

白皮书

# 通过pyDmdReader和Python 实现自动化生成报告

采用Python中集成Watchdog, pyDmdReader和FPDF库的方法



DEWETRON

## 摘要

利用 Python 及其库 (Watchdog、pyDmdReader 和 FPDF) 实现的自动化测试报告生成流程为数据分析和报告生成提供了一种快速高效的解决方案，增强了 OXYGEN 的功能。Watchdog 库会监测数据目录，并在完成新的记录后启动报告生成流程。pyDmdReader 库将测量数据导入 Python，使其可用于后续分析。最后，FPDF 库用于生成表格、整合图像，并将报告导出为 PDF 格式。

此解决方案不仅节省时间，还能确保生成报告的一致性。通过使用 pyinstaller 将 Python 脚本转换为 Windows 可执行文件，该脚本可以在任何计算机上作为独立应用程序运行。这种测试报告自动化的方法是将 DEWETRON 硬件和 OXYGEN 集成到测量流程中的宝贵工具，能够实现测试结果的快速且一致的评估。

## 引言

DEWETRON 系统在获取高精度测量数据方面表现非常出色。在 OXYGEN 中，有多种分析选项可供选择，并且可以基于测量屏幕生成报告，生成的报告可直接导出为 PDF 格式，见图 2。然而，在某些情况下，还需要一些额外的功能。例如，标准规定了飞机网络中高频电流的限值。为了检查某些谐波是否超过限值，测量的均方根电流和限值的可视化表示非常有用。图 1 展示了测量谐波电流失真与限值的一个示例。该图是使用 Python 和 Matplotlib 库创建的，并基于使用 OXYGEN 以 .dmd 文件格式记录的数据。在附录中附有一个示例报告。本白皮书中提出的解决方案描述了设置一个脚本的方法，该脚本会在 OXYGEN 中每次完成新记录时自动生成包含图像、表格和特征值的报告。该脚本以及随后生成的 Windows 可执行文件结合了三个 Python 模块，这些模块能够自动将记录的数据转换为测试报告。与使用 OXYGEN SDK 通过 C++ 创建插件相比，此基于 Python 的解决方案提供了更便捷的替代方案。

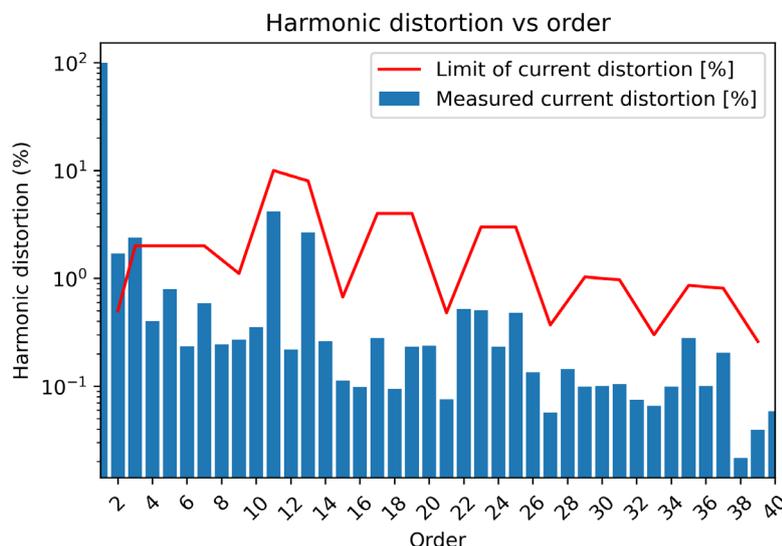


图1: 谐波电流失真与限值

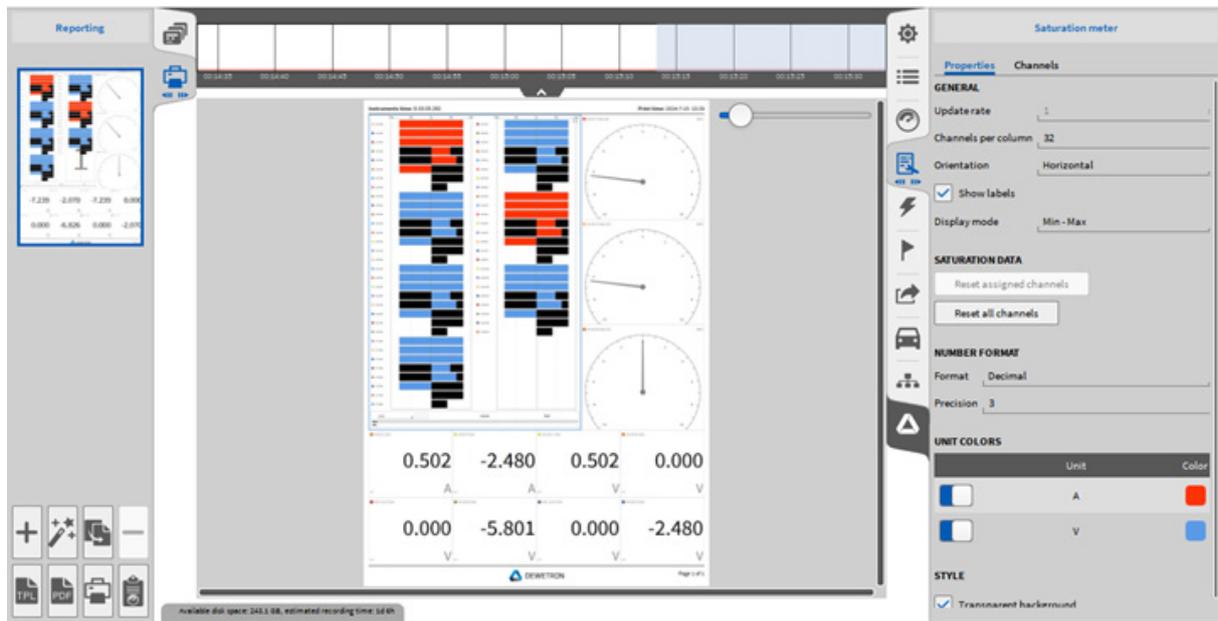


图2: OXYGEN 中的报告选项

## 传统分析VS自动化生成测试报告

在数据分析和报告生成领域，Excel和MATLAB是许多单位的标准工具。它们为数据处理、分析和可视化提供了一个强大的平台，不过自动化方法也可以通过其他方式实现。

### 使用 Excel 或 MATLAB 生成测试报告

Excel和MATLAB是广泛用于数据分析和报告生成的工具。它们提供了从基本的数据操作到复杂的数学计算和可视化的一系列功能。

Excel凭借其广为人知的界面和多种数据处理工具，非常适合处理小型数据集。但在自动化方面存在一定的局限性，要生成格式清晰的PDF报告，仍需进行一些手动操作。

MATLAB是一种高级语言和交互式环境，擅长于密集型计算和分析。它在处理矩阵和向量数据方面特别有用。但不足之处在于，MATLAB是收费的，并且库比Python少，这限制了它的灵活性和易用性。

尽管这些方法被广泛使用，但它们仍存在显著的局限性。从数据导入到分析和报告，这些方法在各个阶段都需要人工干预，或者需要付出费用。这种对人工输入的依赖增加了出错的风险，使整个过程更加耗时且效率更低。

### 使用 \*.dmd 数据和 Python 库生成测试报告

DEWETRON 的自动化报告生成解决方案采用 Python 这种强大的编程语言以及多个库，这些库包括：Watchdog、DMDReader 和 FPDF。这3个库完美配合，能够在新记录完成后自动生成测试报告。

Python 是一种高级编程语言，以其简洁性著称，因此被广泛应用于各种领域，包括数据分析和自动化。以下是所用库的简要概述：

### ▶ Watchdog

这个 Python 库用于监测文件系统事件。它可以监测文件夹或文件的变化，并在发生此类事件时触发操作。此功能用于触发 pyDmdReader 读取数据记录。

### ▶ pyDmdReader

此库用于读取 \*.dmd 文件，这种文件通常用于存储 OXYGEN 的测量数据。借助 pyDmdReader，系统能够轻松导入并处理 DMD 文件中的数据。

### ▶ FPDF

这个库用于在 Python 中生成 PDF 文件。它能让系统创建包含文本、图形和表格的 PDF 报告，从而全面展示数据分析结果。

在 Python 中编写完数据分析和报告生成定义的代码后，可以使用 pyInstaller 将脚本进一步转换为可执行文件。这样，自动化系统就可以作为独立的应用程序运行，从而进一步提高其可用性和可移植性。

## ▼ 示例脚本：使用WATCHDOG, pyDmdReader & FPDF

在本章中，将介绍这三个库之间的协同作用以及为您的用例生成测试报告的一种可能的解决方案。这三个库均可免费使用，并且可以按如下方式下载：

1. Watchdog: <https://pypi.org/project/watchdog/>
2. pyDmdReader: <https://github.com/DEWETRON/pyDmdReader>
3. FPDF: <https://pypi.org/project/fpdf/>

这个方法很简单，就是设置 OXYGEN 保存其数据的目录，并利用这样一个过程：当数据存储完成时，文件名的后缀会从 “\*.unfinished.dmd” 变为 “\*.dmd”，这会触发脚本读取所需的通道及其数据以生成测试报告。在这个阶段，所有 Python 工具都可用于进一步分析和绘制数据，一旦完成，就可以使用下一个库 fpdf 来定义包含图形和表格的输出 PDF。一旦创建了 PDF 并且经过了休眠时间，可能会处理新的存储内容，从而实现连续的测量评估和报告生成。

要使用该脚本，在 OXYGEN 中无需进行任何调整，只需确保保存目录与脚本中的扫描目录一致，例如 “C:\DATA”。

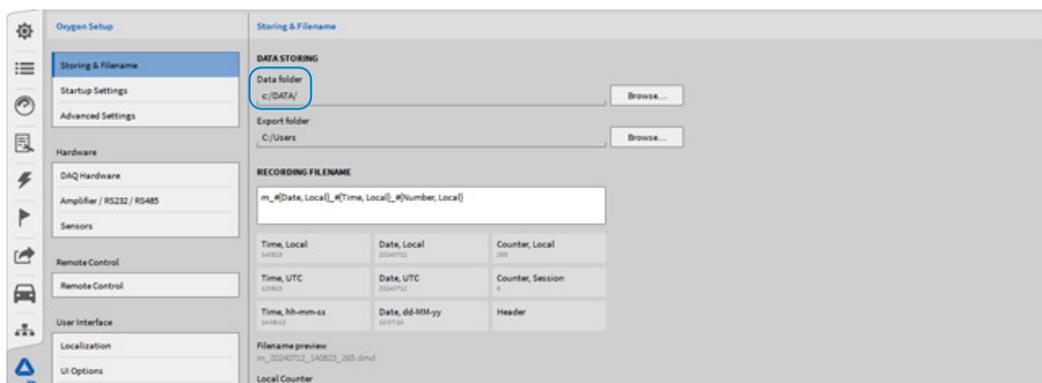


图3: OXYGEN的数据存储目录

## 简单的 pyDmdReader 示例

接下来是对 pyDmdReader 库中包含的命令的简短概述。从 \*.dmd 文件中读取测量数据的三个最重要的方法是：

- ▶ read\_array(ch\_names, start\_time, end\_time, timestamp\_format, max\_samples) – 获取为数组
- ▶ read\_dataframe(ch\_names, start\_time, end\_time, timestamp\_format, max\_samples) – 获取为数据帧
- ▶ read\_reduced(channel\_names, max\_samples) – a 获取为数据帧的统计值 (Min, Max, AVG, RMS)

```
import pyDmdReader

file_path='C:\DATA\m_20240726_092323_282_example.dmd'
dmd = pyDmdReader.DmdReader(file_path) # Read dmd file as DmdReader class
print(f"Channels found in file: {dmd.channel_names}")
# Read configuration of first channel
first_channel = dmd.channels[dmd.channel_names[0]] # Read channel as ChannelConfig class
print(f"Name: {first_channel.name}")
print(f"Unit: {first_channel.unit}")
print(f"Samplerate: {first_channel.sample_rate} Hz")
# Read all data from channel 'AI 1/1'
data = dmd.read_dataframe('AI 1/1 Sim')
print(data)
# Read data from channel 'AI 1/1' from 1s to 3s after recording start
data = dmd.read_dataframe('AI 1/1 Sim', start_time = 1, end_time = 3)
dmd.close()
```

若要获取有关设置和通道配置的其他信息，您可以通过 pyDmdReader 类的属性来获取。“dmd” 的 “pyDmdReader.\_dmd\_reader.DmdReader” 的属性有：

```
dir(dmd) #Attributes of dmd=pyDmdReader.DmdReader()
['allchannels', 'channel_ids', 'channel_names', 'channels', 'close', 'configuration_etree',
'configuration_xml', 'headers', 'markers', 'measurement_duration', 'measurement_start_time_local',
'measurement_start_time_utc', 'read_array', 'read_array_by_index', 'read_dataframe',
'read_reduced', 'version']
```

“first\_channel” 的属性 “pyDmdReader.\_channel\_config.ChannelConfig” 包括：

```
dir(first_channel) #Attributes of first_channel=dmd.channels[]
['description', 'dtype', 'handle', 'max_sample_dimension', 'measurement_duration', 'name',
'range_max', 'range_min', 'raw_sample_type', 'reduced_sample_type', 'reduced_sweeps',
'sample_rate', 'sweeps', 'type', 'unit']
```

## 报告自动化示例：watchdog, pyDmdReader, FPDF

以下来自 Spyder (Python 3.11) 编辑器的代码是一个基于 \*.dmd 文件使用 watchdog 和 FPDF 生成测试报告的示例。

```

import numpy as np
import pyDmdReader
import time
from watchdog.observers import Observer
from watchdog.events import FileSystemEventHandler
from fpdf import FPDF
import matplotlib.pyplot as plt
import datetime

data = {} #Raw data
results = {} #Calculated results
#Folder in which measurements (*.dmd) are saved and which should be scanned
scan_directory = 'C:\DATA' #Directory to watch for new recordings
def on_moved(event):
    if event.dest_path.endswith('.dmd') and not
event.dest_path.endswith('unfinished.dmd'):
        print(f'New .dmd file detected in: {event.dest_path}.')
        new_file_time = time.time()
        global results
        dmd_file = pyDmdReader.DmdReader(event.dest_path)
        all_channel_names = dmd_file.channel_names
        channels = defined_channels #Define all channels/data to create the test report
        for channel in channels:
            data[channel], ts = dmd_file.read_array(channel)
        dmd_file.close()
#Data processing e.g. harmonics distortion per harmonic
#Save results to global variables for additional analysis
#Create plot/s
plt.savefig(f'plot_{date_time}.png', dpi=300)
#Create PDF
class PDF(FPDF):
    def header(self):
    def footer(self):
#Write report text and values and tables
pdf.image(f'plot_{date_time}.png', x=20, y=y, w=180) # Add plot/s
#Save PDF
pdf.output(filename, 'F')
print('Report finished!\n')
print('Waiting for new recording in "C:\DATA".\n')

if __name__ == "__main__":
    event_handler = FileSystemEventHandler()
    event_handler.on_moved = on_moved #Condition to trigger the watchdog
    observer = Observer()
    observer.schedule(event_handler, path= scan_directory , recursive=False)
    observer.start()
    try:
        while True:
            time.sleep(10)
    except KeyboardInterrupt:
        observer.stop()
    observer.join()

```

## 将 Python 脚本转换为 Windows 可执行文件 (.exe)

对于未安装Python的系统，或者为了完全避免安装必要的库，将Python脚本转换为Windows可执行文件是一种可行的解决方案。这可以通过使用 pyinstaller 库来实现，并将生成的 \*.exe 文件复制到未安装 Python 的测量设备上。

以下命令行将 Python 脚本转换为 \*.exe 文件：

- ▶ “--onefile” 用于创建包含所有依赖项的单个可执行文件 (exe)
- ▶ “--noconfirm” 如果你想通过凭证，请跳过所有问题
- ▶ “--add-data” 从当前的 OXYGEN 目录添加 dmd\_reader.dll 文件
- ▶ “--exclude-module PyQt5” 防止出现与图表相关的警告

```
(base) C:\path> pyinstaller --onefile --noconfirm --add-data "C:\Program Files\DEWETRON\OXYGEN\bin\dmd_reader_api_x64.dll;" --exclude-module PyQt5 Watchdog.py
```

当此过程完成时，应显示出如下所示的消息。

```
415686 INFO: Building EXE from EXE-00.toc completed successfully.
```

生成的 \*.exe 文件现在可以转移到任何其他系统，在每次OXYGEN数据存储完成后，将在同一目录中创建报告。

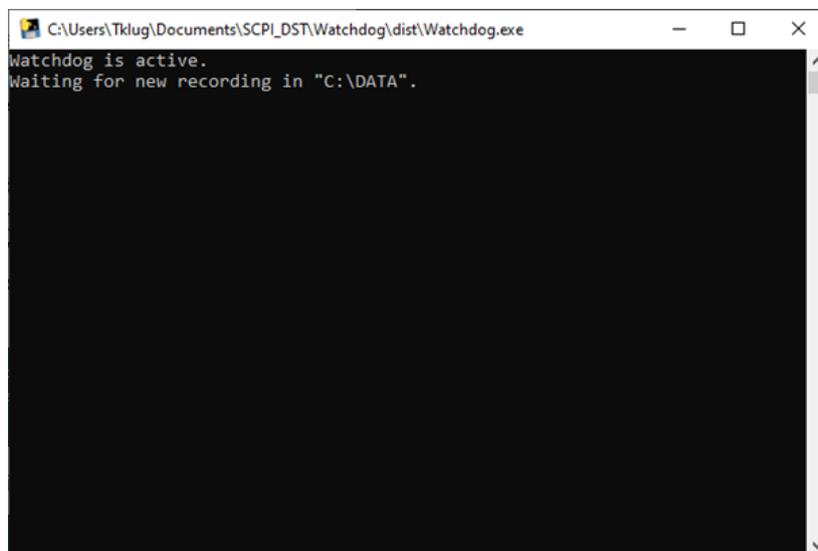


图4: Watchdog.exe.

## 总结

使用Python及其库（Watchdog、pyDmdReader和FPDF）的自动化测试报告生成流程为数据分析和报告生成提供了一种强大且高效的解决方案，扩展了OXYGEN的功能。Watchdog库会监测数据文件夹，并在新数据存储完成后触发报告生成流程。pyDmdReader库将测量数据读入Python，使其可用于所有后续分析。最后，FPDF库用于生成表格、整合图像，并将报告以PDF格式导出。

此解决方案节省了时间，并提高了报告生成的一致性。通过使用 `pyinstaller` 将 Python 脚本转换为 Windows 可执行文件，该脚本可以在任何计算机上作为独立应用程序运行。

总之，所提出的自动化生成测试报告的方法可以成为将 DEWETRON 硬件和 OXYGEN 集成到测量流程中的有用工具，能够快速且一致地评估测试结果。



## 作者

Thomas Klug



Thomas Klug毕业于格拉茨技术大学，拥有自然科学和材料科学方面的背景。在本科学习期间，他专注于地球物理学和太阳能电池领域；研究生阶段，他则专注于半导体和生物基材料。在求学期间，他曾就职于AVL和虚拟车辆研究中心。2023年毕业后，他加入了DEWETRON，担任工业制造、软件接口以及通用测试和测量解决方案方面的应用工程师。

## 附录

下面是一个示例报告，展示了最大频率、最大总均方根电流、一次最大均方根电流，随后是各次的均方根电流和失真。

*Report automation via pyDmdReader and python*



### Report automation via pyDmdReader and python

The following report shows the maximum RMS values over all phases and relates them to the RMS current of the first order.

The fundamental frequency is: 60.00 Hz

The maximum current (total RMS) is: 1876.069 Ampere

The maximum current of the first order (RMS) is: 1525.087 Ampere

The total harmonic distortion is: 5.479 %

**List of orders and their maximum current aswell as their distortion:**

0. order:	44.822 A	2.94% harmonic distortion
1. order:	1525.087 A	100.00% harmonic distortion
2. order:	26.037 A	1.71% harmonic distortion
3. order:	36.403 A	2.39% harmonic distortion
4. order:	6.116 A	0.40% harmonic distortion
5. order:	12.154 A	0.80% harmonic distortion
6. order:	3.569 A	0.23% harmonic distortion
7. order:	9.015 A	0.59% harmonic distortion
8. order:	3.719 A	0.24% harmonic distortion
9. order:	4.140 A	0.27% harmonic distortion
10. order:	5.384 A	0.35% harmonic distortion
11. order:	63.950 A	4.19% harmonic distortion
12. order:	3.322 A	0.22% harmonic distortion
13. order:	40.483 A	2.65% harmonic distortion
14. order:	3.972 A	0.26% harmonic distortion
15. order:	1.719 A	0.11% harmonic distortion

Date and time: 2024-08-07\_11-10-42 Page 1/2

图5: 示例报告

下图展示了已定义各次谐波的限值以及所测得的谐波失真，此外还有针对所有次谐波的总限值，即 5 A 或 0.25% 的失真。

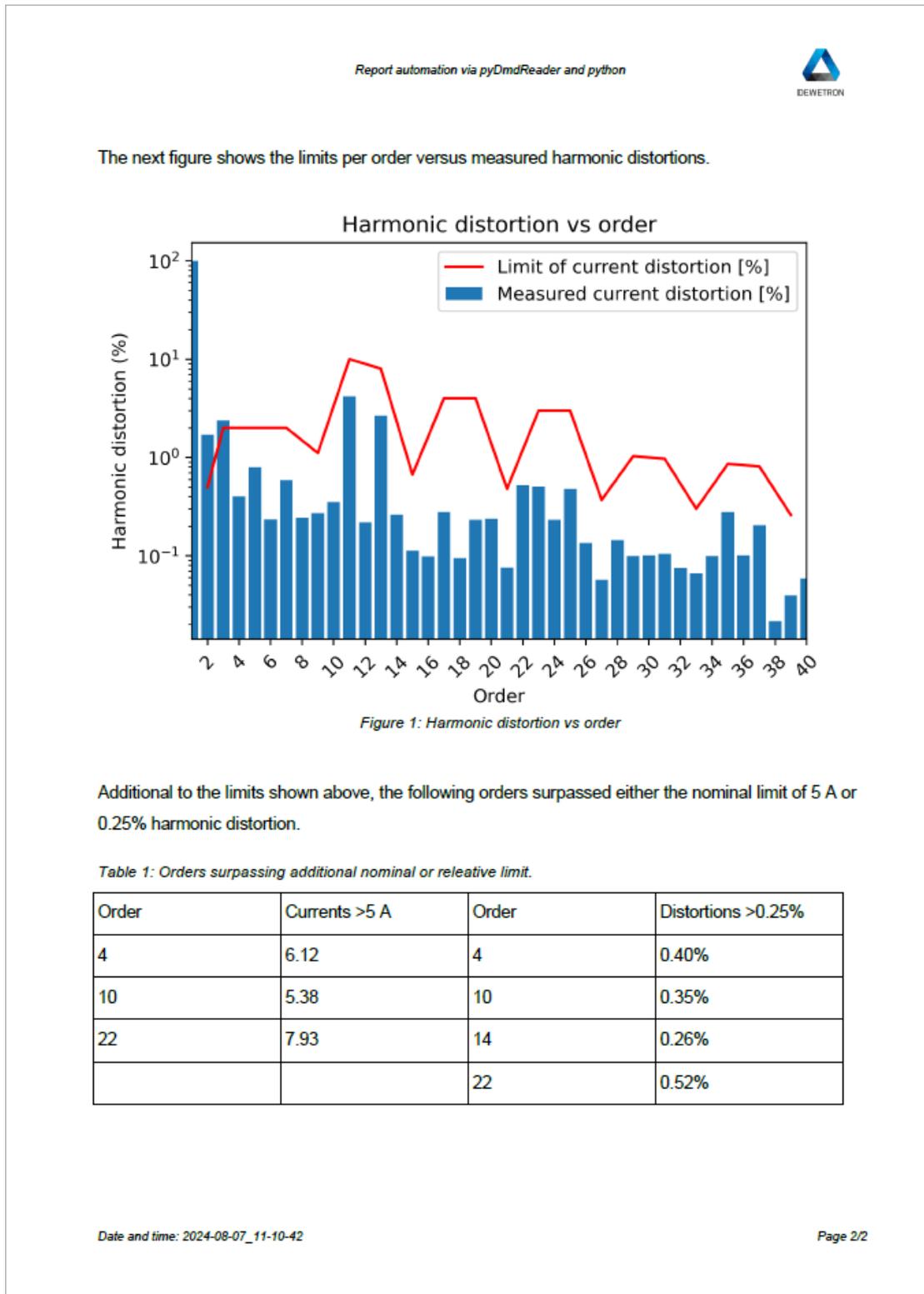


图6: 规定的谐波限值和测量的谐波失真



## 关于DEWETRON

DEWETRON是一家精密测试和测量系统的制造商，旨在帮助我们的客户使世界更可预测，高效和安全。我们的优势在于定制化的解决方案，并可以立即使用，同时也能快速适应能源、汽车、交通和航空航天行业的测试环境和复杂技术的不断变化的需求。

超过30年的经验和创新使DEWETRON赢得了全球市场的信任和尊重。在全球知名公司中，有超过25,000套DEWETRON测量系统和超过400,000个测量通道在使用。

DEWETRON 在25个国家拥有超过120名员工，隶属于TKH集团，TKH集团是一家全球性的公司，专门从事全球创新解决方案的开发和供应。

DEWETRON的产品质量通过了ISO9001和ISO14001认证。测量数据的高度完整性由我们自己认可的校准实验室根据ISO17025保证。

关注德维创  
官方微信公众号



THE MEASURABLE DIFFERENCE.



DEWETRON

德维创测试设备（北京）有限公司  
北京市朝阳区华腾大厦1205室  
010-6777 7287

德维创检测设备（上海）有限公司  
上海市静安区国旅大厦1403室  
021-6289 0027

