LNA Dewar Monitoring Project Progress 2

From the last set of experiments, it was noticed that keeping the slicePeriod to one day indeed showed the widest spread of dominant frequencies before the event of maintenance. For the following discussion, a slicePeriod equal to 1 is considered.

It was also noted that the raw measurements of the second stage and the pressure readings were significantly noisy. So before applying FFT on these signals, it was decided to use Kalman filter to better estimate the true values of these signals.

Kalman Filter

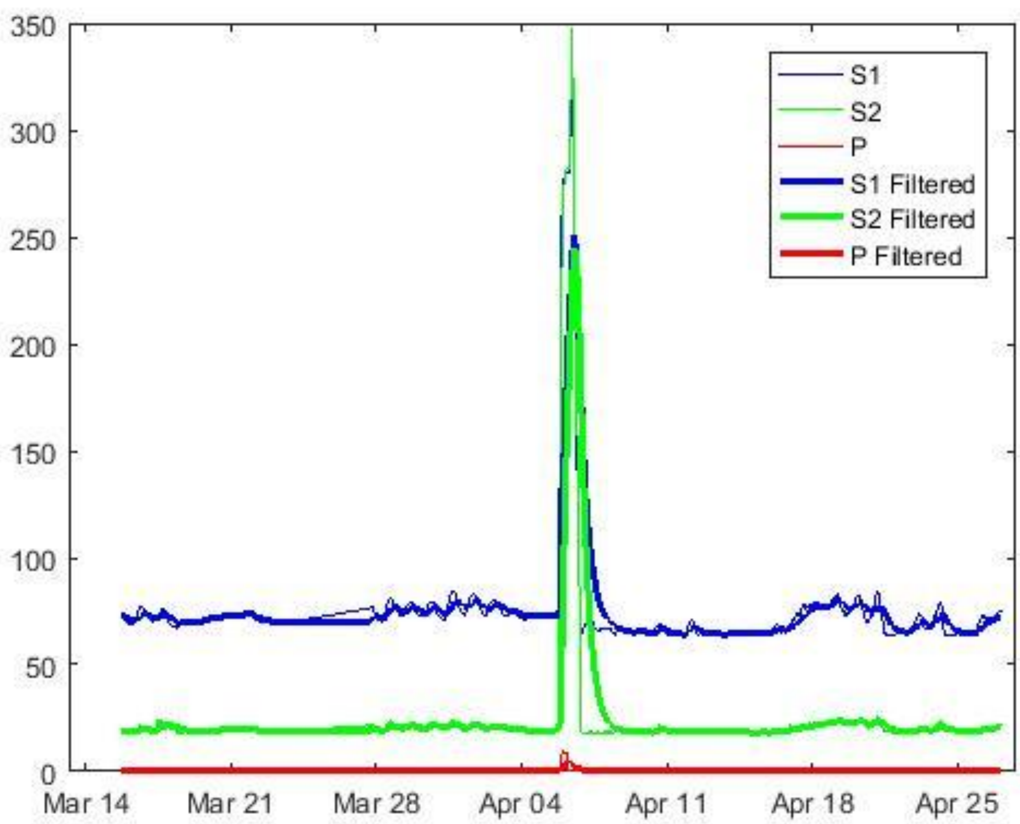
For applying the Kalman filter, it is desired that the process noise variance and the measurement noise variance is known beforehand. While there are some complex algorithms that try to estimate these quantities from the measurements themselves, I have for now assumed the values.

The process noise variance is defined as var_P and measurement noise variance is defined as var_M.

The assumptions for the first tests were:

```
var_P = 0.0000001;
var_M = 0.5;
```

which produced the result:



The effect of decreasing var_P is reduction in over fitting of the estimation curve onto the original measurements curve while decreasing var_M overfits the estimation onto with the original one.

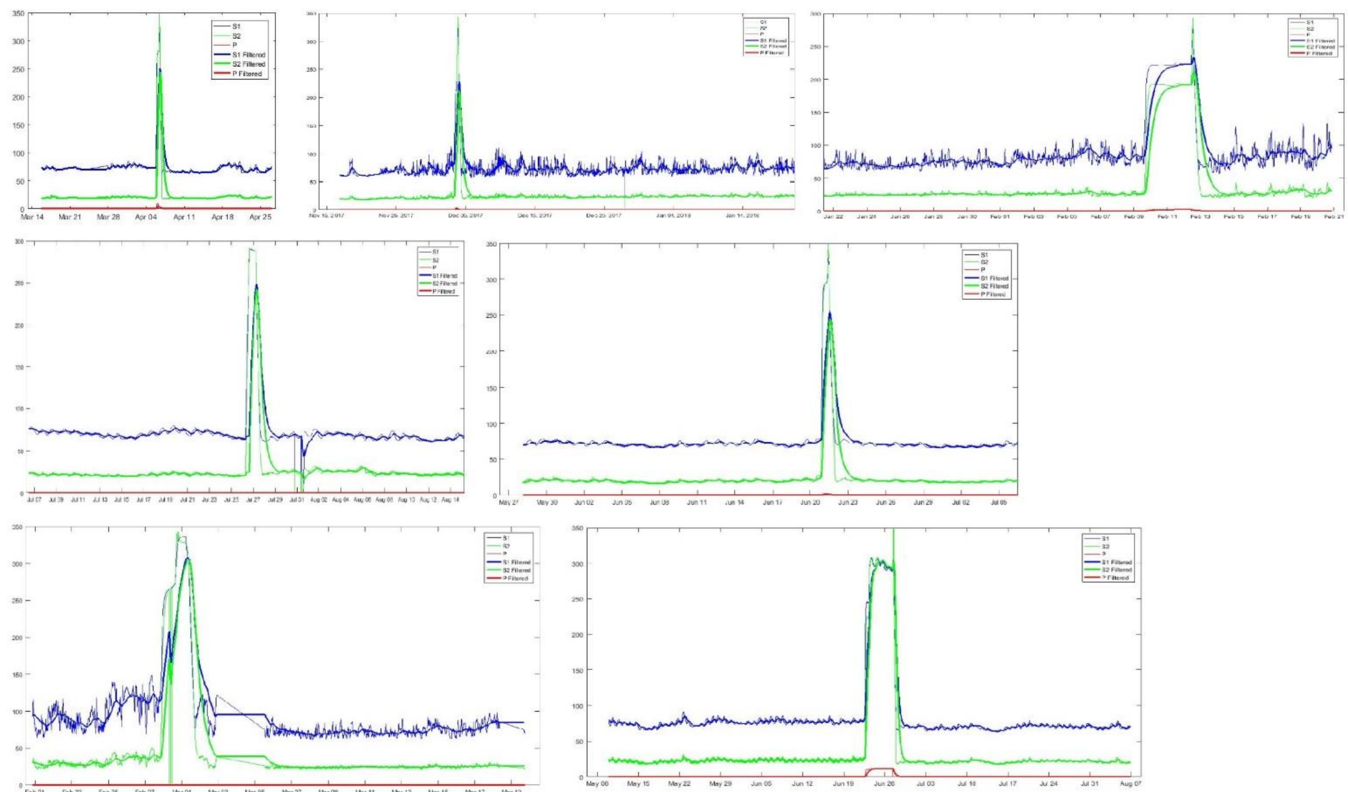
Therefore, at hand both these variables are tuning parameters that can be tuned to get a specific filtered output of the measurement curve.

Short-Time Fourier Transform

I have tried to develop a custom function for calculating the Short Time Fourier Transform of the filtered signals. Previously, the function divided the signal into numOfTotalSlices but the slices did not mutually overlap with each other. As part of the generally used STFT algorithm, the slices should have an overlap and should be truncated using a non-rectangular windowing function.

In view of this, I introduced a variable overLap that gives me control over the percentage of overlap I need between successive slices. Also I multiply the Hann window function with each slice.

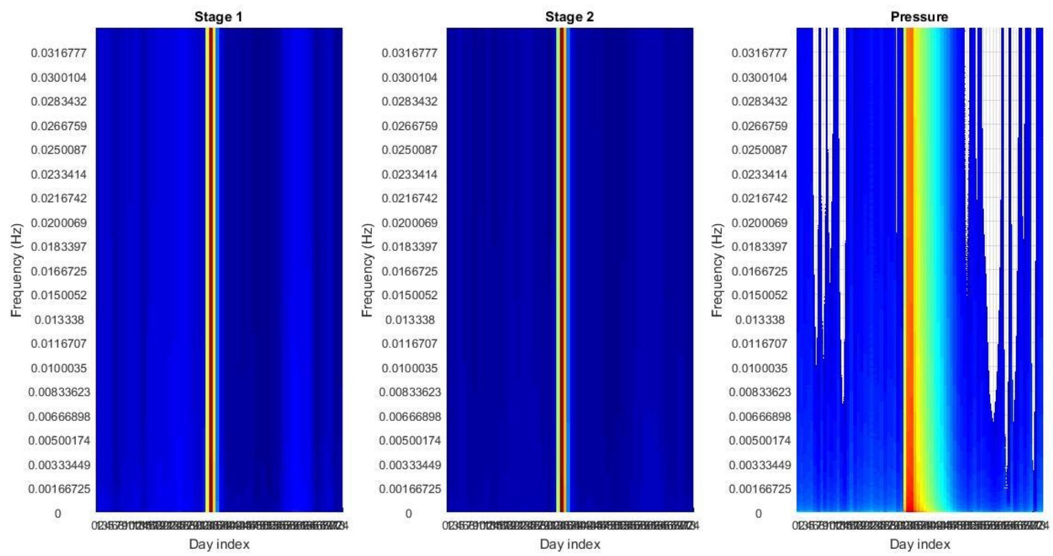
From considering different parts of the text files that you shared, I have extracted uptil now about 7 maintenance events.



Besides the daily variations in the readings (especially in temperature readings, pressure above is not shown to scale), the most apparent observation is that the mean of the signal (especially for Stage 1 temperature curve) rises before the maintenance event. This can be seen visually also in majority of the plots however there are some exceptions also.

Assuming it to be a general case i.e the mean of the signals rise before a maintenance event and fall after the event, I have also scaled the values of each of the truncated slice with its mean before applying FFT on it.

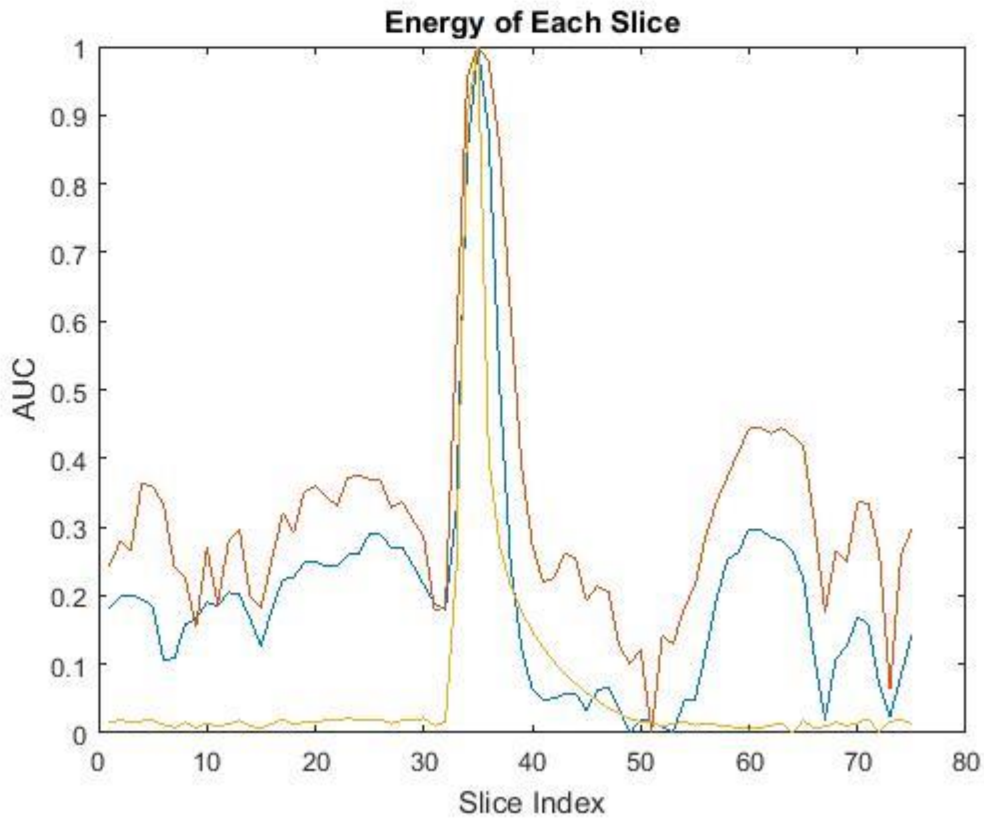
To put simply this should imply that for the same dominant frequencies before and after the event (which means that there is no significant change before and after the event in frequency domain), the part of spectrum before the event should be scaled up in magnitude if the mean of that part is higher in time domain.



Area Under Curve

The generated spectrogram gives a representation of change in magnitudes of different frequencies over progressing time. To get the energy content of each slice, I integrated the FFT result for that slice.

So as a result for all slices, I get a vector with length equal to numOfTotalSlices. Each element of this vector gives the integral (area under curve) result for the slice it corresponds to.

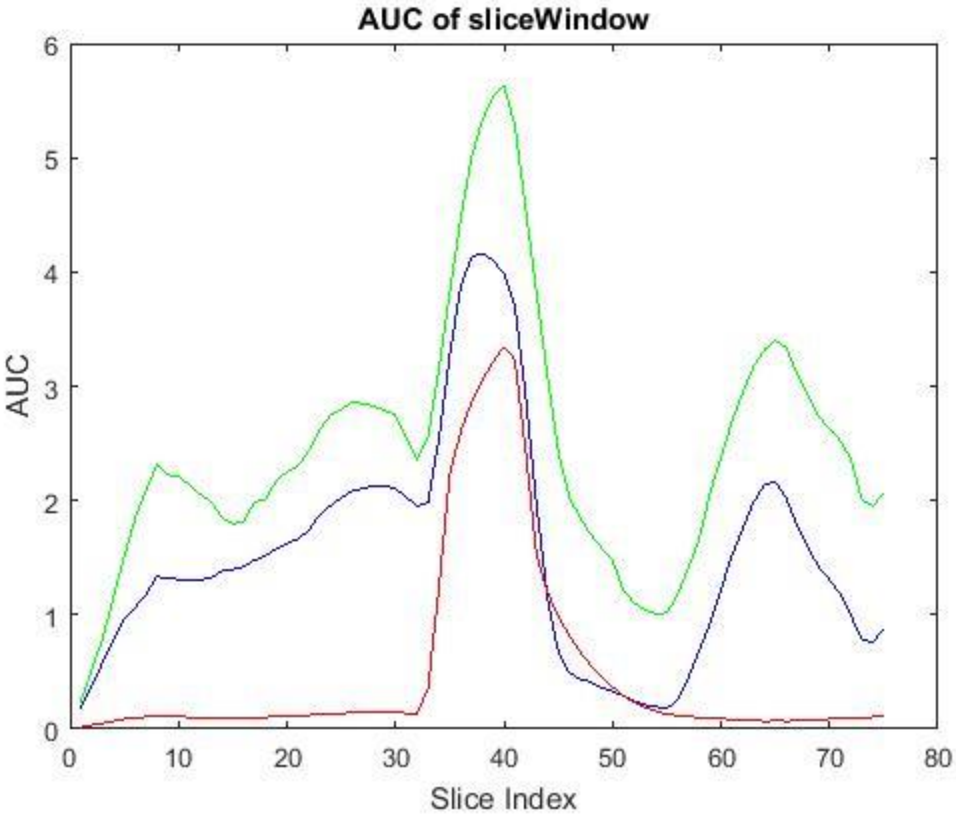


The plot above provides the energy content of each slice. It can be observed that the energy of slices before the event is higher than of the ones come immediately after the event. However, it can be the case that some slices that come after the event have high energy due to various reasons but the number of these consecutive slices should be less which means that we may observe some narrow peaks after the event but not wide ones!

Integral Along Time Domain of AUC Curves

The idea is that the AUC is higher for more slices before the maintenance event than for slices after the event. So if we calculate the integral for some sliceWindowSize, starting from first slice and moving the window till the last slice, we can get the area under different portions of the AUC curve. We can expect the area to be higher before the event and smaller after the event.

I selected a sliceWindowSize to be 7 (units are slices) and moved this window along the slice axis. This gave the ‘AUC of sliceWindow Curves’ which are as:

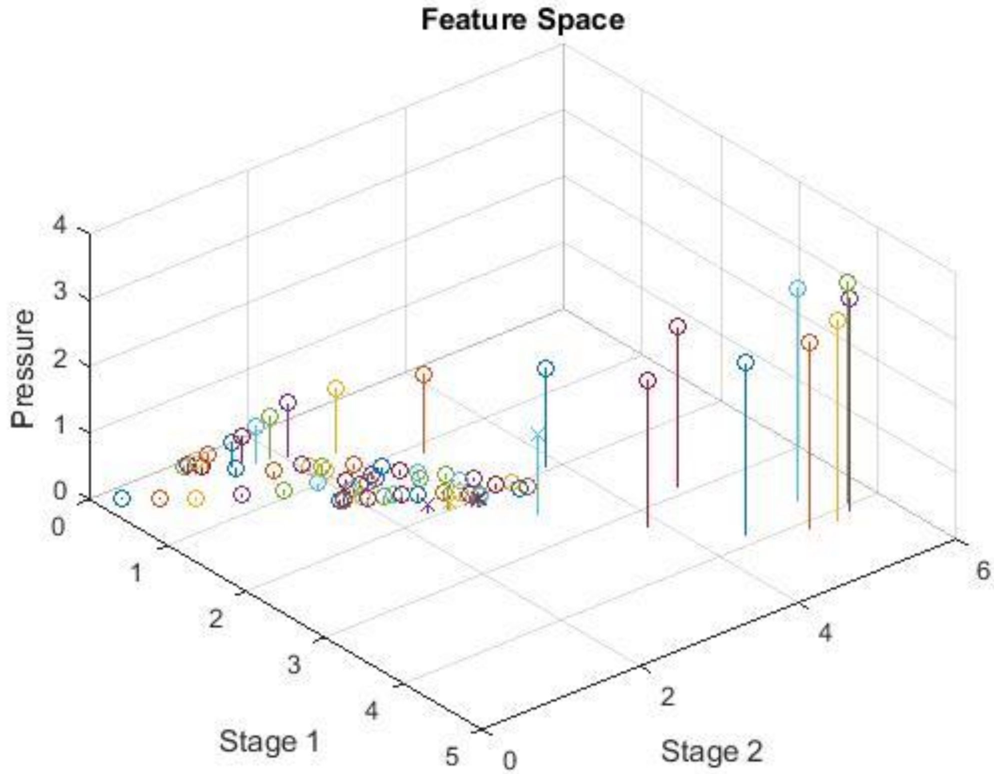


We can observe that for the temperature curves at least, the AUC of sliceWindow drops after the event. Having said this, we can also observe another rise after the drop which does not necessarily correspond to a maintenance event. The second peak after the maintenance event has to be filtered somehow. I am working on this.

Transformation to Feature Space and Decision Making

I think that the 'AUC of sliceWindow Curves' can be helpful in determining if a system is going towards a maintenance activity. What I propose is that we can consider the values of these the 'AUC of sliceWindow Curves' as features of our system that we can plot together for both the temperatures and pressure using scatter plot.

This will transform our analysis to the feature space. Something like this:

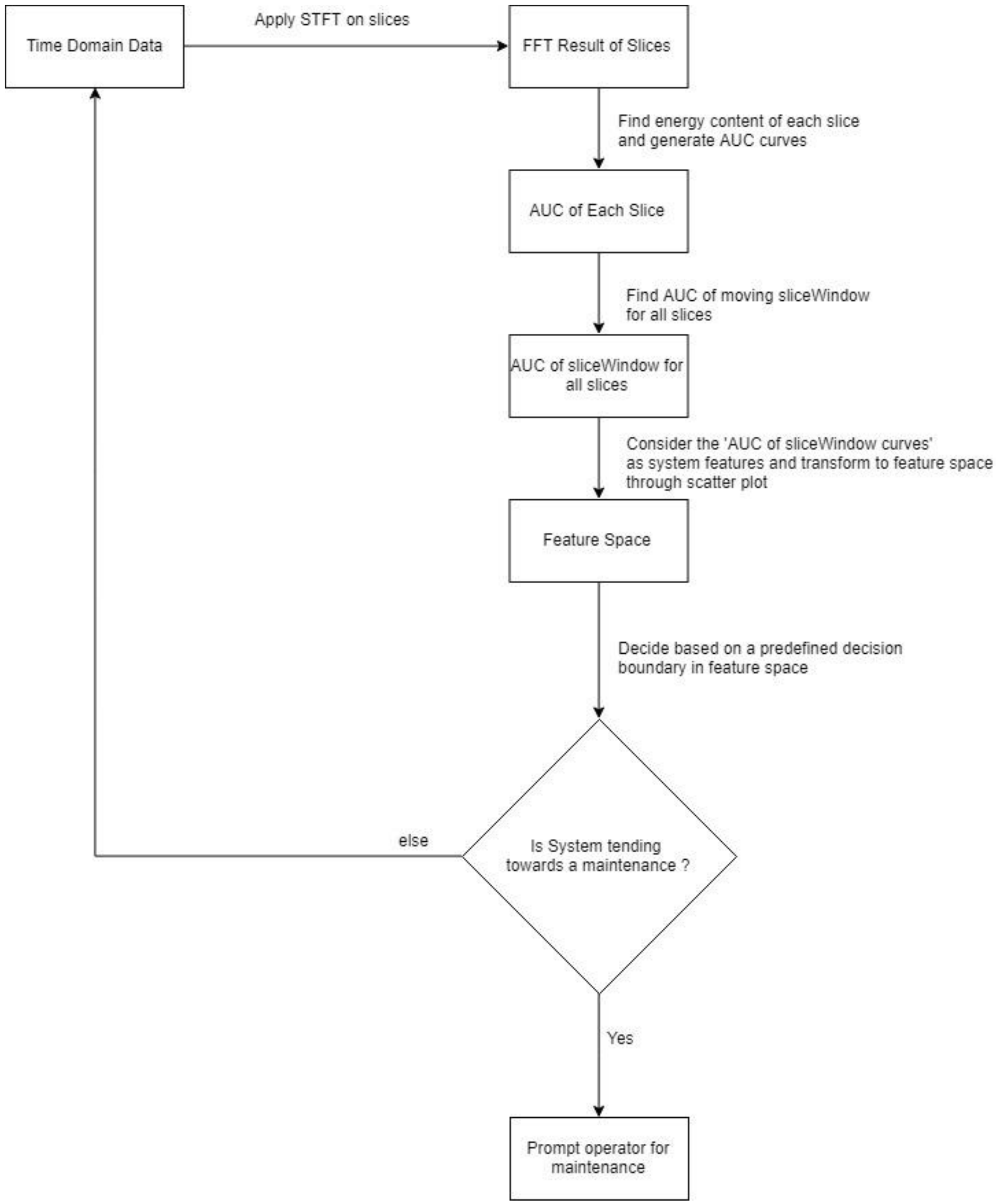


If we are able to identify a distinguishable pattern in this feature space for the slices that come before the maintenance and for the ones that come after the maintenance, we can draw a decision boundary using some regression technique and then use this boundary as a discriminator for decision making.

First Impression of Algorithm

Based on the discussion and results above, I want to share a very basic impression of the algorithm that I am thinking of for this project.

The algorithm flow is as:



Key-Observations

- The exercise uptil now has been done to identify a concrete distinguishable pattern of some values before the maintenance and after the maintenance. Although there are some test cases which show a distinguishable behavior in the feature space, I have not been able to find a general pattern applicable to all or most of the cases. This is something I am working on.
- I think my STFT function needs to perform better and for this I'll am going to see other alternatives that I can get for STFT codes.
- Also, uptil now there are about 5 parameters that can be tuned, namely:

  slicePeriod
  overlap (overlap allowed between slices for STFT)
  var_P (process noise variance for Kalman filter)
  var_M (measurement noise variance for Kalman filter)
  windowSize (Determines the window size for AUC calculation)

- Tuning some of the parameters for one case makes the algorithm better for one case but bad for the other. I am still evaluating the appropriate methods to make things more general.

Please review the results and provide your valuable suggestions and feedback. Thanks!