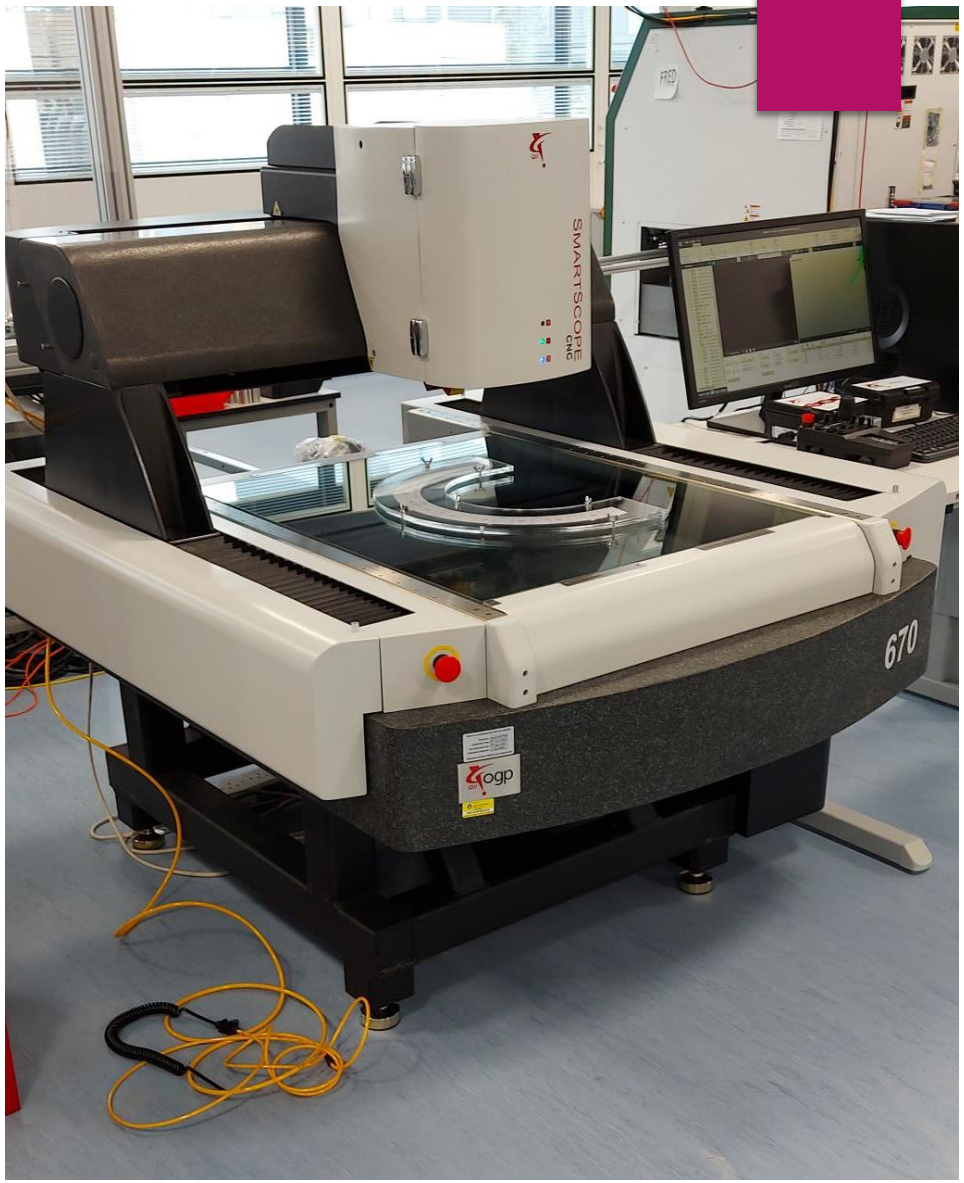
- Metrology – measure the positions and heights of the modules mounted on the half rings in an accurate and efficient manner
- Develop code that compares the positions of modules mounted by different groups, regardless of the coordinate system use

Metrology of the modules

- One of the tasks is to accurately, and efficiently, determine the position of modules after they are mounted, to check they are with in tolerance
- Locating the module corners is an easy method to see how well the modules have been placed
- Calculating the height of the module means the glue thickness can be determined
- For these measurements the smart scope will be used

Smart scope

- The smart scope is an Optical Gaging Products (OGP) system
- A smart scope is a combination of optical and laser measurement that can be automated to carry out multiple measurements quickly and with high precision
- The aim is to write a program to carry out the measurements of the pixel module position and heights on the half ring
- I was introduced to the smart scope's basic operations by Emma Buchanan, who offered help when I asked
- Initial testing was required as to familiarize myself with how it works and gauge its capabilities
- Various methods were tested on how to locate certain features and take different measurements with a dummy module, i.e., the module corners and the height

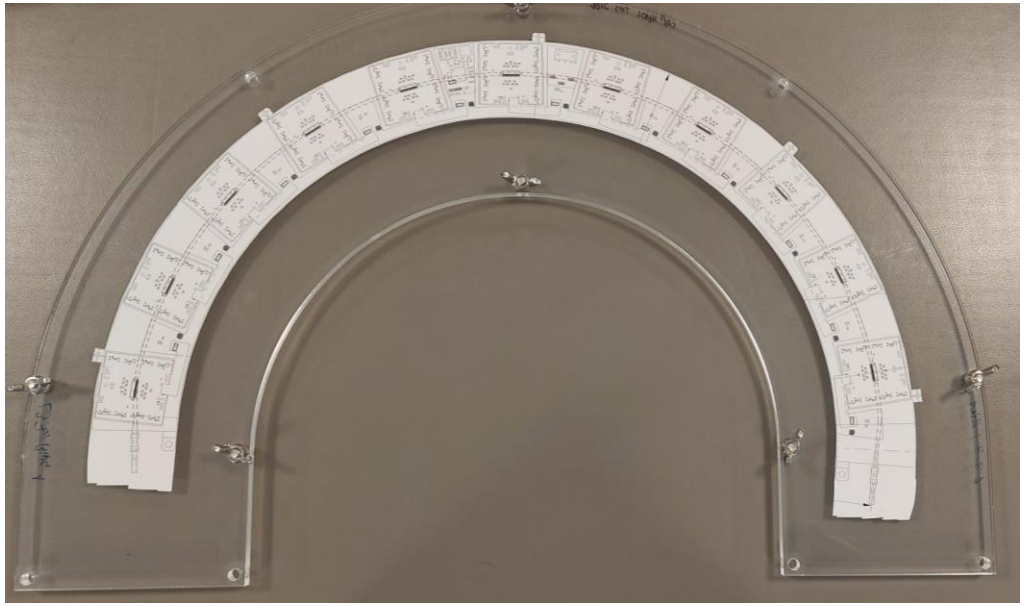


Smart scope – CAD file

- The smart scope has a feature that allows us to upload a CAD file, which provides nominal/ideal values which allow for comparison with measured values
- Uploading the CAD file of the half ring with the mounted modules provided nominal values for the height and position of the modules
- The initial attempt of using the CAD file caused the program to slow down to much due to much detail in the CAD file
- A meeting with Peter Sutcliffe (University of Liverpool) helped us make a simplified CAD file which worked perfectly

Smart scope – progress to date

- Due to issues with the half ring handling frame, and a lack of modules, a model was made from a printout placed between two plastic sheets
- The current program I have written can successfully locate each module and give the position of each of its corners in Cartesian coordinates
- The coordinates measured by the smart scope are automatically compared to the CAD's nominal values and states whether they are within tolerance or not
- Current issue is using the laser to measure the height of the modules, it has trouble locating and focusing due to the plastic sheeting reflective qualities
- Height measurements have been carried out using a prototype, but there is no corresponding CAD file of it

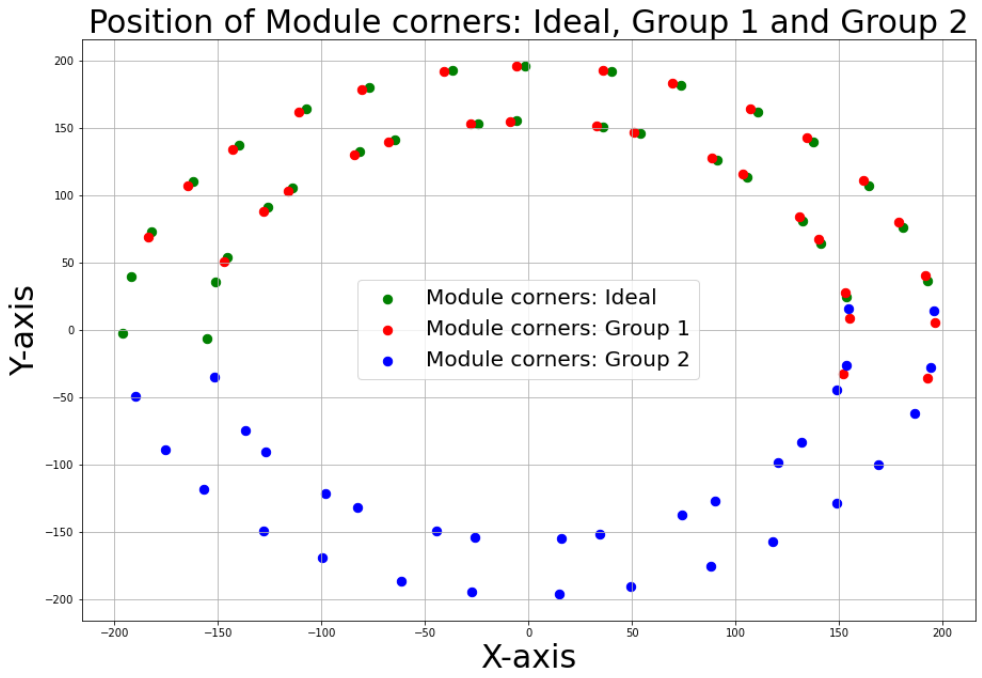


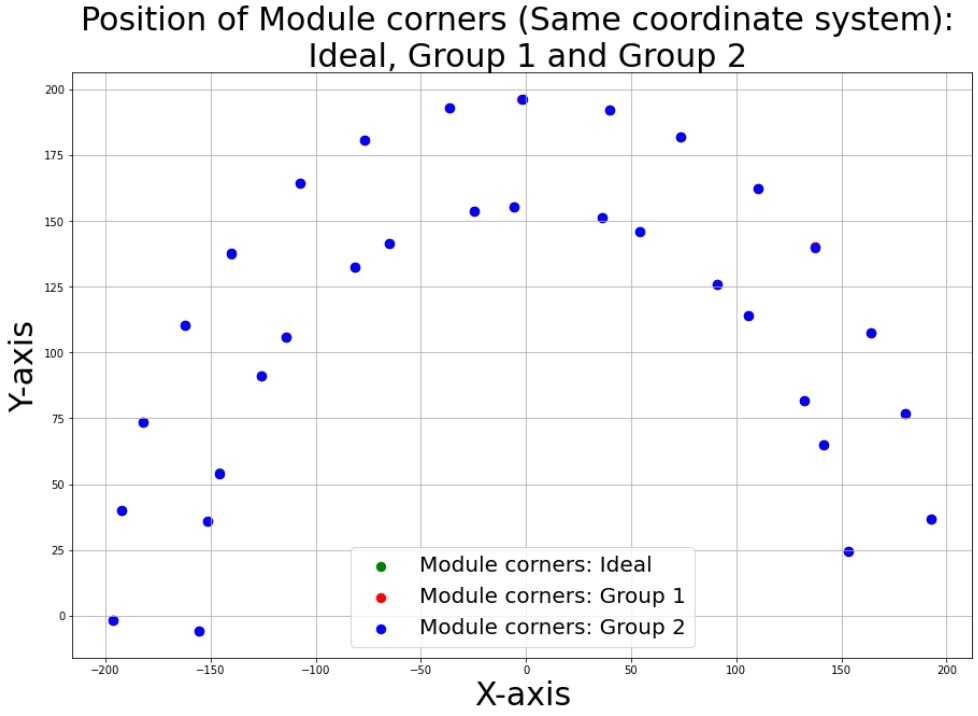
Comparing measurements

- Another task is to write a piece of code that compares the measurements of the module positions to an ideal
- Additionally, it should compare the placement of the modules between two different groups, in this case between RAL and Lecce
- My code was written in python with Jupyter notebook

Position comparison - Different coordinate systems

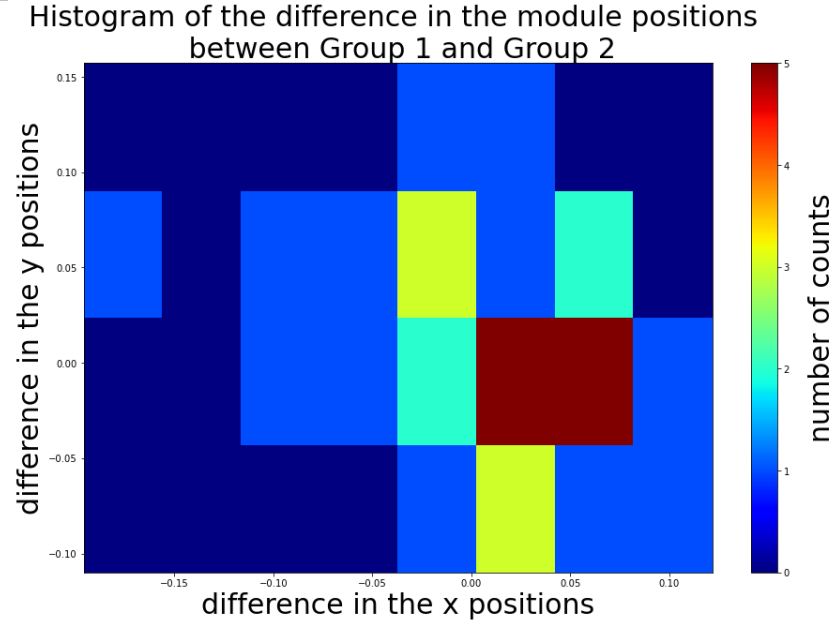
- The main problem with comparing the positions is that the UK and Italy using different coordinate system to place and measure the modules, making comparison difficult
- My code brings the two sets of data into the same coordinate system as a set of ideal data
- Half ring positions are defined via the fiducial markers
- The code treats the sets of coordinates as matrices then determines and applies matrix transformations
- These transformations are stored and applied to the module position data, bringing them all into the same coordinate system

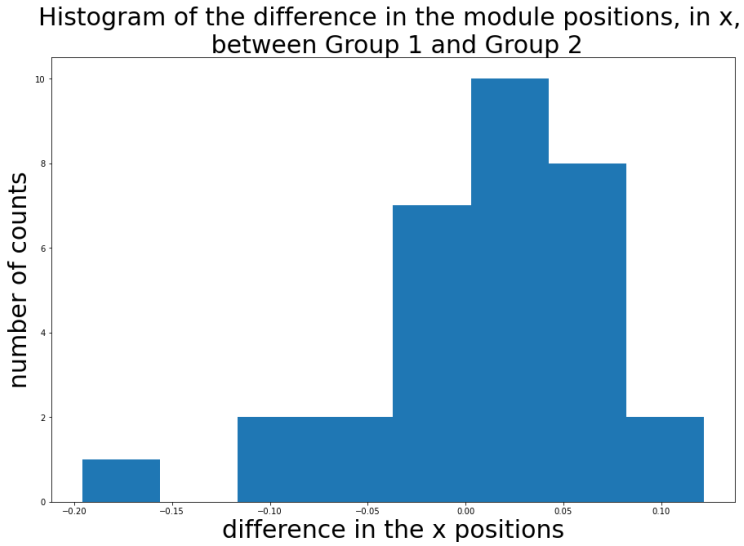


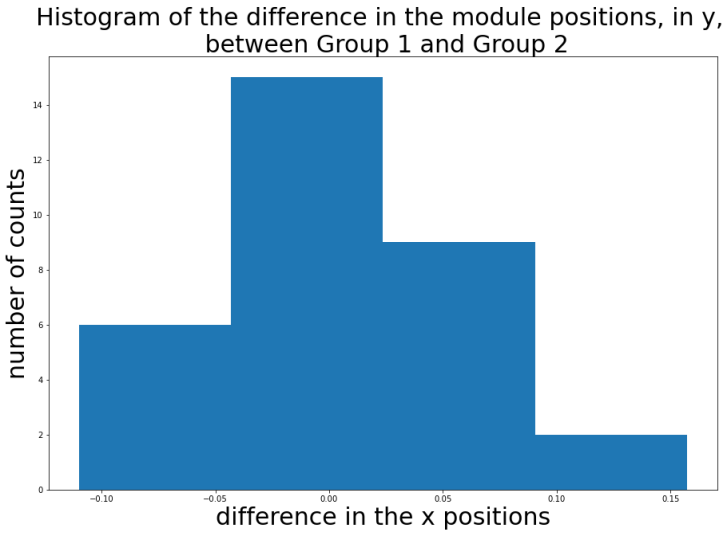


Position comparison - Difference in the measurements

- Once the data sets are brought into the same coordinate system they are compared with the ideal, with each other, and all the relevant information is saved to a python dictionary
- The dictionary is saved to a file, and then read into a separate piece of code that plots the differences of the measurements
- The code also calculates standard deviations on the differences in the placements
- Thanks go towards Ben Smart and John Matheson, offering suggestions and advice when needed

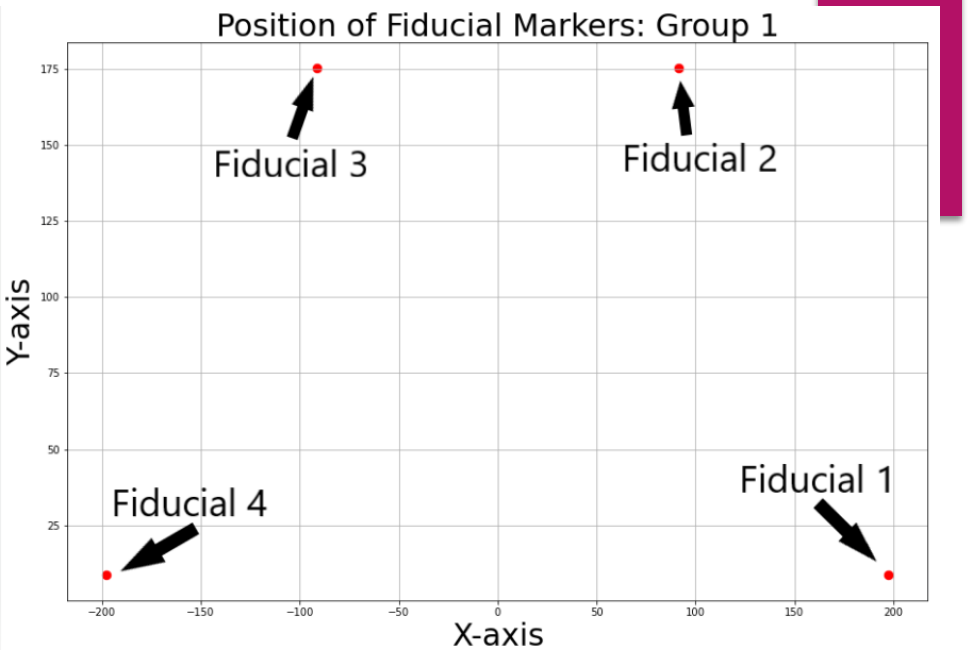


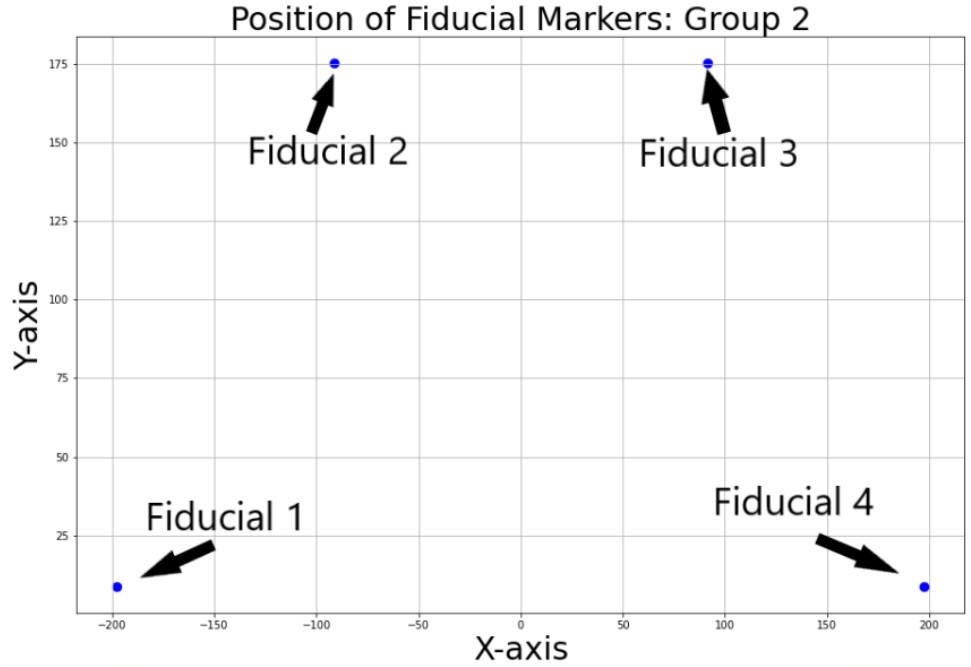




Position comparison - Current issues, and future developments

- Due to symmetries in the layout of the fiducial the user must correctly label their fiducials in the input datasets
- Additionally, the code currently requires that the modules be in the same order in each data set
- To solve the module order issue, code will be written to self identify and pair up the appropriate positions automatically





Summary

- My goal was to develop a method of measuring the height and position of the modules and to compare these measurements both to an ideal positions and measurements carried out by another group
- A program that measures the positions of the modules on the smart scope is complete
- A program for data comparison has be written and will be developed further
- To finish: develop comparison code that works regardless of the order of the module positions in the input data
- For the future: comparison of the module heights to CAD data using code I have written

Thank you for listening

Any questions?