Science and Technology Facilities Council

ISIS Neutron and Muon Source

Welcome

Science and
Technology
Facilities Council

ISIS Neutron and
Muon Source

Using EPICS to facilitate Machine Learning (ML) at ISIS

EPICS Collaboration Meeting September 2022

Kathryn Baker

ML in the ISIS Accelerators

The transition to EPICS to control the ISIS accelerators presents an opportunity for us to integrate Machine Learning into our operations for a variety of reasons:

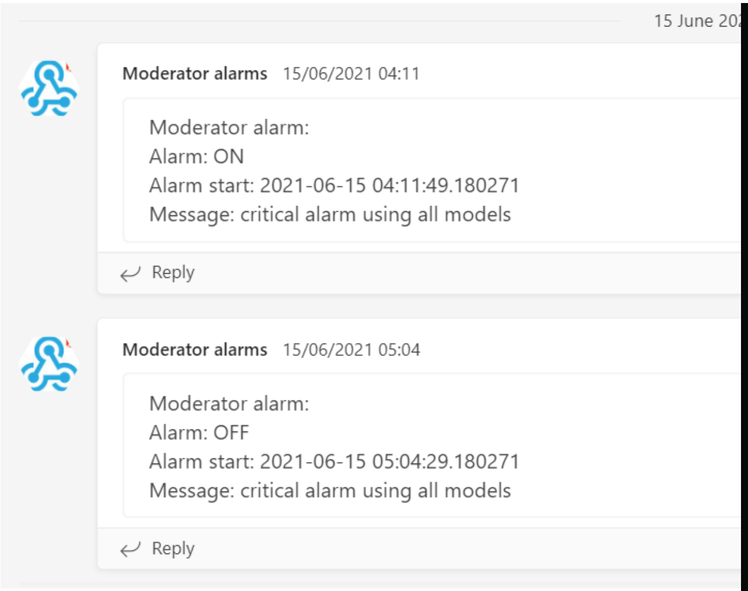
- Linux is easier to interact with than OpenVMS
- Extensive Python libraries make it easy to get and set values in EPICS

Science and Technology Facilities Council

ISIS Neutron and Muon Source

Current Projects – Anomaly Detection

Although in early days, we have a number of projects currently ongoing at ISIS that make use of Machine Learning and optimisation.



15 June 202

Moderator alarms 15/06/2021 04:11

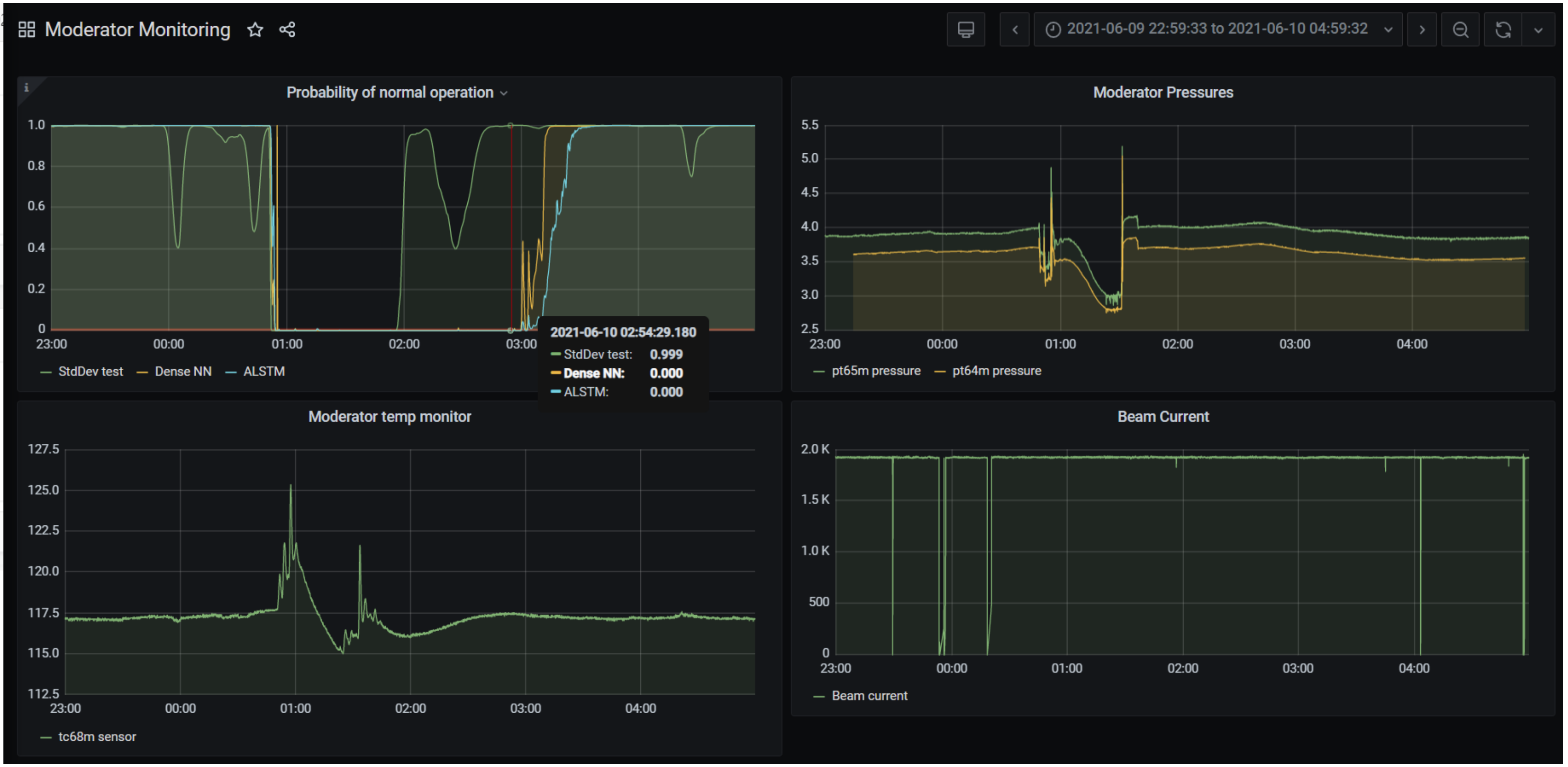
Moderator alarm:
Alarm: ON
Alarm start: 2021-06-15 04:11:49.180271
Message: critical alarm using all models

Reply

Moderator alarms 15/06/2021 05:04

Moderator alarm:
Alarm: OFF
Alarm start: 2021-06-15 05:04:29.180271
Message: critical alarm using all models

Reply

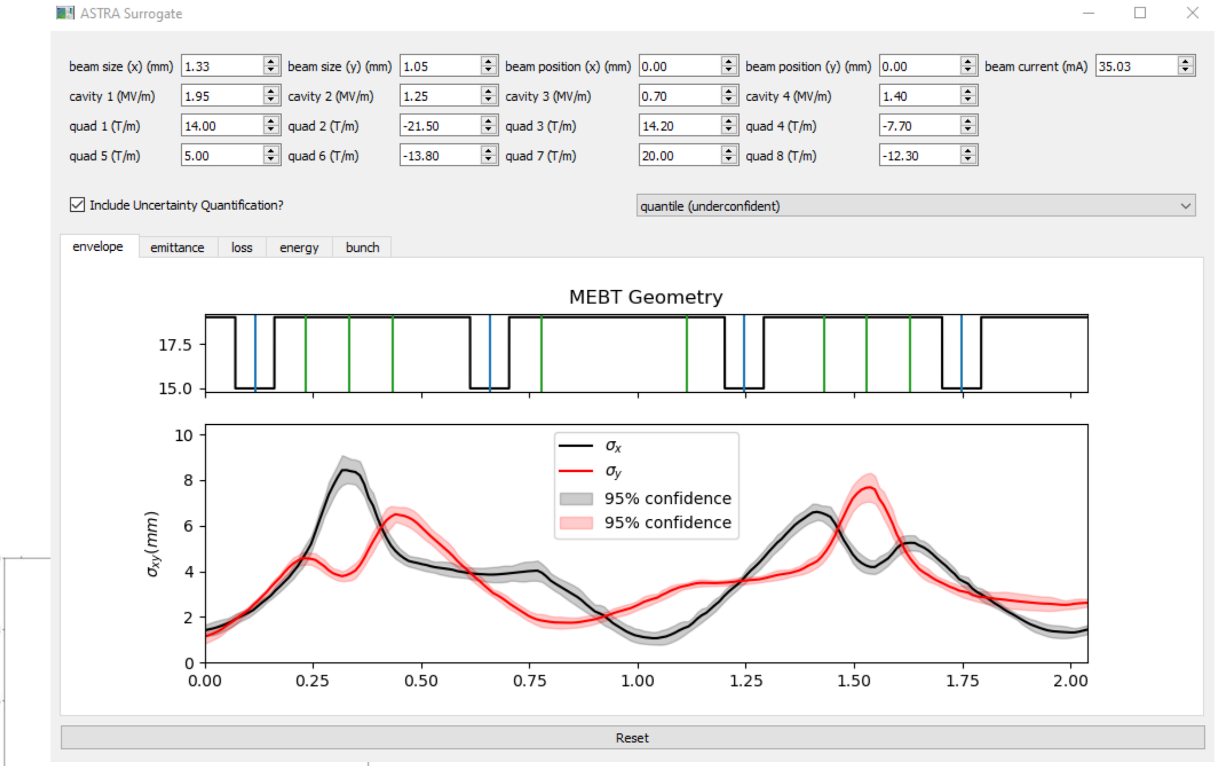


Science and Technology Facilities Council

ISIS Neutron and Muon Source

Current Projects – Virtual Diagnostics

Although in early days, we have a number of projects currently ongoing at ISIS that make use of Machine Learning and optimisation.



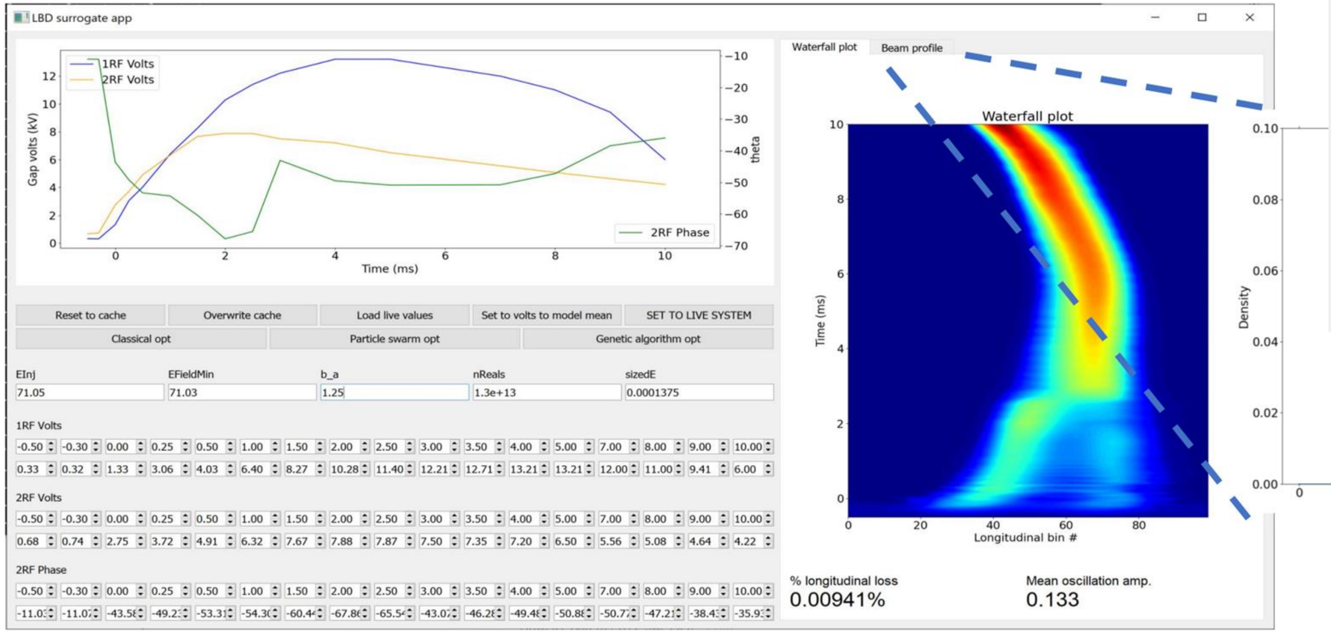
beam size (x) (mm) 1.33 beam size (y) (mm) 1.05 beam position (x) (mm) 0.00 beam position (y) (mm) 0.00 beam current (mA) 35.03

cavity 1 (MV/m) 1.95 cavity 2 (MV/m) 1.25 cavity 3 (MV/m) 0.70 cavity 4 (MV/m) 1.40

quad 1 (T/m) 14.00 quad 2 (T/m) -21.50 quad 3 (T/m) 14.20 quad 4 (T/m) -7.70

quad 5 (T/m) 5.00 quad 6 (T/m) -13.80 quad 7 (T/m) 20.00 quad 8 (T/m) -12.30

☑ Include Uncertainty Quantification? quantile (underconfident)



EInj 71.05

EFieldMin 71.03

b_a 1.25

nReals 1.3e+13

sizedE 0.0001375

1RF Volts

-0.50	-0.30	0.00	0.25	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	5.00	7.00	8.00	9.00	10.00
0.33	0.32	1.33	3.06	4.03	6.40	8.27	10.28	11.40	12.21	12.71	13.21	13.21	12.00	11.00	9.41	6.00

2RF Volts

-0.50	-0.30	0.00	0.25	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	5.00	7.00	8.00	9.00	10.00
0.68	0.74	2.75	3.72	4.91	6.32	7.67	7.88	7.87	7.50	7.35	7.20	6.50	5.56	5.08	4.64	4.22

2RF Phase

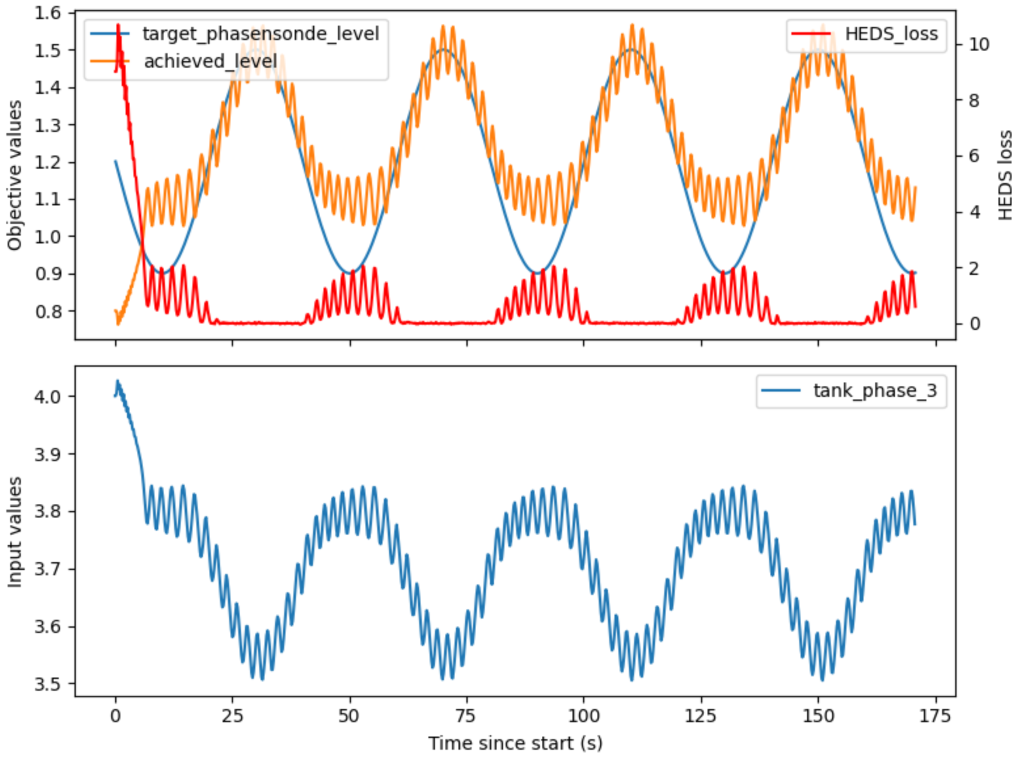
-0.50	-0.30	0.00	0.25	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	5.00	7.00	8.00	9.00	10.00
-11.03	-11.07	-43.58	-49.23	-53.31	-54.30	-60.44	-67.86	-65.54	-43.07	-46.28	-49.48	-50.88	-50.77	-47.21	-38.43	-35.93

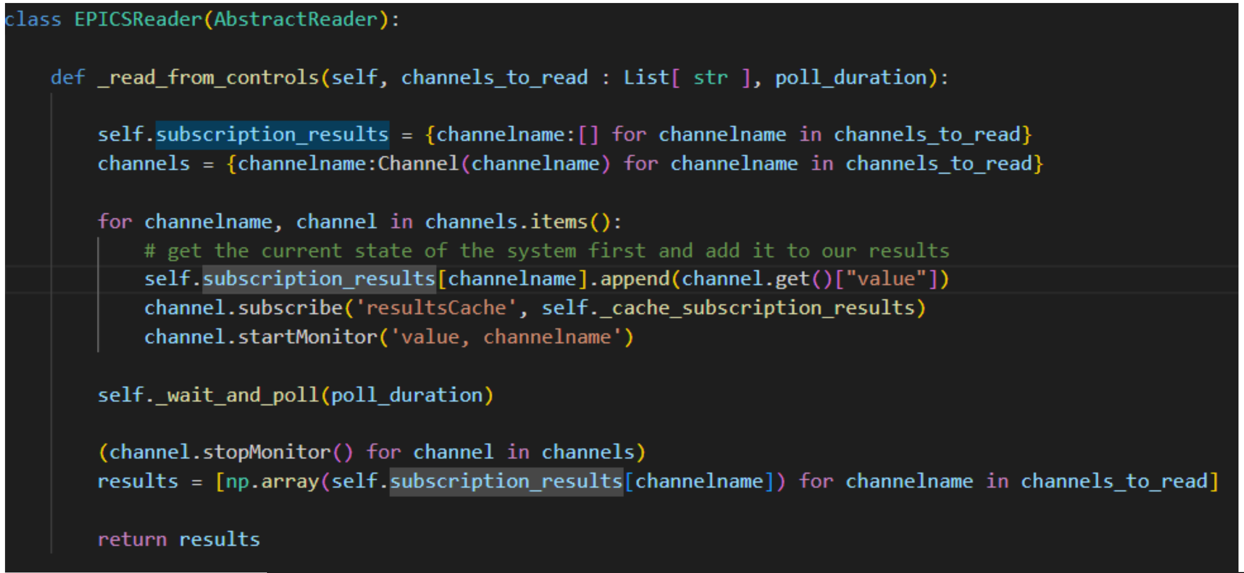
% longitudinal loss 0.00941%

Mean oscillation amp. 0.133

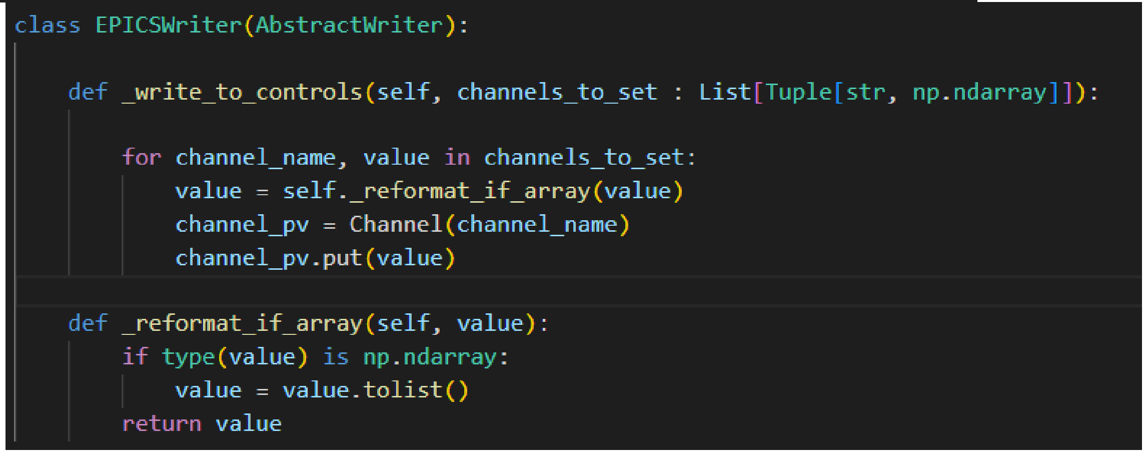
Current Projects - Optimisation

Although in early days, we have a number of projects currently ongoing at ISIS that make use of Machine Learning and optimisation.



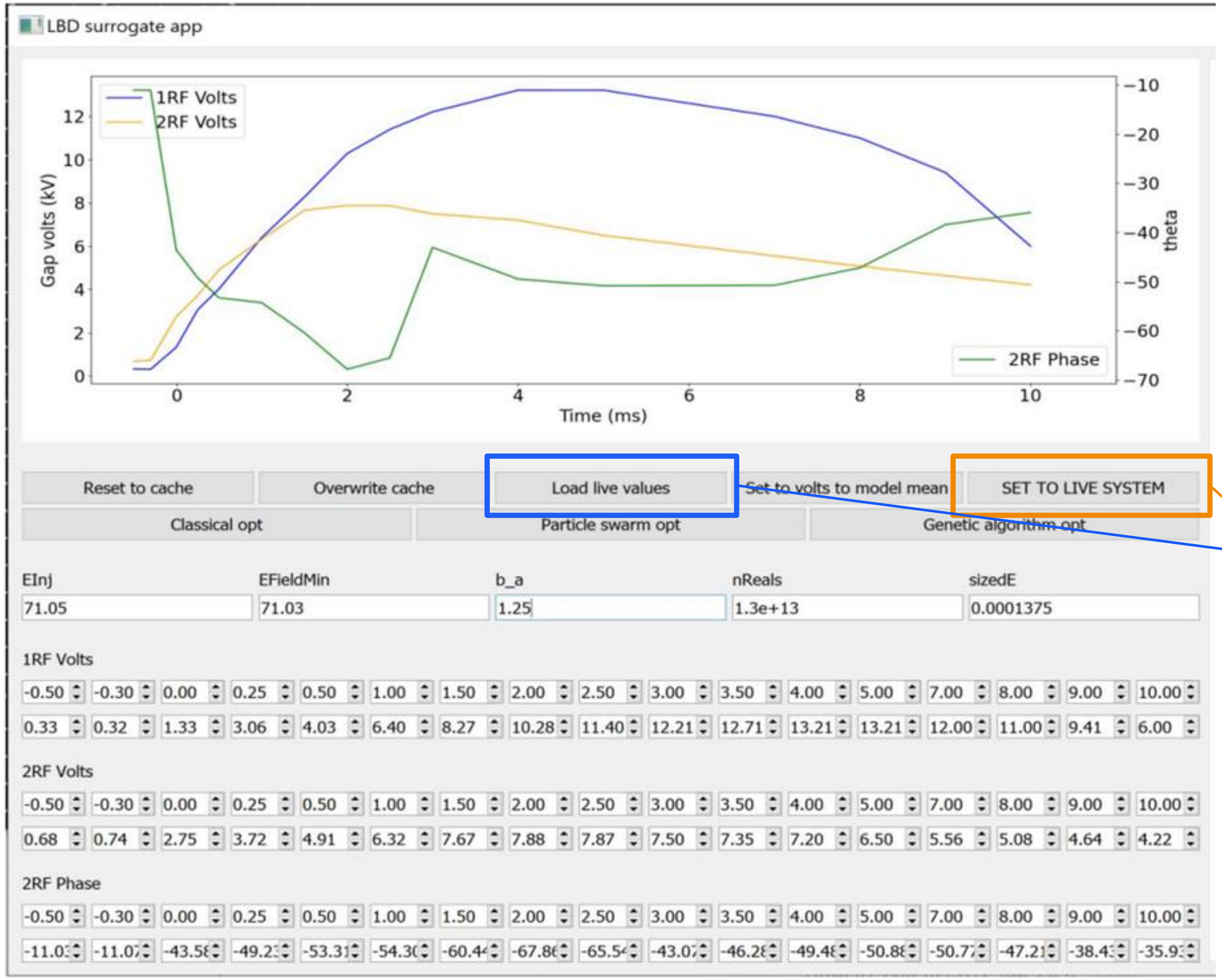


```
class EPICSReader(AbstractReader):

    def _read_from_controls(self, channels_to_read : List[ str ], poll_duration):

        self.subscription_results = {channelname:[] for channelname in channels_to_read}
        channels = {channelname:Channel(channelname) for channelname in channels_to_read}

        for channelname, channel in channels.items():
            # get the current state of the system first and add it to our results
            self.subscription_results[channelname].append(channel.get()["value"])
            channel.subscribe('resultsCache', self._cache_subscription_results)
            channel.startMonitor('value, channelname')

        self._wait_and_poll(poll_duration)

        (channel.stopMonitor() for channel in channels)
        results = [np.array(self.subscription_results[channelname]) for channelname in channels_to_read]

        return results
```



```
class EPICSWriter(AbstractWriter):

    def _write_to_controls(self, channels_to_set : List[Tuple[str, np.ndarray]]):

        for channel_name, value in channels_to_set:
            value = self._reformat_if_array(value)
            channel_pv = Channel(channel_name)
            channel_pv.put(value)

    def _reformat_if_array(self, value):
        if type(value) is np.ndarray:
            value = value.tolist()
        return value
```

EPICS Integration



EInj	EFieldMin	b_a	nReals	sizedE
71.05	71.03	1.25	1.3e+13	0.0001375

1RF Volts

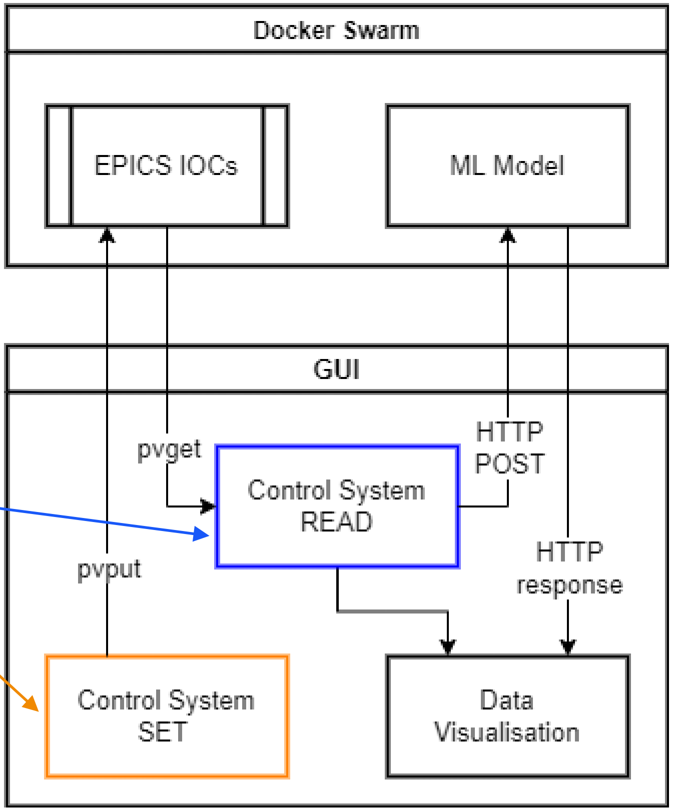
-0.50	-0.30	0.00	0.25	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	5.00	7.00	8.00	9.00	10.00
0.33	0.32	1.33	3.06	4.03	6.40	8.27	10.28	11.40	12.21	12.71	13.21	13.21	12.00	11.00	9.41	6.00

2RF Volts

-0.50	-0.30	0.00	0.25	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	5.00	7.00	8.00	9.00	10.00
0.68	0.74	2.75	3.72	4.91	6.32	7.67	7.88	7.87	7.50	7.35	7.20	6.50	5.56	5.08	4.64	4.22

2RF Phase

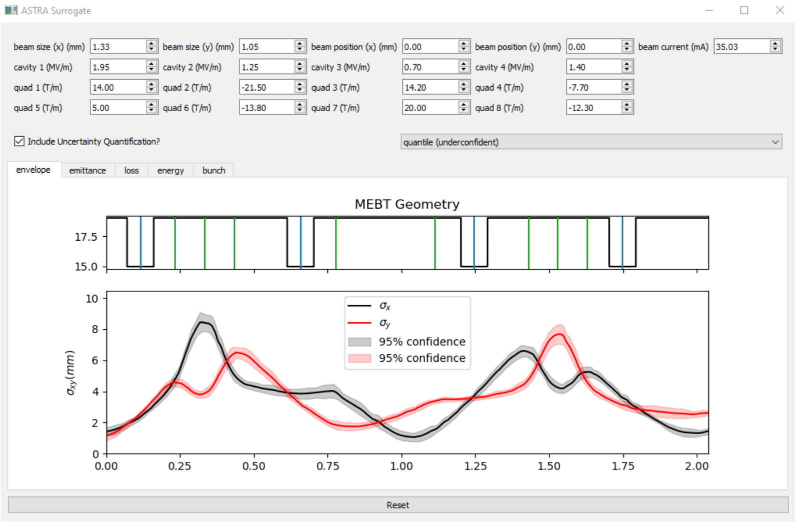
-0.50	-0.30	0.00	0.25	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	5.00	7.00	8.00	9.00	10.00
-11.03	-11.07	-43.58	-49.23	-53.31	-54.30	-60.44	-67.86	-65.54	-43.07	-46.28	-49.48	-50.88	-50.77	-47.21	-38.43	-35.93



Deployment?

Still not sure how to deploy them…

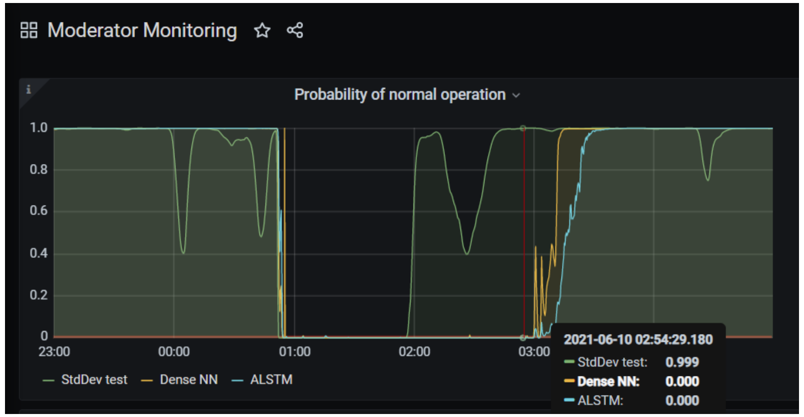
- **PyQt GUIs** – simple but requires developer effort to create
- **Web based** control screen – a chance to adopt something new?
- **Phoebus** (or alternative?) screen – great ‘out-of-the-box’ solution but unknown how to integrate ML models



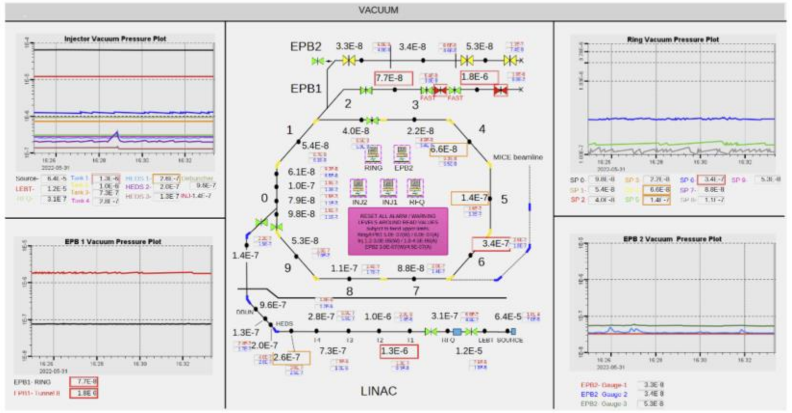
ASTRA Surrogate

beam size (x) (mm) 1.33 | beam size (y) (mm) 1.05 | beam position (x) (mm) 0.00 | beam position (y) (mm) 0.00 | beam current (mA) 35.03
cavity 1 (MV/m) 1.95 | cavity 2 (MV/m) 1.25 | cavity 3 (MV/m) 0.70 | cavity 4 (MV/m) 1.40
quad 1 (T/m) 14.00 | quad 2 (T/m) -21.50 | quad 3 (T/m) 14.20 | quad 4 (T/m) -7.70
quad 5 (T/m) 5.00 | quad 6 (T/m) -13.80 | quad 7 (T/m) 20.00 | quad 8 (T/m) -12.30

☑ Include Uncertainty Quantification? quantile (underconfident)



2021-06-10 02:54:29.180
StdDev test: 0.999
Dense NN: 0.000
ALSTM: 0.000



Science and Technology Facilities Council

ISIS Neutron and Muon Source

Science and
Technology
Facilities Council

ISIS Neutron and
Muon Source

Thank you

isis.stfc.ac.uk @ISISNeutronMuon STFC