**Report Title:**

Comparison test of original Precision Acoustics LabVIEW software to new 8Fold Manufacturing python software on existing wet test fixture hardware.

**Contributions:**

Testing, Isaiah Baker, Daniel Murcia

Report preparation, Isaiah Baker

**Client:**

Profound Medical Inc, 2400 Skymark Ave #6, Mississauga, ON L4W 5K5, Canada

**Date:**

July 26th, 2022

In February 2022 8fold Manufacturing was contracted by Profound Medical to create a new graphical user interface for their existing and future Wet Test Fixture (WTF) systems. The goal was to transition from a closed-source LabView interface to a modular, easy-to-maintain, and well documented Python application that can be improved and kept up to date by either 8fold or Profound in the future.

Several months after the project was contracted it was expanded to include the construction of an additional WTF system, using a combination of parts provided by Profound and parts purchased by 8Fold. The system includes a new motor controller assembly, featuring a Galil 4123 controller. This decision was made because the Parker VIX 250 IM Drive controllers are end of life and use obsolete software. In addition, testing with them showed that communication with them is slow and unreliable by comparison. The new motor controller assembly is fully backwards compatible with the existing systems. Pending complete validation of the new software and new motion controller system, existing WTF systems will be able to be seamlessly upgraded at Profound’s discretion, making them more futureproof, reliable, and easy to maintain.

Prior to this report’s preparation the software was tested on the newly finished WTF system in Dallas, and preliminary data comparing the results from the new hardware and software was collected. One of the recommended next steps from that report was to conduct tests on a production Wet Test Fixture with both the new Python software and original LabVIEW software, to determine if there are any differences in how the software itself collects and analyzes data.

The goal of this report is to conduct tests using the same ultrasound applicators (UAs) and the same Wet Test Fixture on both the original LabVIEW software and the new Python software, and comment on whether the differences in their results are acceptable compared to the intrinsic variability between tests on existing hardware and software.

**Devices Tested**

* HC0897
* GM0591
* GM0912
* HB0398
* HB0670
* HC0893
* HB0381
* GK1641

**Test Groups**

Three sets of data were compared. Original wet test data when available (obtained with LabVIEW on WTF3), data from a recent retest (obtained with LabVIEW on WTF2), and finally the data obtained with the new software under test (obtained with Python on WTF2).

The purpose of the inclusion of original test data in the analysis is to provide a point of comparison for the differences that are typical when retesting a UA on the LabVIEW software. Retests with LabVIEW are the control and retests with Python are the experimental group.

**Elements Analyzed**

From the 8 UAs tested, all elements were analyzed with the following exceptions:

1. The test was terminated prior to that element due to test failure
2. If the data from the LabVIEW test showed clear irregularities (Element 1 of GK1641)
3. The element showed no acoustic power (Element 1 of GM0912)

In total, the number of analyzable elements tested the LabVIEW software and Python software was 47. The comparison between the original LabVIEW test and the LabVIEW retest was more limited because not every device’s data was available, with a total of 21 elements compared.

The following measurements were extracted from the test result summary files.

* **Beam Angle (deg)**
* **Low Frequency Efficiency (LF.Eff) (%)**
* **Low Frequency Reflected Power (LF.Rfl) (%)**
* **Low Frequency Forward Power Max (LF.Pf (max)) (W)**
* **High Frequency Efficiency (HF.Eff) (%)**
* **High Frequency Reflected Power (HF.Rfl) (%)**
* **High Frequency Forward Power Max (HF.Pf (max)) (W)**

When comparing two test groups, the following calculations were made using the data from all elements for a specific measurement.

**Difference Between Means**

This is calculated by taking the average of a certain measurement across all elements for each of the two test groups and subtracting them. It is intended to quantify systematic error, or the tendency for the measurements of one dataset to be greater or less than the other dataset. Closer to zero is better since a consistent error in a certain direction indicates a consistent difference in how the data was collected or processed.

**Measurement Differences**

This simply refers to the difference between one dataset’s value for a specific measurement of a specific element and the other’s. It is expressed in the same units as the measurement itself.

**STDev of Differences**

This is calculated by taking the standard deviation (STDEV.S in Excel) of the measurement differences across all the elements in a dataset. It is intended to quantify random error, or the tendency for the differences to be spread out rather than off by a specific amount. One property of the standard deviation is that roughly 68% of the errors will fall within one standard deviation of the mean, which here means that if you were to correct for systematic error, 68% of the data would have an error of less than the STDev of Errors, with no tendency in a particular direction.

**Methods**

8 UAs were tested using WTF2 2 with the original LabVIEW software using the Standard Test Script Version 1.2. According to these tests, 3 of the UAs passed and 4 failed. One was excluded from the analysis because it was later discovered that the results were invalid (likely due to interference from the strings suspending the absorber).

**Results Summary**

To provide a baseline for the repeatability of the results between different tests of the same UAs, the results for 3 of these UAs were compared to their original test data on WTF 3.

Note that each of these original tests were conducted on a different date, ranging from about 1 month to 1 year prior to this report.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Measurement** | **Mean (Retest)** | **Mean (Original)** | **STDev (Retest)** | **STDev (Original)** | **Difference of Means** | **STDev of Differences** |
| Beam Angle (deg) | -88.93 | -91.80 | 1.34 | 2.75 | 2.87 | 1.56 |
| LF.Eff (%) | 65.19 | 67.95 | 10.08 | 5.66 | -2.76 | 7.39 |
| LF.Rfl (%) | 6.66 | 4.43 | 7.08 | 5.60 | 2.23 | 4.79 |
| LF.Pf (max) (W) | 6.81 | 6.20 | 1.58 | 0.45 | 0.61 | 1.54 |
| HF.Eff (%) | 40.65 | 39.95 | 5.44 | 5.25 | 0.70 | 1.59 |
| HF.Rfl (%) | 4.40 | 4.48 | 3.14 | 2.44 | -0.07 | 1.99 |
| HF.Pf (max) (W) | 5.28 | 5.34 | 0.93 | 0.81 | -0.06 | 0.33 |

The results are compared in the table below.

Table Comparison of Retest Results From WTF2 and Original Results from WTF3

**Discussion**

The goal of comparing a sample of this dataset to the initial testing of these UAs on a different production Wet Test Fixture (WTF3) but with the same software was to provide a reference for the variability that can be expected with the current software and hardware used currently for testing production UAs. These differences may include hardware factors, day-to-day variability in the UAs themselves, and variability in test conditions.

In the following section, the results on Wet Test Fixture 2 with the new Python software will be compared to the retest results on Wet Test Fixture 2 with the LabVIEW software, using this comparison data as a benchmark for the variability that is to be expected.

**Methods**

Approximately a week later, the same 7 UAs were tested using the same script with the new Python-based software developed by 8Fold Manufacturing. The results for these 7 UAs were analyzed in the same manner as described in the previous section and compared to the retest results for the same 7 UAs done with the LabVIEW software.

Once again the two metrics being analyzed are the difference of means, which quantifies systematic error, and the STDev of differences, which quantifies random error.

**Results Summary**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Measurement** | **Mean (LabVIEW)** | **Mean (Python)** | **STDev (LabVIEW)** | **STDev (Python)** | **Difference of Means** | **STDev of Differences** |
| Beam Angle (deg) | -90.47 | -90.49 | 2.13 | 1.91 | 0.03 | 1.25 |
| LF.Eff (%) | 64.40 | 64.83 | 11.54 | 8.83 | -0.43 | 5.98 |
| LF.Rfl (%) | 9.37 | 10.61 | 6.88 | 6.85 | -1.23 | 5.15 |
| LF.Pf (max) (W) | 7.30 | 7.20 | 2.56 | 2.17 | 0.10 | 1.06 |
| HF.Eff (%) | 39.89 | 38.94 | 7.47 | 7.02 | 0.95 | 2.84 |
| HF.Rfl (%) | 3.57 | 3.33 | 2.70 | 2.60 | 0.24 | 0.91 |
| HF.Pf (max) (W) | 5.50 | 5.39 | 1.52 | 1.51 | 0.11 | 0.70 |

Table Comparison of results From WTF2 with LabVIEW software and Python software

**Notable Element Case Studies**

Element 1 of GM0912 failed its LabVIEW test due to the LF.Pf (max) (W) exceeding the limit (13.4W > 12W). With the Python software, this quantity (10.7 W) did not exceed the limit, despite being close. Upon retesting, the result was closer to failing, but still passed by a narrow margin (11.3 W). The difference between this result and the initial result (2.1 W) is greater than the average error for LF.Pf (max) (1.06 W), but is not the greatest error observed. This element is borderline according to this pass/fail criterion and as such may have tested one way or the other regardless of the software used.

Element 6 of HB0398 failed its LabVIEW test and failed its Python test, though despite the initial test showing 40% LF efficiency, the Python test showed no power from this element whatsoever. Presumably this element went from underperforming to completely dead, or it has an intermittent connection. It was excluded from the analysis because it did not have a valid Pf (max) and the lack of any acoustic power delivered by the element cannot be attributed to the software.

**Discussion**

The first question worth asking when comparing the results from the existing LabVIEW application to the newly developed Python application is whether there is a systematic difference between them, or in other words, if the average results with the new software tend to be greater or less than the current software. If there was any meaningful difference in the calculations performed, the accuracy of locating the elements, or the ability to isolate the intervals of stable data where the UA is on/off, one would expect that there would be a tendency for the results from the new software to be different on average. As it turns out, they are remarkably close. The average beam angle is virtually identical, with both applications agreeing within a tenth of a degree. The LF eff (%) and HF eff (%) are also very similar, with systematic errors of less than .5% and 1% respectively. The Pf (max) for each frequency range agrees within about 0.1 Watts. The greatest systematic error observed was the Lf.Rfl (%), which still only differed by 1.25%. This clearly shows that the Python software does not tend to provide significantly higher or lower results on average. Furthermore, the standard deviations of each measurement across all elements are very comparable and in fact they are slightly less than the current software in every case, indicating that there is not a greater propensity towards random error.

The differences observed between the results of the new software and the current software can be attributed almost fully to random error, with little if any tendency towards producing higher or lower measurements. The magnitude of this random error depends on the measurement, with the greatest being LF efficiency percent. Still, the STDev of error is less than 6% on average. This means that 68% of the data agrees with the measurement from LabView within 6%. This difference may seem substantial and, in some cases, can be the difference between a UA passing and failing, but in the next section we will discuss whether these differences are acceptable in the context of the variability of the currently used software and hardware.

**Methods**

This section will compare the differences observed between the results obtained with the new Python software and the existing LabVIEW software to the differences observed between the retests with LabVIEW and their original tests. The purpose of this comparison is to investigate whether the differences between testing done with the Python software and testing done with the LabVIEW software are comparable to the differences observed between multiple tests with the same hardware. Once again, the two metrics being analyzed are the difference of means, which quantifies systematic error, and the STDev of differences, which quantifies random error.

**Results Summary**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Measurement** | **Difference of Means (Original/Retest)** | **Difference of Means (LabVIEW/Python)** | | **Better/Worse** | | **STDev of Differences (Original/Retest)** | **STDev of Differences (LabVIEW/Python)** | **Better/Worse** |
| Beam Angle (deg) | 2.87 | 0.03 | + | | 1.56 | | 1.25 | + |
| LF.Eff (%) | -2.76 | -0.43 | + | | 7.39 | | 5.98 | + |
| LF.Rfl (%) | 2.23 | -1.23 | + | | 4.79 | | 5.15 | - |
| LF.Pf (max) (W) | 0.61 | 0.10 | + | | 1.54 | | 1.06 | + |
| HF.Eff (%) | 0.70 | 0.95 | - | | 1.59 | | 2.84 | - |
| HF.Rfl (%) | -0.07 | 0.24 | + | | 1.99 | | 0.91 | + |
| HF.Pf (max) (W) | -0.06 | 0.11 | + | | 0.33 | | 0.70 | - |

Table Comparison of Variability between LabVIEW and Python to Variability between WTF2 and WTF3

**Discussion**

The systematic error between results from Python and LabVIEW was less than the systematic error between the retest and the original test in every measurement except one. The random error is more comparable, but in general is still less for the Python software. This indicates that the random error observed with the Python software is well within the acceptable range. It is plausible that most of this is attributable to the day-to-day variability in the UAs themselves.

These tests suggest that there is extremely little systematic difference in the results of the new Python Wet Test Fixture interface and the existing LabVIEW application. This bodes well for the new software’s ability to locate elements, read forward and reflected electrical power, obtain accurate balance readings and isolate the stable on and off intervals, and finally to calculate acoustic power. If any of these key components were not functioning as well as the original system, there would be a tendency for the results to be meaningfully greater or less than the original system on average, which is not the case.

The random variability observed between the results with the original LabVIEW software and the results with the new Python software was less than 6% or 1.1 Watts on average for every quantity tested, which is comparable, and even slightly less than the variability between testing on different days and different wet test fixtures. One element narrowly failed with the current software and narrowly passed on the new software. However, this should be expected to occur occasionally, even when testing with the same software and hardware, due to the day-to-day variability of the UAs.

Because this report finds the results obtained with the new Python software to be within the bounds of normal variability, this report recommends proceeding with thorough acceptance testing of the new Python software.



Table 4 Raw data for Original/Retest comparison



Table 5 Raw Data for Python/LabVIEW comparison