AIOP Photon Source Data Analysis

Will Imoehl

May 2025

Contents

1 Introduction/Motivation

The AIOP project aims to automate the coherent peak "nudging" procedure by using an ML algorithm to continuously adjust the pitch and/or yaw of the diamond. In GlueX, the polarization of the photon peaks near the coherent edge, which is where the steep dropoff in energy occurs. By automating the energy nudging procedure, we ensure a more consistent coherent edge position over time, which will reduce the run-to-run spread in the beam energies. When integrating over the full run period, this will result in a sharper coherent peak edge, as well as a higher level of polarization. This improvement will reduce systematic uncertainties in the polarization, which could make our physics results more sensitive.

Would be good to have a viz of a good coherent peak, and what it looks like if you do time-weighted avg of a couple good ones and a bad one. Maybe with polarization too.

The stated goal of this project is to keep the coherent peak energy within a ±10 MeV window of the desired set-point energy over the full run period. While the primary handle that Hall D has on the coherent edge position is the pitch and yaw of the diamond, we know that there are other variables that can also affect the photon beam energy. These include changes in the position or energy of the electron beam, thermal effects, vibrations on the goniometer on which the diamond is mounted, and diamond degradation over time [1]. Patrick is focusing on studying diamond degradation over time and the effect of the x and y position of the electron beam, while my studies are focused on how nudge sequences change the coherent edge position in existing data. This includes measuring how much the energy changes per nudge for each run period, as well as understanding the impact of backlash in the goniometer.

'HD:coda:daq:run_number'	'HD:coda:daq:status'	'HD:CBREM:PLANE'	'HD:GONI:RADIATOR_ID'
'HD:GONI:X'	'HD:GONI:X.RBV'	'HD:GONI:Y'	'HD:GONI:Y.RBV'
'HD:GONI:PITCH'	'HD:GONI:PITCH.RBV'	'HD:GONI:YAW'	'HD:GONI:YAW.RBV'
'HD:GONI:ROLL'	'HD:GONI:ROLL.RBV'	'HD:CBREM:REQ_EDGE'	'HD:GONI:RADIATOR_NAME'
'HD:CBREM:EDGE'	'HD:CBREM:WIDTH'	'AC:inner:position:x'	'AC:inner:position:y'
'IPM5C11B.XPOS'	'IPM5C11B.YPOS'	'IBCAD00CRCUR6'	'HALLD:p'

Table 1: Table summarizing all the EPICS variables we get from the archive.

2 Data Collection and Processing

This section describes the procedures I use to obtain, clean, and merge the data.

2.1 Obtaining Data From myget

The myget [3] package can retrieve archived data from Mya. Specifically, myget will report each time a variable changes in the Mya archive. We use myget [3] to obtain the full history for each variable listed in Tab. 1. An example myget command showing how we obtain the run number for Spring 2020 is shown below.

```
myget -m history -c 'HD:coda:daq:run_number' -b '2020-01-11 00:00:00' -e '2020-09-21 12:00:00' | tee run_number.txt
```

This procedure is carried out for the Spring 2020, Spring 2023, and Spring 2025 run periods. A list of the exact commands used are given in Appendix B

Most variables are self explanatory. The bottom row has the x and y position, current, and energy of the electron beam. HD:CBREM:WIDTH is the measured uncertainty in the coherent edge position. A link to the spreadsheet showing the EPICS variables and their definitions is given as Ref. [4].

2.2 Conversion to .csv files

The output from myget is initially saved as a series of 24 .txt files for each run period. The code in

```
cleanAndConvertData_txt_to_csv.py
```

converts each of these .txt files into .csv files. In other words, it will take, for example

```
ac_x.txt -> ac_x.csv
```

During this conversion process, the following data cleaning steps are performed:

1. any rows with a "<< Network disconnection >>" message are removed.
2. Time stamps are only recorded every second. Sometimes, there are multiple recorded values in less than a second, especially for the readback values. In these cases, we only take the first row with the given time stamp.
3. Time stamps are reformatted to look like "[YEAR]-[MONTH]-[DAY]T[HOUR]:[MINUTE][SECOND]", which can automatically be converted into a datetime object in Python.

2.3 Merging multiple .csv files into a master .csv

The code for this section is in

```
merge_all_csv.ipynb
```

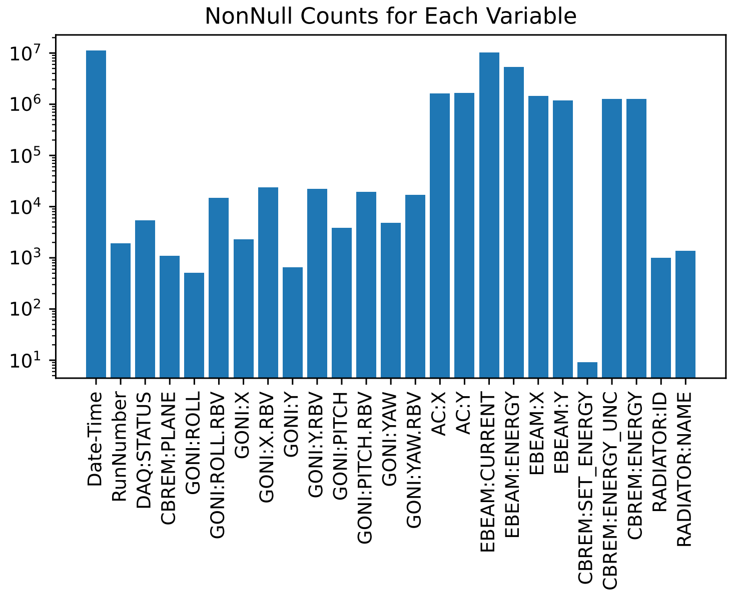

Figure 1: Figure showing the number of not NaN values for each variable after the outer joins are complete. Measured values have the highest counts, while setpoint values are relatively sparse.

2.3.1 Details on Merging and Handling NaN's

The previous step gives us 24 .csv files, one for each variable of interest. We read each of these files into a separate pandas DataFrame, and then perform an outer join on the Date/Time column. Outer joins retain all rows in each column, but place them in the appropriate order by Date-Time. Any missing values are filled with a NaN during the joining procedure. A figure showing the number of non-null values for each variable is shown in Fig. 1. From this figure, it is evident that measured beamline values have the most measured values, while things like set points have substantially fewer recorded values.

To better visualize how the data is modified by the joining process, Table 2 shows a simplified, hypothetical table that could result from the merge process.

There are three types of variables here:

1. Run number: the run number only changes once the DAQ reaches the prestart phase (status 1). The remaining values will be NaN, since myget only records values when they change.
2. Desired/setpoint variable values: These only have values if the set point changed. Therefore, a NaN value means we can forward fill the last non-NaN value.
3. Measured/Readback variable values: these generally fewer missing values. If there is a NaN at a given time, it means there was either a "<<Network disconnection >>" or that the variable is the same as the last reading. In either case, it is reasonable to forward fill.

To deal with missing/NaN values, we use forward-fill, which should give the same behavior as mySampler.

Table 2: Table showing a simplified example of the initial DataFrame after all outer joins are complete. The variable values are made up for this example.

DateTime	Run	DAQ:STATUS	GONI:YAW	GONI:YAW.RBV	EBEAM:CURRENT
2021-09-17T01:11:12	NaN	0	0.80	0.80	150
2021-09-17T01:12:12	NaN	NaN	0.90	0.82	149
2021-09-17T01:12:13	NaN	NaN	NaN	0.84	NaN
2021-09-17T01:12:13	NaN	NaN	NaN	0.86	150
2021-09-17T01:12:13	NaN	NaN	NaN	0.88	NaN
2021-09-17T01:12:13	NaN	NaN	NaN	0.90	151
2021-09-17T01:15:00	70356	1	NaN	NaN	150
2021-09-17T01:15:05	NaN	NaN	NaN	NaN	152
2021-09-17T01:15:37	NaN	NaN	NaN	NaN	NaN
2021-09-17T01:16:00	NaN	2	NaN	NaN	150

Table 3: Table showing a simplified example of the final merged DataFrame after the outer joins and cleanup steps are completed. The variable values are made up for this example.

DateTime	Run	DAQ:STATUS	GONI:YAW	GONI:YAW.RBV	EBEAM:CURRENT
2021-09-17T01:11:12	70356	0	0.80	0.80	150
2021-09-17T01:12:12	70356	0	0.90	0.82	149
2021-09-17T01:12:13	70356	0	0.90	0.84	149
2021-09-17T01:12:13	70356	0	0.90	0.86	150
2021-09-17T01:12:13	70356	0	0.90	0.88	150
2021-09-17T01:12:13	70356	0	0.90	0.90	151
2021-09-17T01:15:00	70356	1	0.90	0.90	150
2021-09-17T01:15:05	70356	1	0.90	0.90	152
2021-09-17T01:15:37	70356	1	0.90	0.90	152
2021-09-17T01:16:00	70356	2	0.90	0.90	150

After this step, there are two issues. The first few rows of the table many have NaN's that still exist, because there is no value to forward fill here. To deal with these, we back-fill after the initial forward fill. A better solution may just be to drop these initial rows, since the set points were necessarily different before this.

The second issue comes with the run number. After the forward fill, run number 70356 will extend from when run 70356 prestarted, all the way until the next run has prestarted. This means all nudges during the setup before the prestart will be assigned to the previous run. To fix this, we iterate through the DataFrame and assign the run number for all rows with DAQ:STATUS of 0 to the next run number defined by DAQ:STATUS 1.

After these steps are carried out, the hypothetical table above is transformed to the one shown in Table 3.

2.3.2 Casting Variable Types

The variables in the DataFrame are all automatically converted to floats during the mergine process. To reduce the size of the data a bit, we convert 'RunNumber', 'DAQ:STATUS', 'CBREM:PLANE', and 'RADIATOR:ID' to integers.

2.4 Filtering Data

Up to this stage, we have not removed any rows from the DataFrame. For Spring 2020, the coherent edge fitting script was broken for some runs. To avoid these runs, we remove all runs before 72094, which is when the script was fixed.

We also require all runs in the final DataFrame are included in the production runs for Spring 2020, given by the RCDB query

```
@is_dirc_production and @status_approved
```

Doing this is debatable, we probably don't want to, because we lose a handle on backlash.

2.5 Defining New Variables

In this subsection, we define new variables in the DataFrame to make selecting interesting features easier. We also want to be able to visualize variables as a function of time, so we create a new column that converts the Date-Time string into a timestamp using the Python datetime package:

```
from datetime import datetime
time_convert = lambda x: x.timestamp()
df_good['TimeStamp'] = pd.to_datetime(df_good['Date-Time']).apply(time_convert)
```

2.5.1 Variables Describing Nudges

The next step is to create variables related to changes in the pitch and yaw set points. We defined a nudge to be any change in the set point for the pitch and yaw. There are two types of variables related to nudges: run-wide properties and nudge-specific properties. For instance, run-wide variables include:

- RunHasNudge - True if there is at least one nudge in the run.
- NudgeDuringRun - True if there was a nudge while DAQ was on (status 2)
- TotalNudgeSize_thisRun - the total (quadrature sum) of the change in pitch and yaw from the start to the end of the nudge sequence
- NudgeSequence - not defined yet. Idea is to encode the nudge sequence as a string
- TotalNudges_thisRun - the total number of nudges that occur during the run
- RunHasAbnormalNudge - boolean - some nudges are abnormally large $> 1°$. These likely occur right after an orientation change.
- RunHasBacklash - True if the nudge direction changes during the run. Currently, I don't look into history past the current run. Therefore, I could miss if the motor motion reverses relative to something that happened in a previous run.

Nudge-specific variables include:

- NudgeOccurred - True only for rows in which the pitch and yaw set points changed.
- NudgeNumber - a unique integer for each nudge in the run.
- NudgePitchSize - signed change in the pitch set point

- NudgeYawSize - signed change in the yaw set point
- MotionDone - True only if the readback values have stabilized
- NudgeEnSize_15s - difference in coherent edge position at the time of the nudge and 15 seconds later
- NudgeEnSize_17s - difference in coherent edge position at the time of the nudge and 17 seconds later
- NudgeEnSize_20s - difference in coherent edge position at the time of the nudge and 20 seconds later

In order for a nudge to be defined, we check to make sure the CBREM:PLANE variable is either 1 or 2. When this variable is 0, it indicates switching between PARA and PERP configurations.

After this process is complete, the resulting DataFrame is saved to a .csv file as Spring2020Merged_goodRuns.csv

3 Variables Impacting Coherent Edge Energy

We know that nudging the pitch and yaw values will change the energy of the coherent edge, but there are several other variables that can impact it too. The full list of variables we expect to have an impact on the coherent edge energy are:

- Pitch
- Yaw
- Beam X position
- Beam Y position
- Electron beam current
- Electron beam energy

We look at the impact of changing the pitch and yaw in the Section 4. In this section, we investigate the correlations between the coherent edge energy and the electron beam positions (x,y), the electron beam current, and the electron beam energy.

3.1 Procedures

We remove runs that have ”nudges”, which gives us a clean data sample where pitch and yaw are stationary. This means only x, y, current, and electron energy should be impacting the coherent peak energy.

This analysis is a bit tricky. There are a few potential pitfalls. For looking at time series data, these include:

1. There is a delay between when electron beam conditions change to when the coherent edge updates.
2. Electron beam conditions can change more rapidly than the coherent edge can. Rapid changes are unlikely to be reflected in the coherent edge position.
3. Any time the beam stops, all variables will go to zero, regardless of if they are correlated with each other.
4. Forward filling missing values could ”create” correlations that are not really there. It could also ”reinforce” data points by including them multiple times in the calculations.
5. Correlation matrices/correlation coefficients assume a linear relationship. There is no reason to suspect a strictly linear relationship here.

The conclusion from the above is that times when the beam current is zero should be removed. We also remove 30 seconds after the beam was gone, to avoid the unstablilities that commonly occur when beam reappears.

3.2 A Naive Look at Correlations

In this section, we ignore all the words of caution and look at the correlation map for the electron beam position, current, and energy related to the coherent edge position. We do this separately for each orientation, since they could reasonable have different behaviors. We require the beam to have been ”up” for at least 30 seconds, in other words, it has to have been at least 30 seconds since the last time the beam dropped. The results are shown in Fig 2.

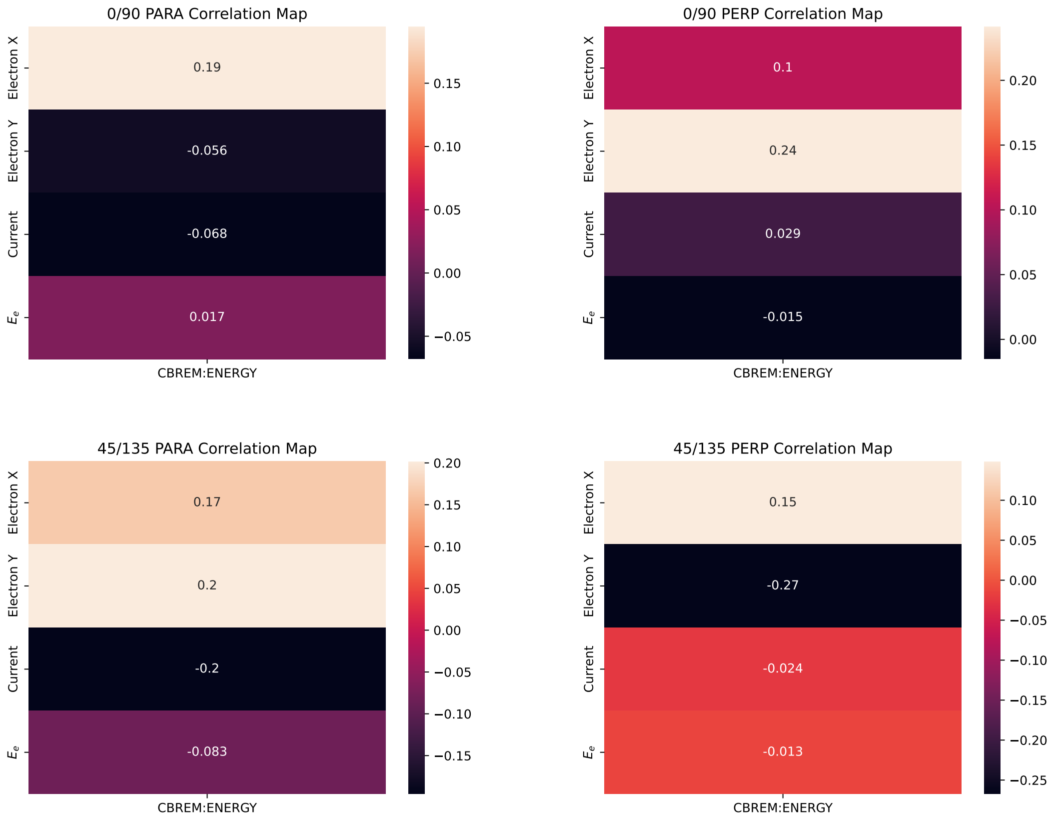

Figure 2: Correlation coefficients for four electron beam variables versus coherent edge energy.

3.3 Extreme Variations of Each Variable

In this section, we try to find individual runs that have a large variance in each electron beam variable. We start by focusing on the electron beam position, and then move onto the current and electron beam energy variables.

3.3.1 Electron Beam Position

Generally, there is a beam lock that constrains the position of the electron beam to a narrow window in x and y. In this section, we look for runs where the beam lock appears to be inactive. These runs generally have larger changes in beam position, which will make it easier to see correlations between position and the coherent edge position. We show the most dramatic example of large changes in the beam position in Fig. 3. We see the x position of the electron beam change by up to 4 mm, which results in coherent edge energies between 8500 and 8600 MeV. We can see there is not a strictly linear relationship between E_γ and electron x position, but in the typical range around -0.5 mm, the behavior looks approximately linear.

We also look at the time series progression for this run in Fig. 4. We see that jumps in the beam x position seem to precede jumps in energy by around 10-15 seconds, which we know is roughly the time it takes the coherent edge position to update.

Next, we look for a run that has a large variance in y. The correlation plots are shown in Fig. 5. We see both x and y have large correlations with the coherent peak energy in this case. The corresponding time series plot is shown in Fig. 6.

3.3.2 Electron Beam Current

Next, we look for a run with large variances in electron beam current. The variances are best visualized in a time series, as shown in Fig. 7. There are very clear oscillations of the beam current as a function of time. The corresponding correlation plots are shown in Fig. 8. One interesting thing to note is that the correlation with the electron current is relatively small, but correlations with electron beam y position and electron energy are large.

3.3.3 Electron Beam Energy

Changes in the electron beam energy will obviously have an impact on the coherent edge position. We show a dramatic example of this in Fig. 9. The coherent edge position slowly drifts over time by 1 MeV, and appears to induce a roughly 10 MeV change in the coherent edge position. The corresponding correlation plots are shown in Fig. **??**.

3.4 Nominal Beam Conditions

The previous section largely focused on runs with extreme values of electron beam parameters. In this section, we remove runs where the beam conditions were unstable.

3.5 Summary of Studies

Here we only looked at a tiny fraction of the available runs, but we see clear examples where the electron beam x and y position impact the coherent edge position. We also saw that the electron beam energy changing can have a large impact on the coherent edge energy. Things are less clear for the electron beam current. The correlation maps for each orientation show no significant ocrrelation, except for the case of the 45/135 PARA configuration, where the correlation coefficient is 0.2

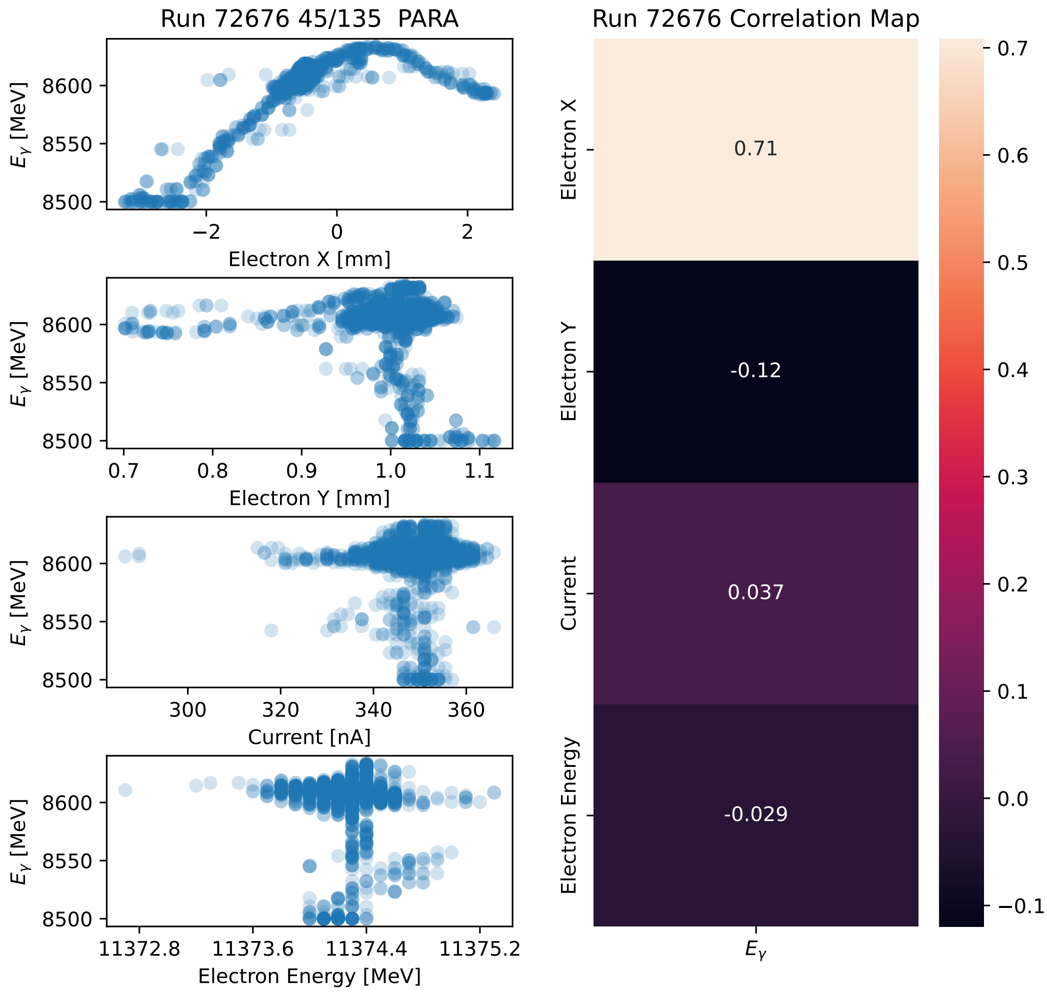

Figure 3: (left) Plots of the coherent edge energy as a function of electron beam x position (top), y position (2nd row), beam current (3rd row), and electron beam energy (final row). The right column shows the correlations between each variable and the coherent peak energy.

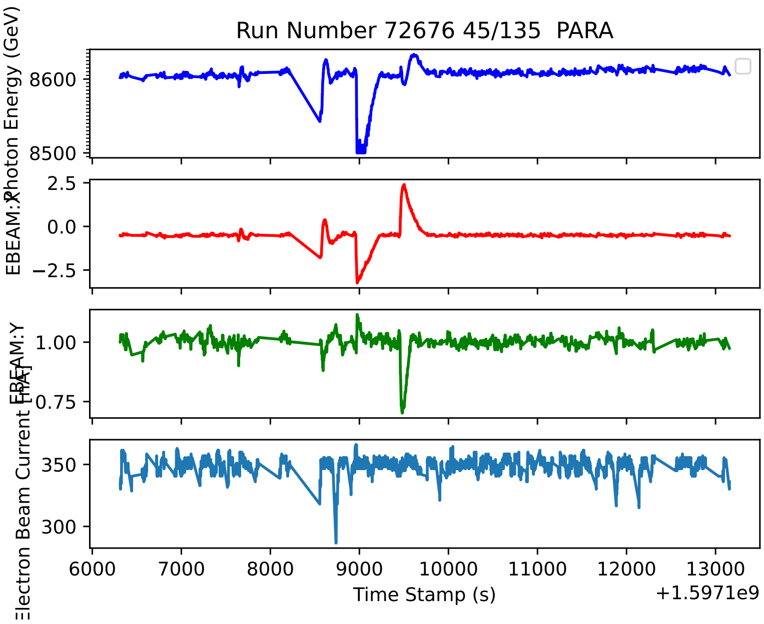

Figure 4: Time series of Run 72676, which has large fluctuations in both the x and y positions of the electron beam.

It seems like the best conclusion from these studies is that x and y appear to be the most important variables for determining beam energy in these cases. If the electron beam energy is not stable, it also needs to be taken into account. Including the beam current in the model may be useful, but so far there is no ”smoking gun” plot that shows it can impact the coherent edge position. Of course, using the beam current as a proxy variable for whether or not there is beam could be useful, but that is redundant with whether the x and y position values are non-zero.

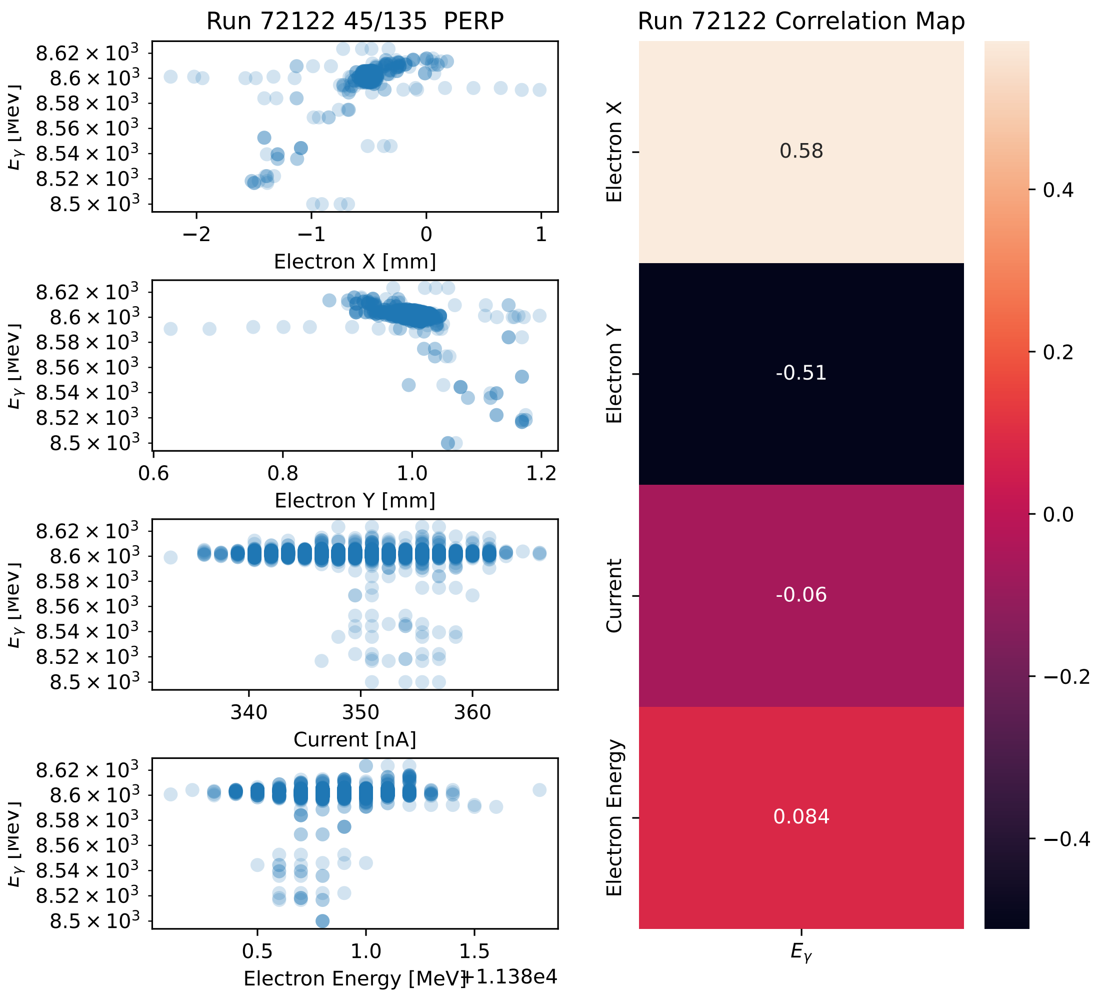

Figure 5: Caption

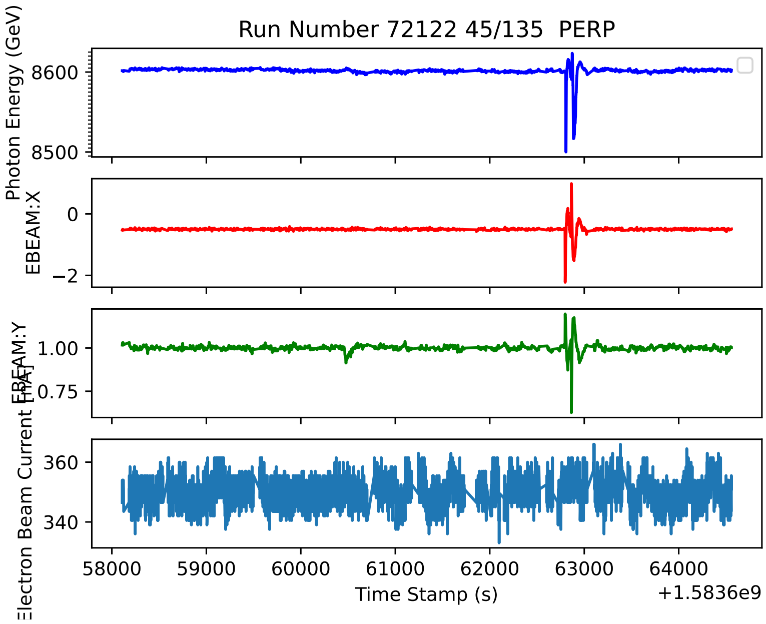

Figure 6: Caption

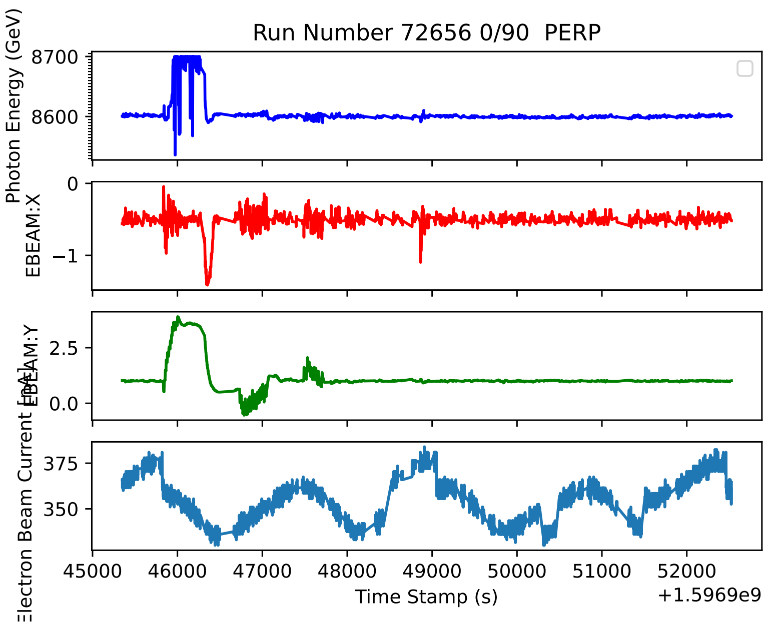

Figure 7: Time series plot where the electron beam current undergoes oscillations.

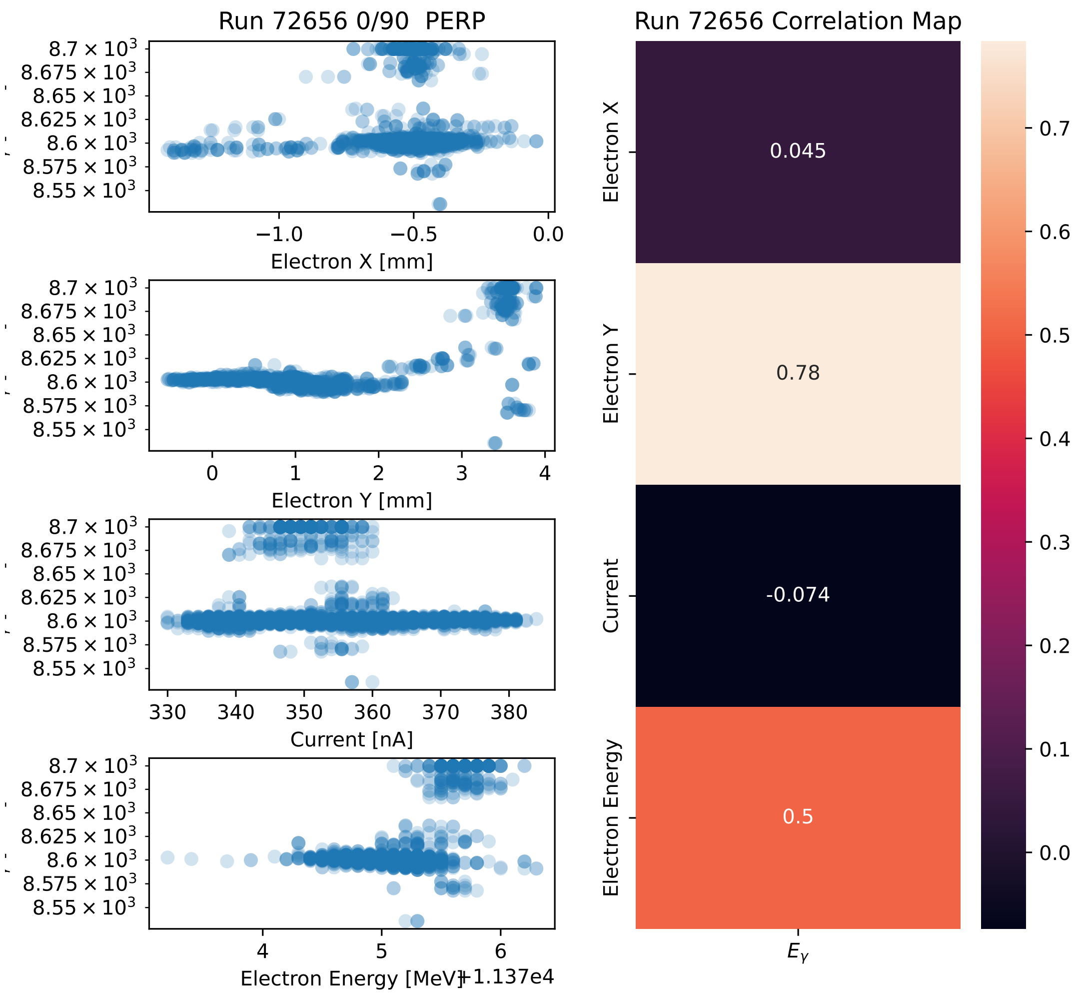

Figure 8: Correlation plots for Run 72656, which has a large periodic fluctuation in beam current.

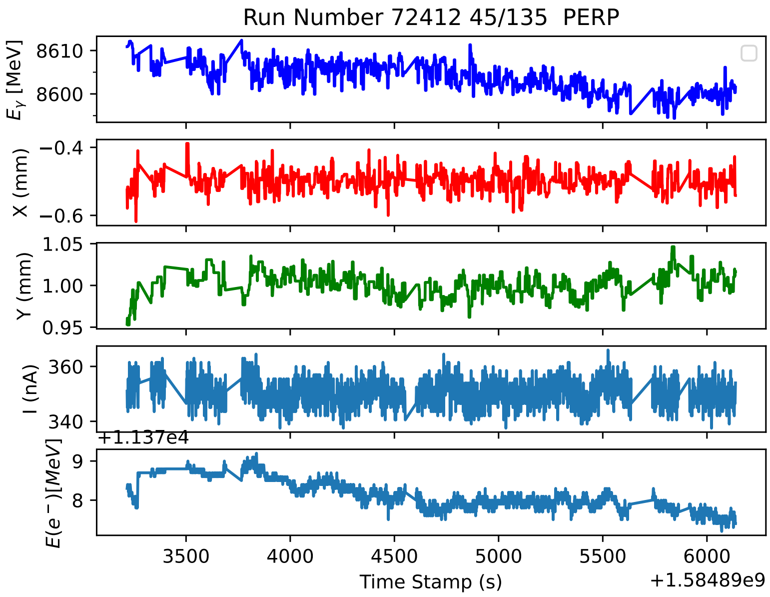

Figure 9: Time series plot for a run where the electron beam energy changes roughly 1 MeV, which appears to induce a change in the coherent edge position of around 10 MeV.

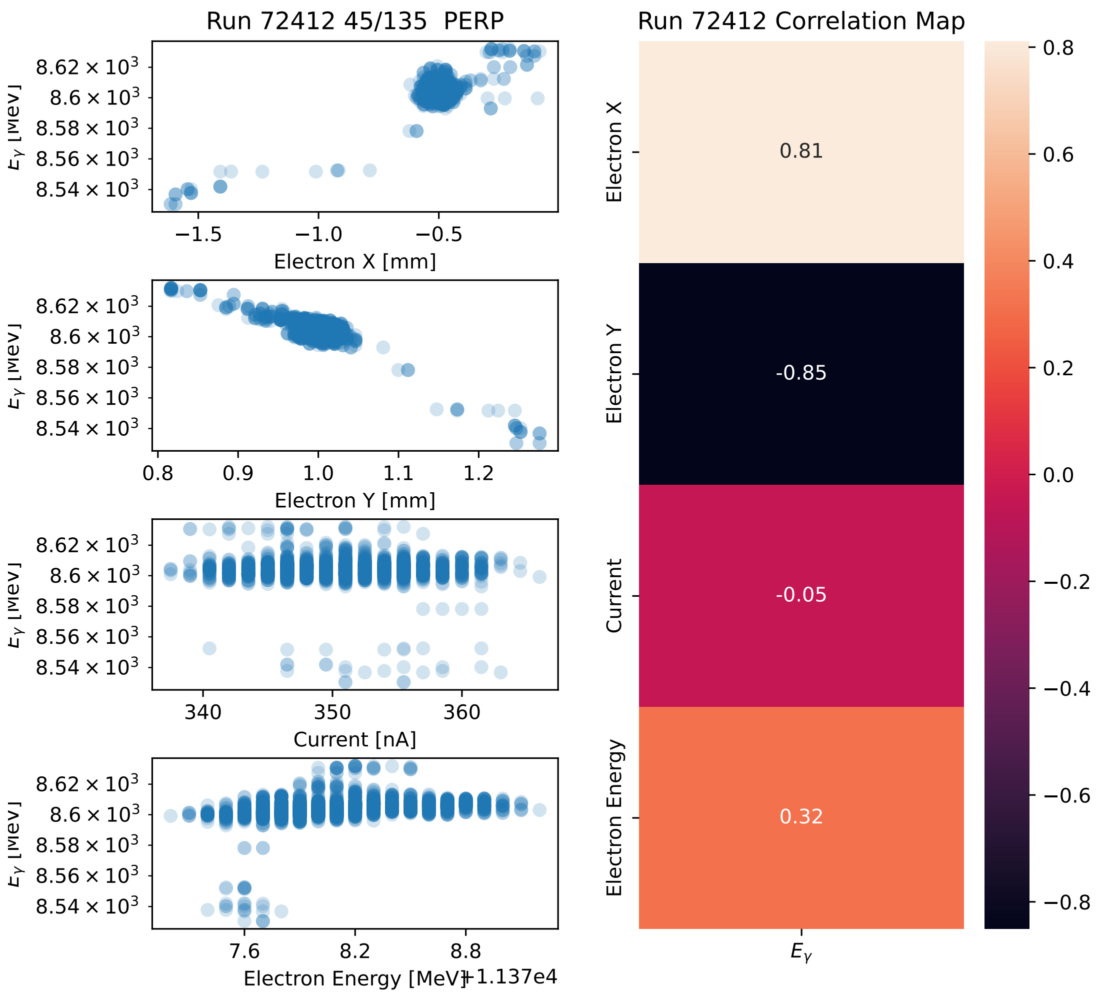

Figure 10: Caption

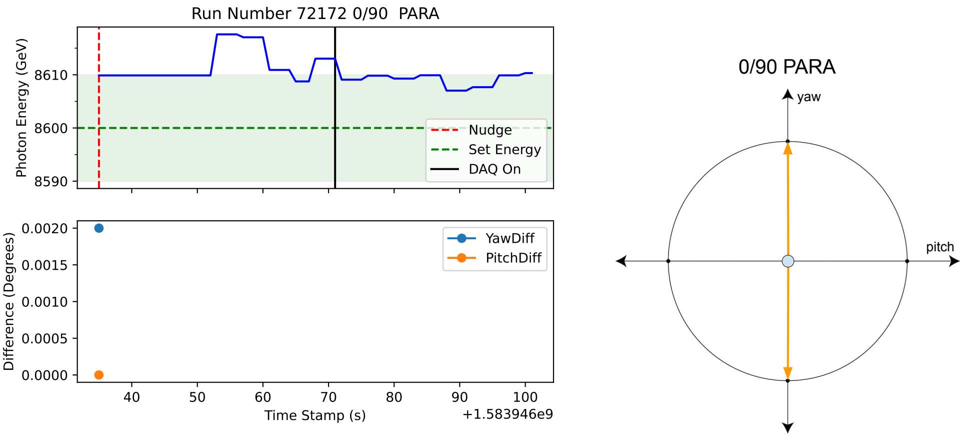

Figure 11: Left: example nudge for the 0/90 PARA configuration. Right: diagram showing nudges only occur in the ± yaw direction for 0/90 PARA.

4 Visualizing and Understanding the Nudges

The code for this section is in singlenudge-study.ipynb

In this section, we visualize how the photon beam coherent edge energy changes in response to nudges. We only use production runs for Spring 2020, after the coherent edge fitting scripts were fixed.

We start by looking at runs that have single nudge events. This allows us to identify

1. the time it takes the beam energy to change after a nudge.

2. how much the energy changes for a single nudge.

There are only 18 runs that have a single nudge. Could expand statistics by looking for well spaced nudges, more easily by looking at Spring 2023 as well.

For each of these single nudge events, we produce a plot that includes the measured photon energy as a function of time (blue), the desired photon energy (dashed green), and the acceptable beam energy range (±10 MeV of desired, green shaded region). For each instant that a nudge occurs, we include a vertical red dashed line. On the lower portion of the plot, we also include how large the change in pitch and yaw was for each nudge. The solid black line indicates the time when the DAQ was turned on for the run.

Nudges can either be small (1 millidegree) or large (10 millidegrees). In addition, since we only record values every second, you can also have integer multiples of these step sizes if somebody hits the nudge button multiple times in one second. The direction in pitch and yaw of the nudge depends on the orientation. We include diagrams of the nudge direction alongside example plots of nudge events. These are shown for 0/90 PARA in Fig. 11, for 0/90 PERP in Fig. 12, for 45/135 PARA in Fig. 13, and for 45/135 PERP in Fig. 14.

We see that for some nudges, the measured beam energy was not being continuously updated, for example, in Fig. 11 and Fig. 12. For other nudge events, like Fig. 13 and Fig. 14, the beam energy measurement was more stable. We see it takes approximately 15 seconds for the beam energy to change, and the change is on the order of 5 MeV for a small nudge.

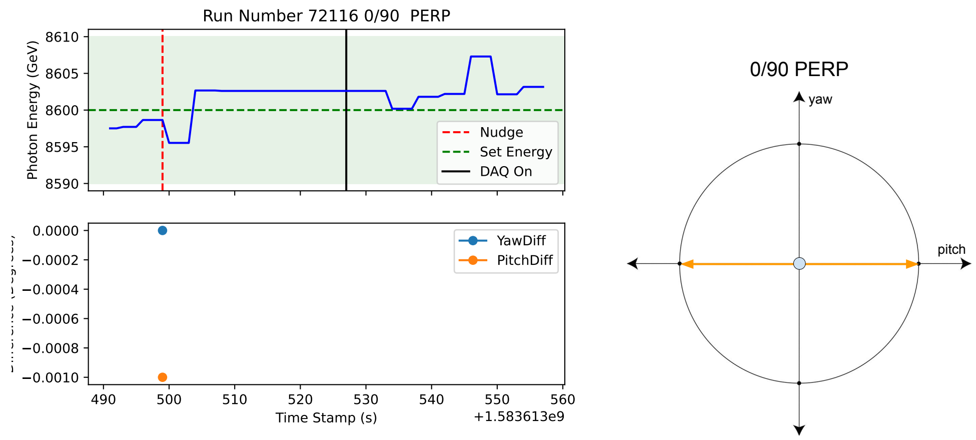

Figure 12: Left: example nudge for the 0/90 PERP configuration. Right: diagram showing nudges only occur in the ± pitch direction for 0/90 PERP.

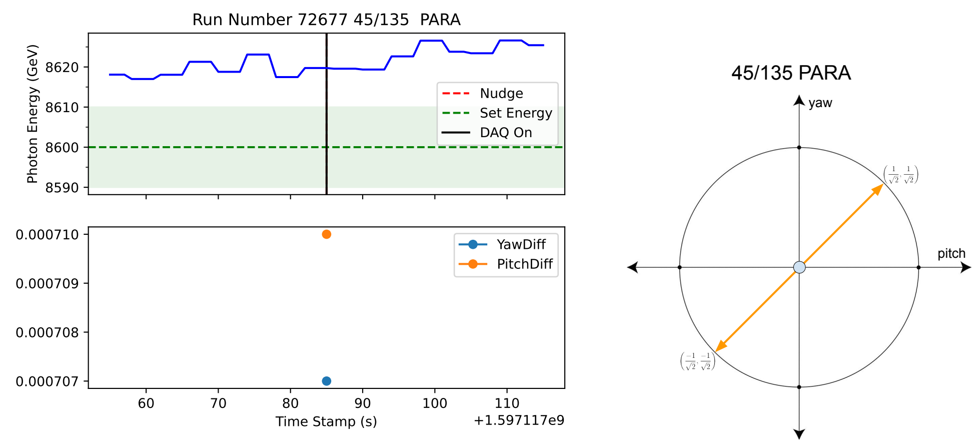

Figure 13: Left: example nudge for the 45/135 PARA configuration. Right: diagram showing nudges change yaw and pitch equally for 45/135 PARA.

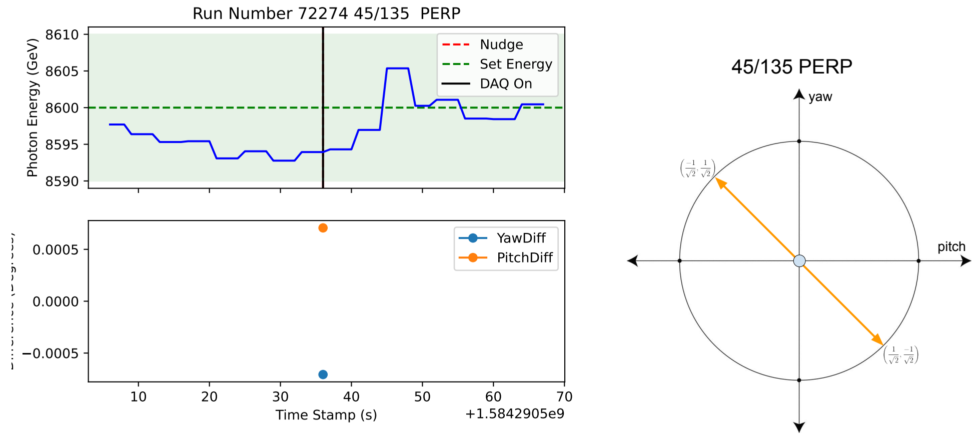

Figure 14: Left: example nudge for the 45/135 PERP configuration. Right: diagram showing nudges change pitch and yaw in opposite directions for 45/135 PERP.

To get a better idea of the energy change per nudge, we look at the change in the coherent edge position 15 and 20 seconds after the nudge occurred. A plot showing the energy change is given in Fig. 15. The statistics are relatively small, but we can see the values above 8 MeV are likely outliers, and the cluster of entries near zero are likely due to unreliable photon beam measurements. Taking the median value of the energy change, we find a 4 MeV change is most typical for a small nudge event. Much more data would be great here. In the next subsection, we look at how the energy change behaves as a function of the number of small nudges.

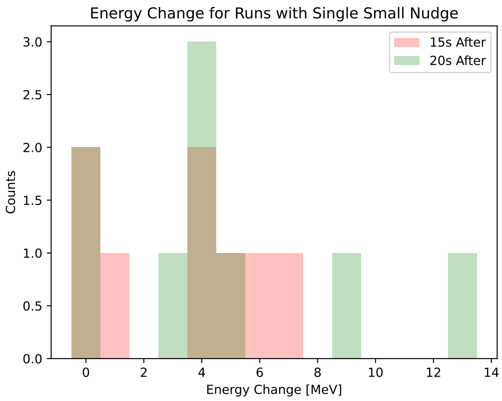

Figure 15: Histogram showing typical energy changes 15 seconds (red) and 20 seconds (green) after a nudge occurred.

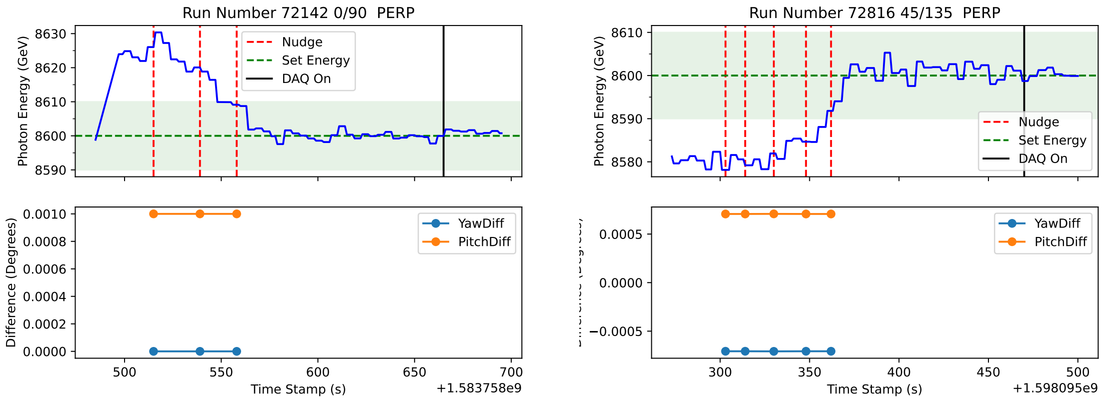

Figure 16: Clean examples of multiple nudges in the 0/90 orientation (left) and 45/135 orientation (right).

4.1 Multiple-Nudge Runs

The estimated energy change per nudge in Fig. 15 is not very reliable. To get a more reliable estimate, we look at runs that have no backlash and multiple nudges. To improve data quality, I also looked through plots of all the runs and picked the ones where there were no signs of backlash (which could exist due to motor motion before the run) and where the beam energy was stable. Examples of clean runs with multiple nudges are shown in Fig. 16 for the 0/90 orientation on the left and the 45/135 orientation on the right.

The plot of 0/90 shows 3 nudges and an energy change of roughly 30 MeV, so we can estimate roughly 10 MeV as the energy change for a single nudge. In the 45/135 configuration, there are 5 nudges, and an energy change of only 20 MeV. This gives a very different energy change per nudge of 4 MeV.

To investigate this further, we measure the difference in the energy of the coherent edge before and after the nudge sequence is finished. The initial energy is taken as the energy when the first nudge occurred. To get a reliable estimate for the ending energy, we look at the time window between 15 and 30 seconds after the final nudge, and take the average energy in that time.

We look at the change in energy as a function of the number of nudges in Fig. 17. Based on our earlier observation that 0/90 and 45/135 appear to have different energy changes per nudge, we split this plot by orientation as well. We fit the corresponding scatterplots with a linear regression model, where we force the y-intercept to be zero, since 0 nudges must cause 0 change in energy. For the 0/90 orientation, we find a slope of around 10 MeV per nudge, consistent with the results above.

For the 45/135 orientation, we get a slope of around 6 MeV per nudge, which is a bit higher than the estimate above. This discrepancy could be due to the fact that the first few nudges in Fig. 16 (right) appear to have no effect, which could be indicative of backlash due to motion before the run started. It looks like the first two nudges are ineffective, so assuming those nudges "don't count", we would get something like 20 MeV/3 nudges = 6.7 MeV per nudge, which is closer to the estimate from the regression fit.

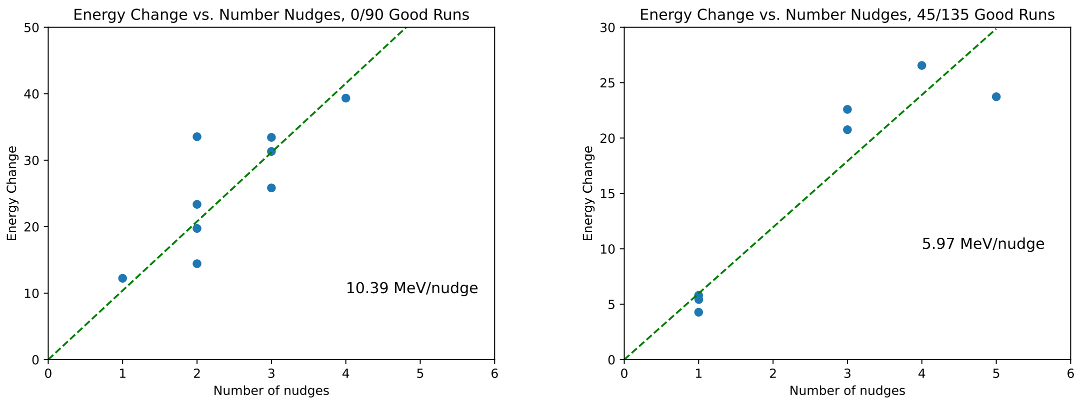

Figure 17: Measured energy changes as a function of the number of nudges in the run.

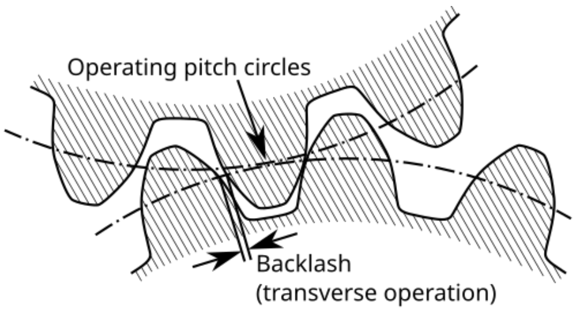

Figure 18: Diagram showing how backlash can appears in a system of gears.

5 Backlash

5.1 Background

5.1.1 Definition

Backlash is a common effect in mechanical systems. As an example, if two gear wheels are fit together, there is likely a small gap between the gears, in other words, they can "wiggle" a bit without putting appreciable force into the system. The amount of free movement allowed is the backlash of the system. The figure I stole off of Wikipedia [5] in Fig. 18 shows the backlash of a system of two gears.

5.1.2 Initial Goniometer Studies

Backlash can impact the goniometer when the motor direction changes, because some of the motion will be lost to backlash. Goniometer studies indicated that there is approximately 2.1 millidegrees of backlash in the angular stages of the goniometer [7]. This corresponds to 2 small nudges if only pitch or yaw are changing, or 3 small nudges if both pitch and yaw change. The expected effects of this are described in the next subsection (so they are easier to find from the table of contents).

5.2 Expected Effects of Backlash

If the motor motion reverses, there is around 2.1 millidegrees of backlash. This means for small nudges

- For the 0/90 orientation, only pitch or yaw changes. Therefore, it takes 2 small nudges to cancel out most of the backlash, and the 3rd small nudge will 10% less effective.
- For 45/135, both pitch and yaw change. There are 2.1 millidegrees of backlash in both directions, and step sizes are $(1/\sqrt{2} \approx 0.7)$ millidegrees in each direction. Therefore, it takes 3 small nudges to cancel out the effect of backlash.

Large nudges will change the coherent edge in the correct direction, but will have a reduced effect:

- For 0/90, the large nudge will be 21% less effective due to backlash.
- For 45/135, the large nudge will be about 30% less effective due to backlash.

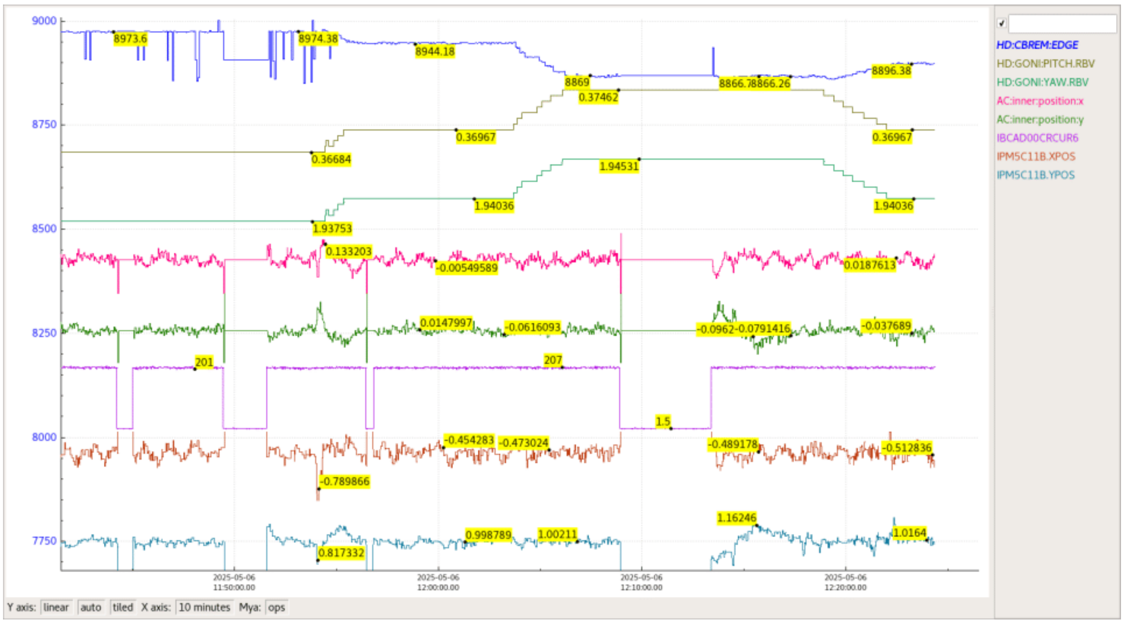

Figure 19: Plot of measured beam energy, pitch, yaw, etc.

5.3 Backlash Observed in 2025 Running

Hovanes noticed an example of backlash during the 2025 data taking, in the logbook entry cited here [6]. In the logbook entry, Hovanes estimates 12 MeV/millidegree as the change in the coherent edge given a nominal beam energy of 8.9 GeV. He assumes roughly 2 millidegrees of backlash, which gives an expected change in energy of around 25 MeV. His calculation is correct assuming only pitch or yaw is changing, but since this is the 45/135 configuration, both pitch and yaw are changing. In this case, the backlash effect is equivalent to three small nudges, so the effect should be closer to 36-40 MeV. The observed energy change was 48 MeV, so the scale is approximately correct, but the measured effect is still bigger than expected. A plot showing the measured beam energy, as well as pitch and yaw readback values is shown in Fig. 19. We see that the coherent edge energy is different for the same readback values in pitch and yaw, which is indicative of backlash in the system.

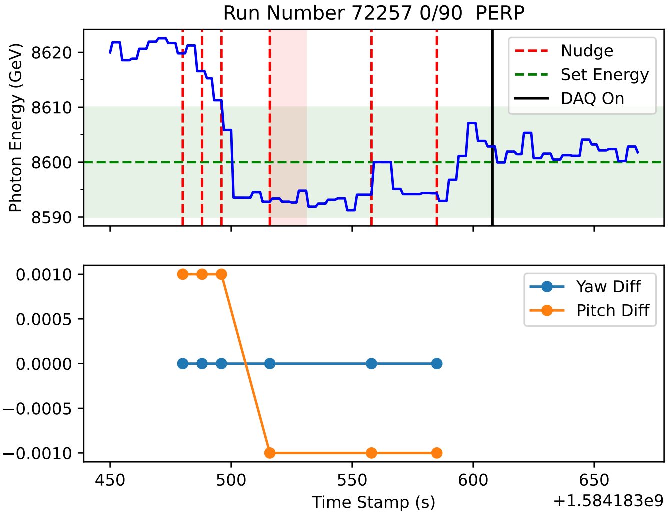

Figure 20: The clearest example of backlash in Spring 2020. Three nudges down followed by three nudges up should result in no energy change, but the energy clearly changes.

5.4 Backlash in Spring 2020 Nudges

In the good runs of Spring 2020 (after the energy fit script was fixed), we have 13 instances of potential backlash, as evidenced by the pitch/yaw setpoints changing directions. Plots for all 13 candidate backlash events in Spring 2020 are in Appendix D. Some of the plots are hard to read because of the large number of nudges, wide time scales, or unstable beam energy conditions. The plots in this section follow the same style as those in the previous section, except they have an additional red shaded region. This region is the 15 seconds after the first backlash nudge, where 15 seconds is the length of time it typically takes for the energy to change for a normal nudge.

5.4.1 Clearest Backlash Example

In Fig. 20, we show the clearest examples of backlash events for the 0/90 orientation. In this example, we see 3 downward nudges followed by 3 upward nudges. We denote this nudge sequence as

$$dddu u u = 0, \tag{1}$$

where the 0 indicates the final pitch and yaw positions will be identical to the starting values. We can see the energy changes from around 8620 MeV to 8600 MeV, which is what we expect if there is backlash. Based on the studies above, we expect the first two nudges to be ineffective, so the "effective" nudge sequence can be written as

$$ddd{\color{red}uu}u = dddu = dd \tag{2}$$

so it is equivalent to just two downward nudges. Estimating the energy change per nudge, we see a 20 MeV change in energy for two "effective" nudges, which gives an energy change per nudge of 10 MeV, consistent with the energy change per nudge measured earlier.

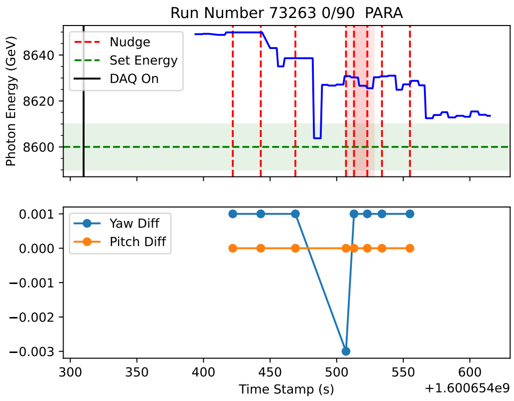

Figure 21: An example where backlash occurs in both directions, effectively canceling out the backlash effect.

5.4.2 Sequential Backlash in Same Run

In this section, we look at a run that has multiple instances of backlash. This is shown in Fig. 21. Note that the upward nudge is 3 small nudges in rapid succession. This gives us a nudge sequence of

$$dddu u u d d d d = dddd, \tag{3}$$

so we estimate a roughly 40 MeV change in energy. Accounting for backlash, we get the "effective" nudge sequence of

$$ddd\textcolor{red}{uu}u\textcolor{red}{dd}dd = dddudd = dddd, \tag{4}$$

which is identical to what we would expect in the absence of backlash. In other words, backlash in opposite directions "cancels each other out".

5.4.3 Additional Backlash Examples

We show an additional backlash plot for the 0/90 orientation in Fig. 22. We see that the first two nudges appear to be ineffective, while the third results in a change in the coherent peak energy. This is exactly the behavior we expect for the 0/90 configuration. The actual nudge sequence of

$$dduuu = u \tag{5}$$

gives an expected energy change of 10 MeV up. Accounting for the backlash, the "effective" nudge sequence is

$$dd\textcolor{red}{uu}u = d \tag{6}$$

which is a 10 MeV downward shift. A 10 MeV downard shift is observed in this case.

We show two examples of backlash events for the 45/135 orientation in Fig. 23.The plot on the left is the only one clean enough to interpret. The initial backlash nudge occurs while the beam is unstable, so we focus on the second nudge. We see a sequence of three nudges, and a very minimal change to the coherent peak position. This is roughly what we expect here, but more examples of backlash in the 45/135 orientation are needed.

Figure 22: Cleanest additional example of backlash for the 0/90 orientation in the Spring 2020 data.

The second plot in Fig. 23 is not clean at all, but shows an interesting change in the photon beam energy. We see that the coherent edge appears to suddenly drop around 40 MeV, and about 30 minutes later, the shift takers appear to notice and aggressively nudge the coherent edge. This is a run that should probably be investigated further, in order to determine what caused the dramatic change in coherent edge energy.

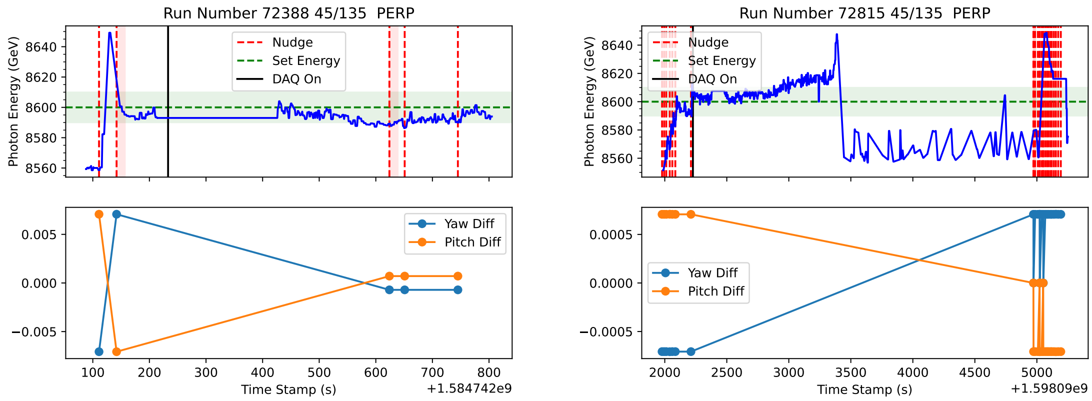

Figure 23: The cleanest examples of backlash for the 45/135 orientation. The bottom right plot isn't clean, but has an interesting change in beam energy that occurs mid-run.

References

[1] AIOP Project Proposal.

[2] mySampler User's Guide. https://halldweb.jlab.org/wiki/images/2/23/MySampler.user.pdf

[3] myget User's Guide https://halldweb.jlab.org/wiki/images/e/e3/Myget.pdf

[4] EPICS Variables and their definitions

https://docs.google.com/spreadsheets/d/1C6lJ3S_5pj_ariDfMVNg5n1iyFbjDV4QYBcpYHo6-90/edit?gid=0#gid=0

[5] Wikipedia https://en.wikipedia.org/wiki/Backlash_(engineering)

[6] Logbook entry of backlash https://logbooks.jlab.org/entry/4370351

[7] Yi's Study of Backlash https://halldweb.jlab.org/wiki/images/5/56/Goniometer_tests-11-2013.pdf

A To Do, Random Facts

A.1 To Do

- Encode nudge sequences in a string, ie "uuUudDu".
- Look at difference in starting and ending angles, as well as start and end energies. Use to get an estimate of energy change per nudge. This will avoid jitter that would be unavoidable using the single nudge data.
- Expand to Spring 2023 data
- Investigate why some nudges are more than 1 degree
- Check run time stamps match before and after the run number fix is done.
- Consider dropping initial rows of NaN's instead of doing backfill during the merge process.
- Check if nudges occur when switching between PARA and PERP

A.2 Random Facts

By comparing the measured roll values to the notes in RCDB, I was able to map the set roll values to the 0/90, 45/135 orientations. I found that for Spring 2020

- Roll of $-10.5^\circ \rightarrow 0/90$
- Roll of $34.5^\circ \rightarrow 45/135$
- CBREM:PLANE of 1 is PARA
- CBREM:PLANE of 2 is PERP
- CBREM:PLANE of 0 indicates switching configs

B myGet Commands For Each Run Period

B.1 Spring 2020

```
myget -m history -c 'HD:coda:daq:run_number' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/run_number.txt
myget -m history -c 'HD:CBREM:PLANE' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/cbrem_plane.txt
myget -m history -c 'HD:coda:daq:status' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/daq_status.txt
myget -m history -c 'HD:GONI:X' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/goni_x.txt
myget -m history -c 'HD:GONI:X.RBV' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/goni_x_rbv.txt
myget -m history -c 'HD:GONI:Y' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/goni_y.txt
myget -m history -c 'HD:GONI:Y.RBV' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/goni_y_rbv.txt
myget -m history -c 'HD:GONI:PITCH' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/goni_pitch.txt
myget -m history -c 'HD:GONI:PITCH.RBV' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/goni_pitch_rbv.txt
myget -m history -c 'HD:GONI:YAW' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/goni_yaw.txt
myget -m history -c 'HD:GONI:YAW.RBV' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/goni_yaw_rbv.txt
myget -m history -c 'HD:GONI:ROLL' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/goni_roll.txt
myget -m history -c 'HD:GONI:ROLL.RBV' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/goni_roll_rbv.txt
myget -m history -c 'HallDACOrb:Suspended' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/beam_lock.txt
myget -m history -c 'HALLD:p' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/ebeam_energy.txt
myget -m history -c 'HD:CBREM:EDGE' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/photonbeam_energy.txt
myget -m history -c 'HD:CBREM:REQ_EDGE' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/photonbeam_energy_desired.txt
myget -m history -c 'HD:CBREM:WIDTH' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/photonbeam_energy_uncertainty.txt
myget -m history -c 'AC:inner:position:x' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/ac_x.txt
myget -m history -c 'AC:inner:position:y' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/ac_y.txt
myget -m history -c 'IPM5C11B.XPOS' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/ebeam_x.txt
myget -m history -c 'IPM5C11B.YPOS' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/ebeam_y.txt
myget -m history -c 'IBCAD00CRCUR6' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/ebeam_current.txt
myget -m history -c 'HD:GONI:RADIATOR_NAME' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/radiator_name.txt
myget -m history -c 'HD:GONI:RADIATOR_ID' -b '2020-01-10 00:00:00' -e '2020-09-21 12:00:00' | tee Spring20/radiator_id.txt
```

B.2 Spring 2023

```
myget -m history -c 'HD:coda:daq:run_number' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/run_number.txt
myget -m history -c 'HD:CBREM:PLANE' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/cbrem_plane.txt
myget -m history -c 'HD:coda:daq:status' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/daq_status.txt
myget -m history -c 'HD:GONI:X' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/goni_x.txt
myget -m history -c 'HD:GONI:X.RBV' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/goni_x_rbv.txt
myget -m history -c 'HD:GONI:Y' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/goni_y.txt
myget -m history -c 'HD:GONI:Y.RBV' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/goni_y_rbv.txt
myget -m history -c 'HD:GONI:PITCH' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/goni_pitch.txt
myget -m history -c 'HD:GONI:PITCH.RBV' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/goni_pitch_rbv.txt
myget -m history -c 'HD:GONI:YAW' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/goni_yaw.txt
myget -m history -c 'HD:GONI:YAW.RBV' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/goni_yaw_rbv.txt
myget -m history -c 'HD:GONI:ROLL' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/goni_roll.txt
myget -m history -c 'HD:GONI:ROLL.RBV' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/goni_roll_rbv.txt
myget -m history -c 'HallDACOrb:Suspended' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/beam_lock.txt
myget -m history -c 'HALLD:p' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/ebeam_energy.txt
myget -m history -c 'HD:CBREM:EDGE' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/photonbeam_energy.txt
myget -m history -c 'HD:CBREM:REQ_EDGE' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/photonbeam_energy_desired.txt
myget -m history -c 'HD:CBREM:WIDTH' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/photonbeam_energy_uncertainty.txt
myget -m history -c 'AC:inner:position:x' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/ac_x.txt
myget -m history -c 'AC:inner:position:y' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/ac_y.txt
myget -m history -c 'IPM5C11B.XPOS' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/ebeam_x.txt
myget -m history -c 'IPM5C11B.YPOS' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/ebeam_y.txt
myget -m history -c 'IBCAD00CRCUR6' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/ebeam_current.txt
myget -m history -c 'HD:GONI:RADIATOR_NAME' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/radiator_name.txt
myget -m history -c 'HD:GONI:RADIATOR_ID' -b '2023-01-21 00:00:00' -e '2023-03-19 22:00:00' | tee Spring23/radiator_id.txt
```

C Spring 2025

```
myget -c 'HD:coda:daq:run_number' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/run_number.txt
myget -c 'HD:CBREM:PLANE' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/cbrem_plane.txt
myget -c 'HD:coda:daq:status' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/daq_status.txt
myget -c 'HD:GONI:X' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/goni_x.txt
myget -c 'HD:GONI:X.RBV' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/goni_x_rbv.txt
myget -c 'HD:GONI:Y' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/goni_y.txt
myget -c 'HD:GONI:Y.RBV' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/goni_y_rbv.txt
myget -c 'HD:GONI:PITCH' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/goni_pitch.txt
```

```
myget -c 'HD:GONI:PITCH.RBV' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/goni_pitch_rbv.txt
myget -c 'HD:GONI:YAW' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/goni_yaw.txt
myget -c 'HD:GONI:YAW.RBV' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00'| tee Spring25/goni_yaw_rbv.txt
myget -c 'HD:GONI:ROLL' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/goni_roll.txt
myget -c 'HD:GONI:ROLL.RBV' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/goni_roll_rbv.txt
myget -c 'HallDACOrb:Suspended' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/beam_lock.txt
myget -c 'HALLD:p' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/ebeam_energy.txt
myget -c 'HD:CBREM:EDGE' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/photonbeam_energy.txt
myget -c 'HD:CBREM:REQ_EDGE' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/photonbeam_energy_desired.txt
myget -c 'HD:CBREM:WIDTH' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/photonbeam_energy_uncertainty.txt
myget -c 'AC:inner:position:x' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/ac_x.txt
myget -c 'AC:inner:position:y' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/ac_y.txt
myget -c 'IPM5C11B.XPOS' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/ebeam_x.txt
myget -c 'IPM5C11B.YPOS' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/ebeam_y.txt
myget -c 'IBCAD00CRCUR6' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/ebeam_current.txt
myget -c 'HD:GONI:RADIATOR_NAME' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/radiator_name.txt
myget -c 'HD:GONI:RADIATOR_ID' -b '2025-05-03 20:00:00' -e '2025-06-15 23:59:00' | tee Spring25/radiator_id.txt
```

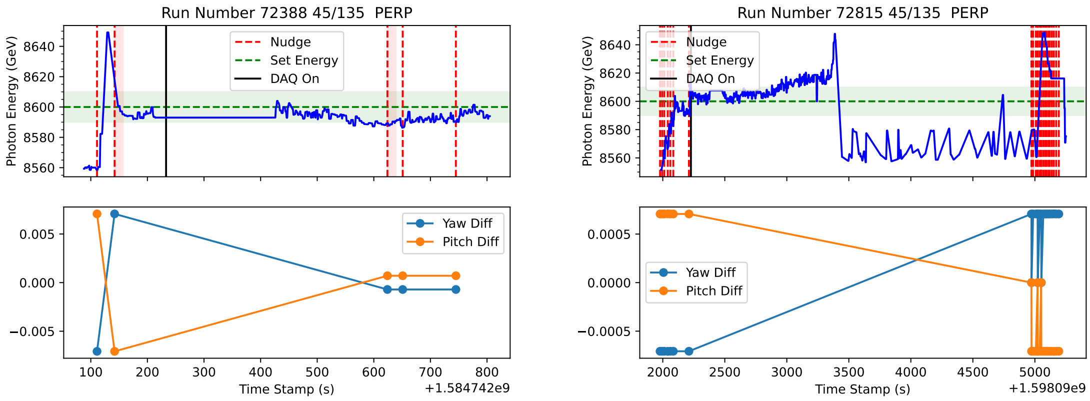

Figure 24: Backlash events for the 45/135 PERP configuration.

D Backlash Plots

In this section, we show all the backlash plots for the Spring 2020 data.

D.1 45/135 PERP Configuration

The backlash events for the 45/135 PERP configuration are shown in Fig. 24. Run 72388 has two instance of backlash. Run 72815 has backlash that occurs while the DAQ is on. Around 3500, the beam energy changes dramatically, with no corresponding changes in the pitch/yaw set points.

D.2 0/90 PARA Configuration

The backlash events for 0/90 PARA are shown in Fig. 25. There are a total of 6 runs with backlash in this configuration. Runs 72102 (top left) and 73263 (bottom right) seem to give the clearest visuals here.

D.3 0/90 PERP Configuration

The backlash events for 0/90 PERP are shown in Fig. 26. There are a total of five runs that have backlash in this configuration. Runs 72127 and 72257 (top row) appear to give the best visuals here.

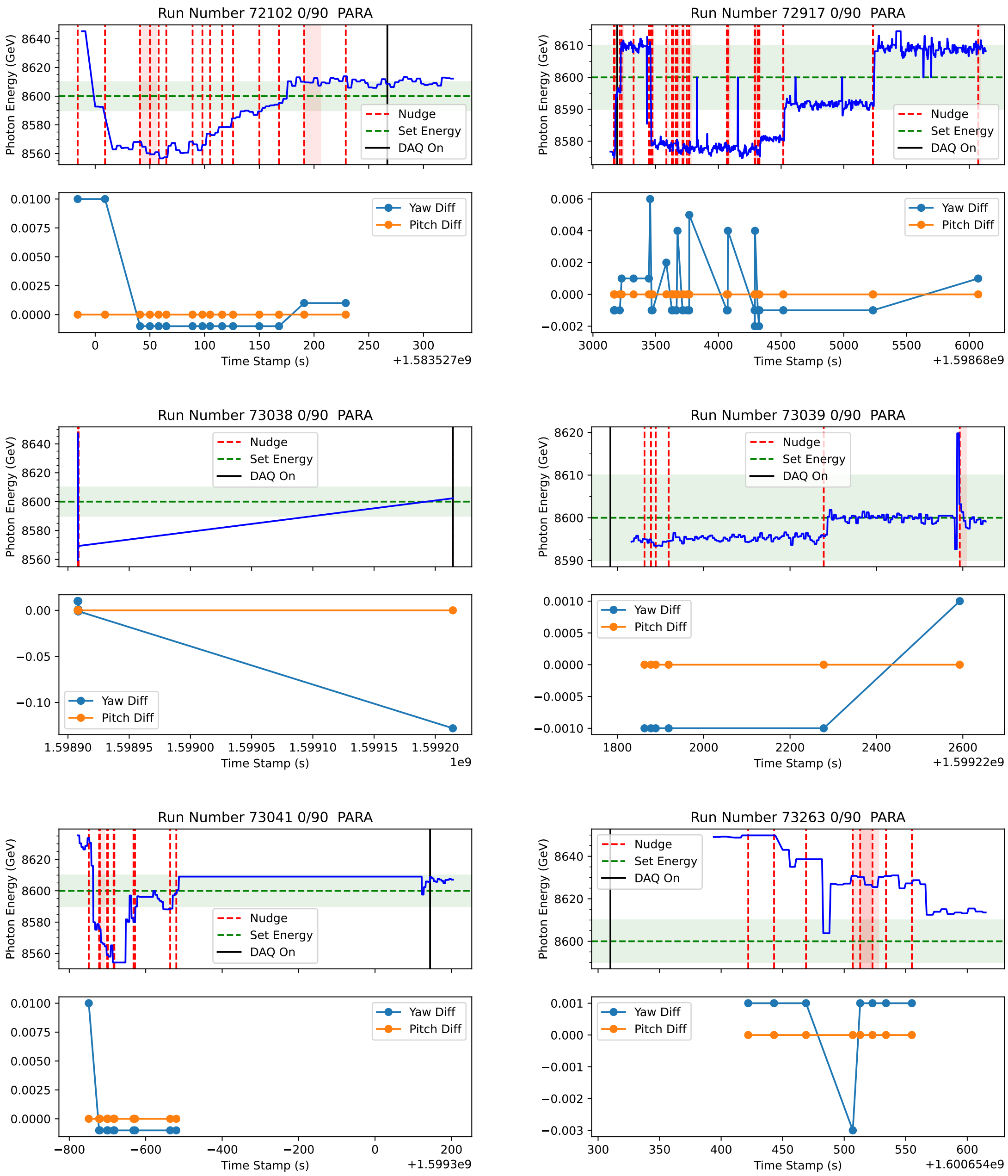

Figure 25: Backlash events for the 0/90 PARA configuration

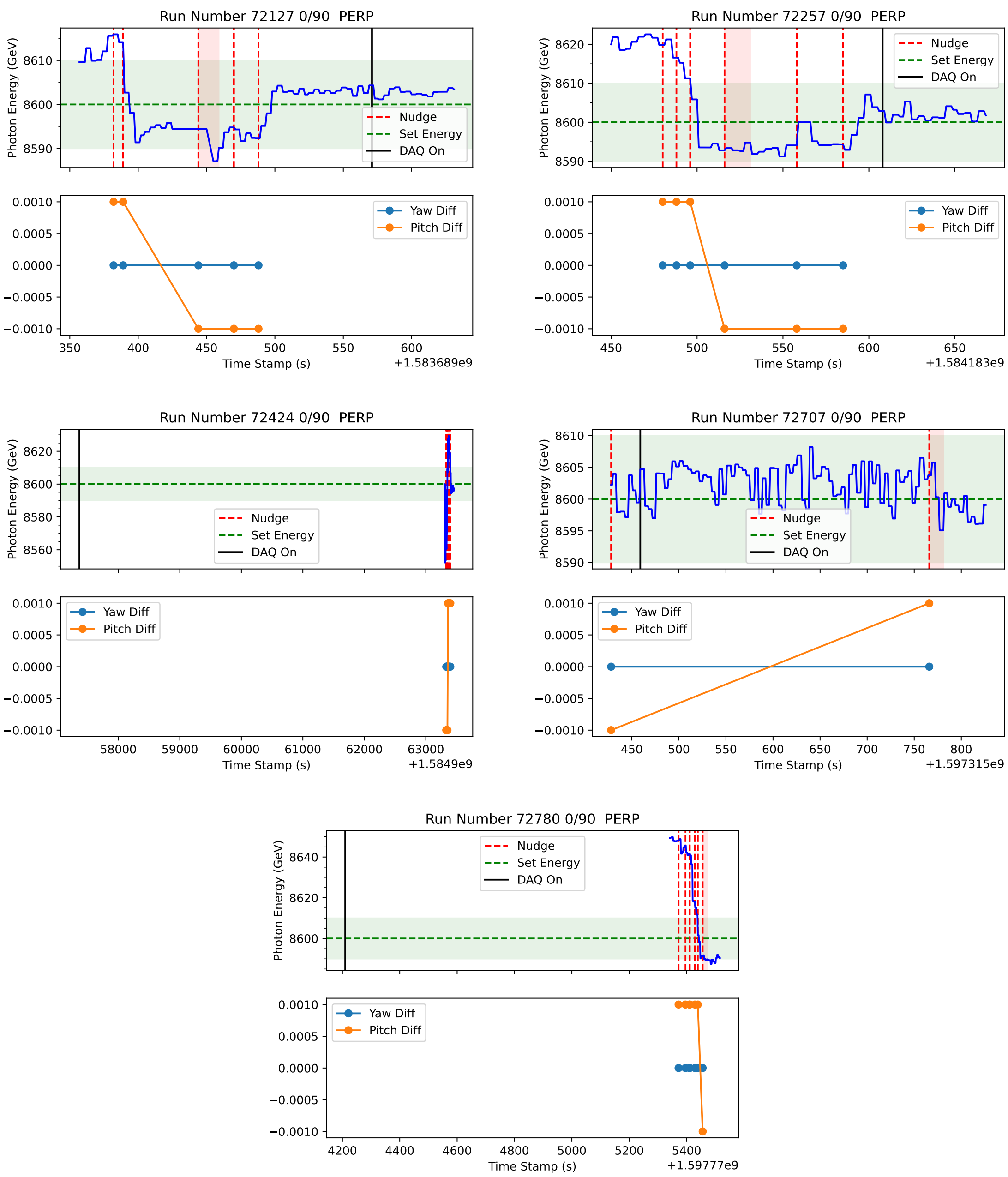

Figure 26: Backlash events for the 0/90 PERP configuration

E Logging Into Gluons

Logging into the gluons is relatively simple. First, login to login.jlab.org as normal. Then, instead of logging into ifarm, you can ssh into one of the gluons. gluon150-155 are for general use.