

Streamlining TMO Operations with Custom GUIs and Intelligent Camera Monitoring

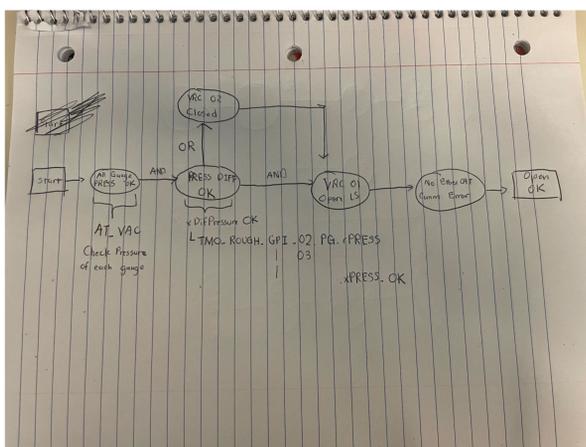
Leo Gong, Tong Ju, Jing Yin

TMO, LCLS, SLAC

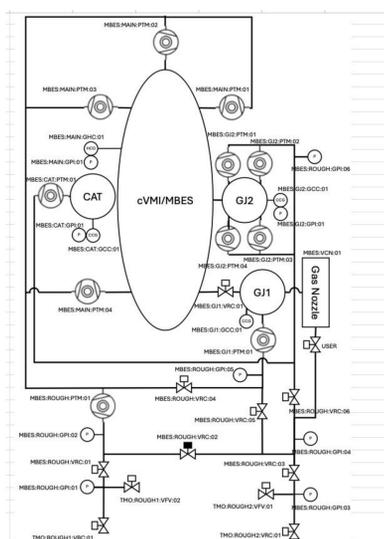


BACKGROUND

- Reliable, user-friendly controls for high-rep-rate TMO experiments.
- Pain points: uninformative GUI, safety interlocks being unintuitive, misaligned diagnostic cameras
- Objectives: Build two GUIs (MBES control & Interlock control) that lower setup time and reduce operator error.

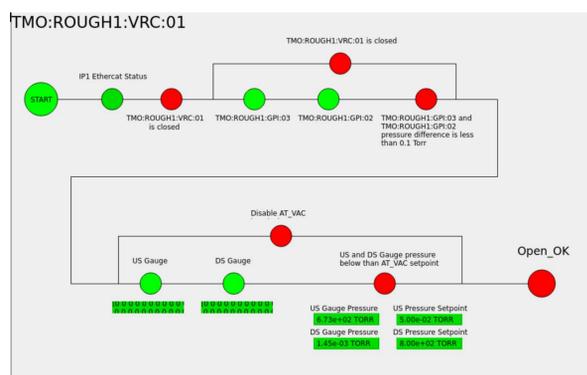


- Fix a previous camera-misalignment detector script to automatically flag bumped or moved cameras.

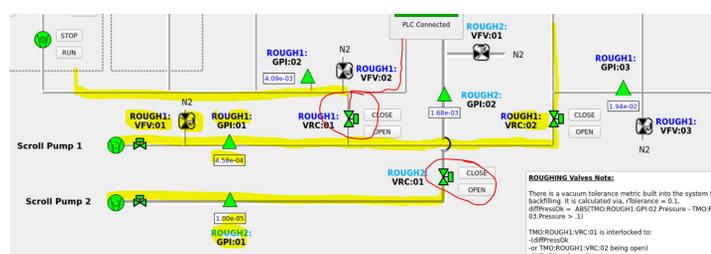


GUI DELIVERABLES

- MBES (not fully built physically yet) GUI finished.
- EPICS PVs hooked up for when building is finished
- May be a little big but can be iterated on in the future



- Interlock GUI iterations created
- Still require scientist feedback for further improvements
- Much more intuitive, much more information with less clutter
- Other interlocks not finished but are far simpler than ROUGH1.



CAMERA SCRIPT

- Concept: Uses reference frame subtraction + PSNR/SSIM metrics to detect shifts > N pixels after grabbing database of devices.
- Runs as a Bluesky callback, and alerts users if there is enough of a discrepancy detected
- To Do: Fully fix the script's environment to modern day conda environment due to the script being over two years old.
- Very helpful for scientists to ensure reduced downtimes and accurate camera readings.

FUTURE WORK

- Integrate auto-re-zeroing of cameras.
- Extend to multiple cameras and TMOs.

ACKNOWLEDGEMENTS

Robert Tang Kong - Helped a lot through maneuvering around the environment and some of the software

REFERENCES

PyDM, Bluesky, OpenCV