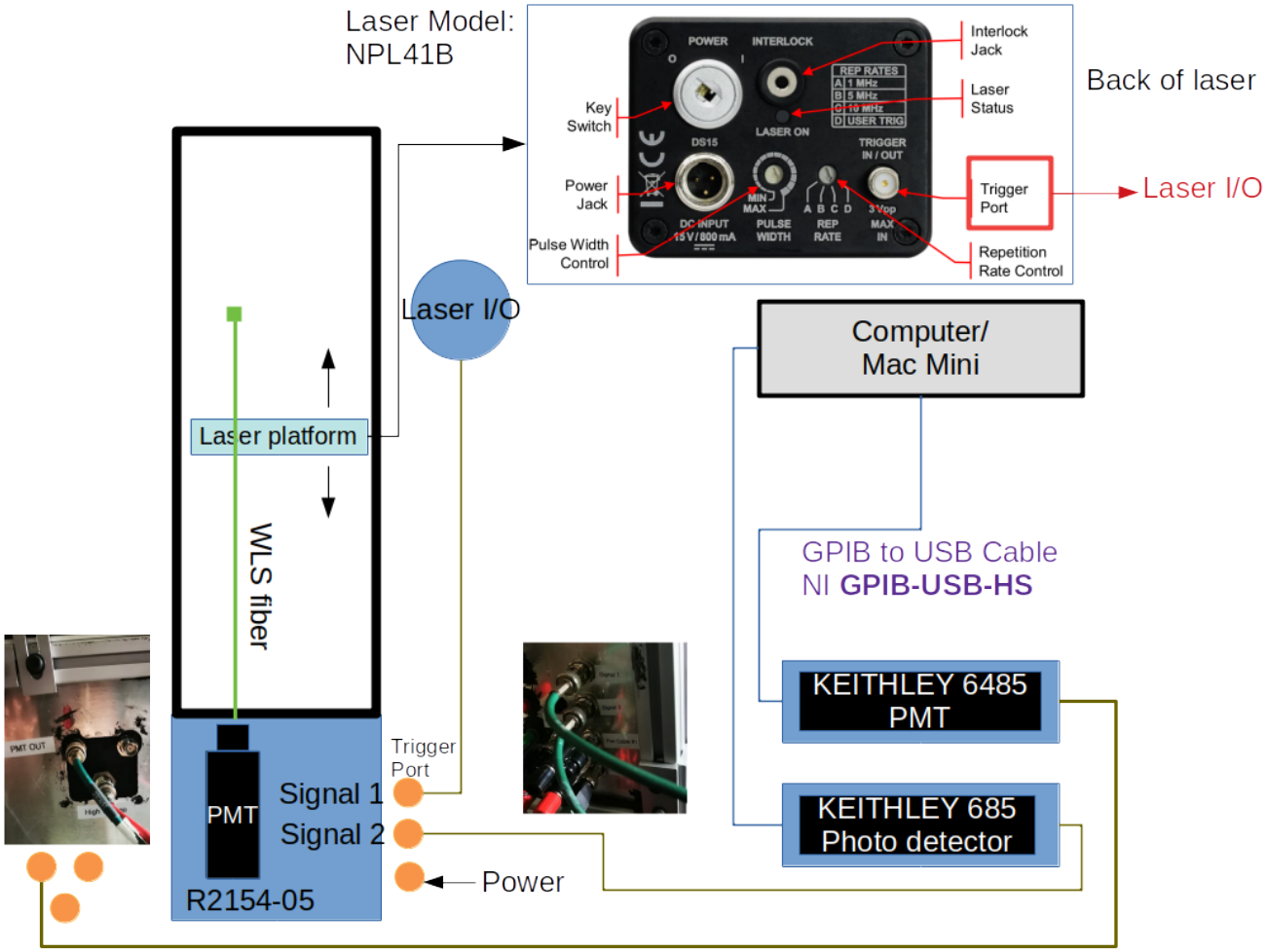
System Setup

Setup

The fiber testing box is a rectangular black box 2.55 x 0.40 x 0.34 m in dimension. An Arduino board controls the motion of a aluminum platform with a laser (NPL41B), splitter, test box system. The laser operates at 405 nm, the beam is split by a beam splitter cube allowing half of the beam to be reflected vertically and monitored by a photodetector. The horizontal beam then passes into the test box, illuminating a diffuser that then light up a wavelength-shifting (WLS) fiber.

Light from the WLS is sent to a PMT. The PMT is in a light-tight compartment with an iris in front of the PMT. The iris can be shut off with a switch, which prevents any light to reach the PMT when needed.

*Please refer to the manual on the table on how to open and close the black box.



Laser Model: NPL41B

PMT Model: R2154-05

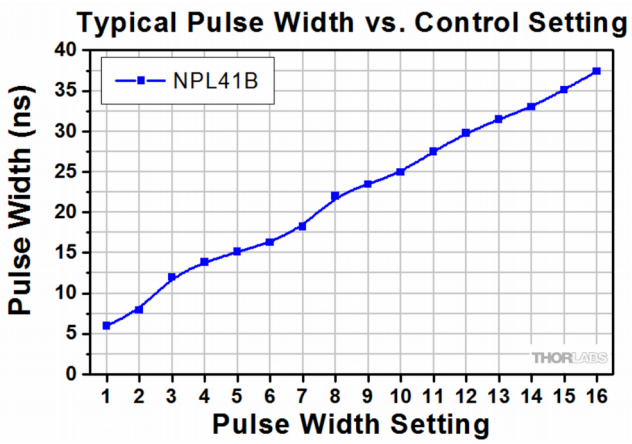
Photodetector: Det100A

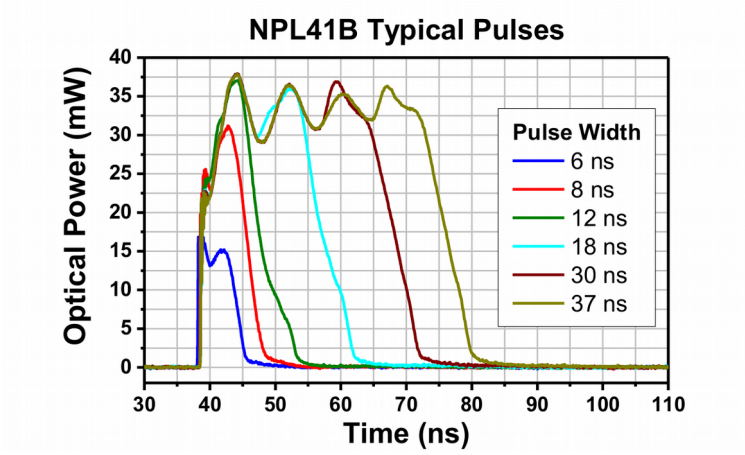
Picoammeters: Keithley 6485, 685, and the GPIB-USB-HS operations.

Laser Characteristics

The laser operates at 405 nm. It has 3 internal trigger settings and an external trigger setting tuned by the Rep Rate dial (see figure on previous page).

Pulse width





Adjusting the pulse width effectively changes the maximum number of photons going into the photo detector and the PMT. The photodetector (DET100A) used to monitor incoming laser beam cannot discern the shape of the laser pulse because it has a rise time of 43 ns. The PMT rise time is 3.4 ns (for R2154-02, which should be similar to R2154-05), so in principle it can measure the laser pulse shape at larger pulse width.

Repetition Rate

The laser has 4 rep settings (see figure on previous page). For A, B, and C, the trigger port output the internal trigger of the NPL41B. For D, where the trigger port accepts external trigger, connect the cable (signal 1) to a frequency generator to drive the laser at arbitrary rate.

PMT Characteristics

The model is R2154-05, but there is only datasheet for R2154-02. I assume both models are just variations of the same core PMT, so that their specs are the same.

Anode pulse rise time: 3.4 ns

Transit time spread (FWHM): 3.6 ns

Figure 1: Typical Spectral Response

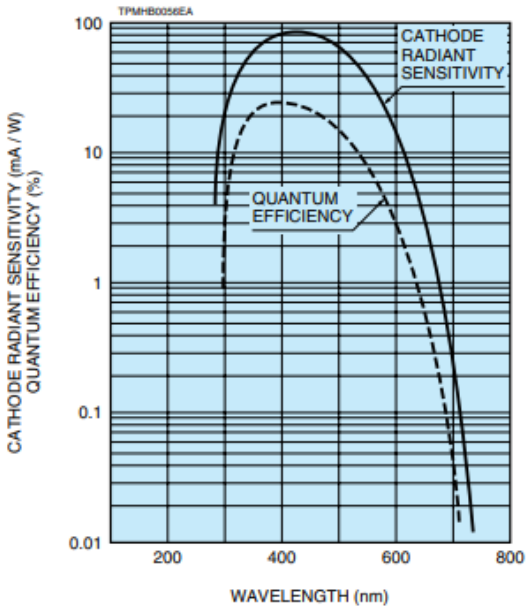
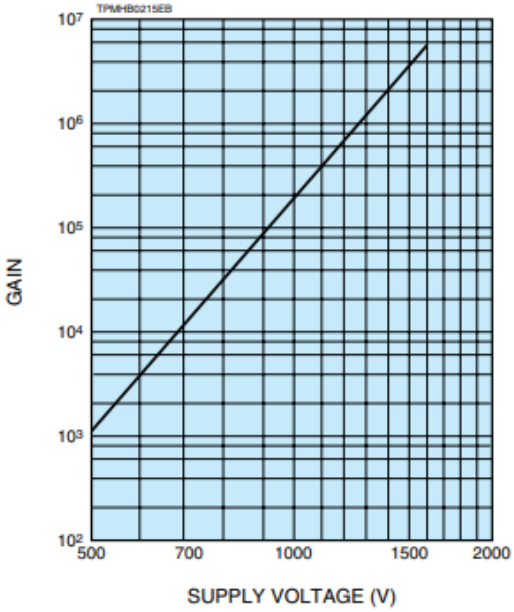


Figure 2: Typical Gain Characteristics



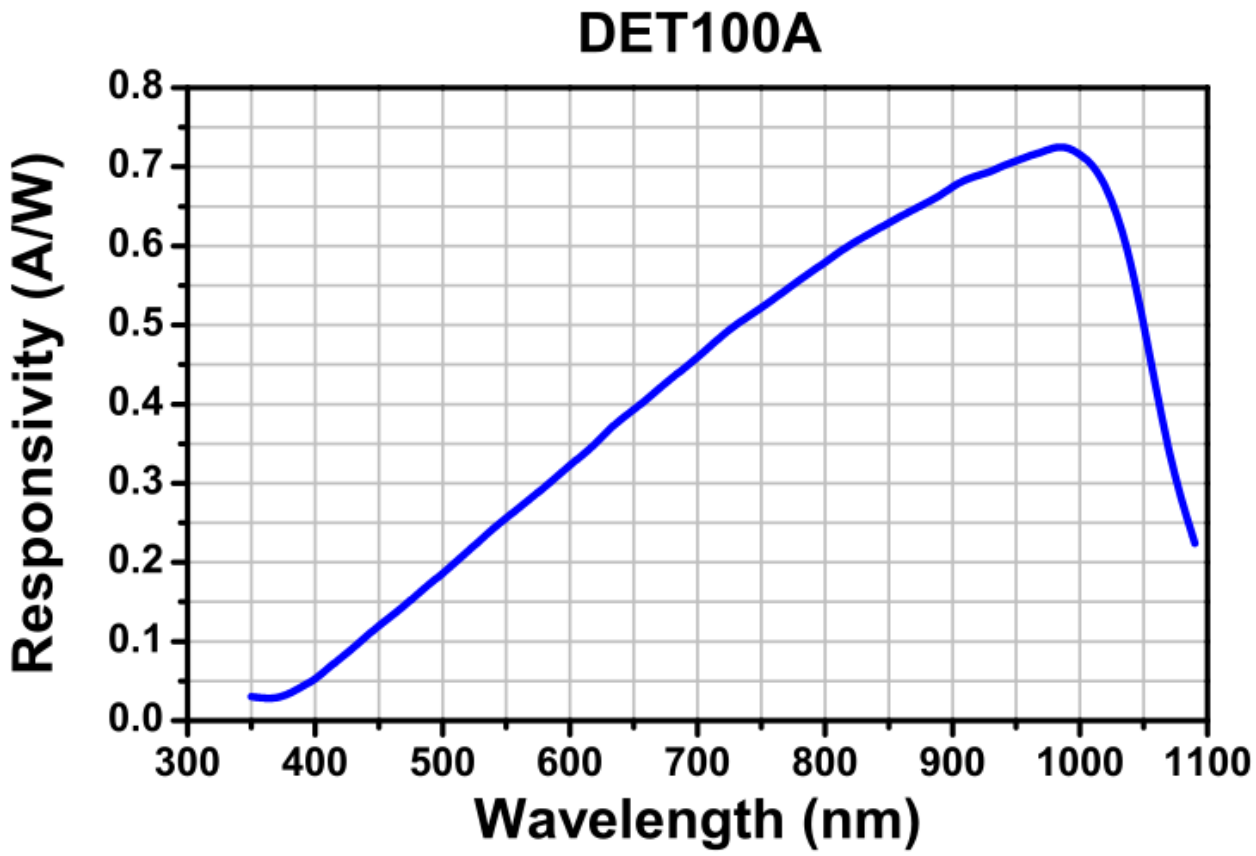
Warning:

The supplied voltage CANNOT exceed **1750 V**.

The maximum current CANNOT exceed **100 µA**.

Det100A

The laser beam is splitted by a beam splitter cube inside the test box. The Det100A monitors the splitted portion to provide a measure of the laser intensity. Variation in the laser intensity changes the amount of light going into the fiber, and the PMT. The laser monitor therefore serves as a crucial control to correct the effects due to the intrinsic variation in the laser pulse.



Picoammters and remote operation using the GPIB-USB-HS cables

Keithley 6485 is a digital picoammter and it measures the PMT output. The Keithley 685 is an analog picoammeter to monitor the photodetector readings.

Both are connected to a computer to allow remote reading of the data which is crucial for the automated data acquisition (DAQ) system.

Setting up GPIB support

We use two GPIB-USB-HS cables to enable connect. The cable is not plug-and-play and the appropriate drivers and backends are required before it works.

1. Download and install NI-488.2 driver: https://www.ni.com/en-us/support/downloads/drivers/download.ni-488-2.html#345631
2. Download and install the Virtual Instrument Software Architecture (VISA) library from NI: https://www.ni.com/en-us/support/downloads/drivers/download.ni-visa.html#346210
3. Download and install the pyvisa library to enable python control: https://pyvisa.readthedocs.io/en/latest/introduction/getting.html

Once finished, please read the basic instructions for the pyvisa library (pyvisa website in 3) to get started:

```
>>> import pyvisa
>>> rm = pyvisa.ResourceManager()
>>> rm.list_resources()
('ASRL1::INSTR', 'ASRL2::INSTR', 'GPIB0::14::INSTR')
>>> my_instrument = rm.open_resource('GPIB0::14::INSTR')
>>> print(my_instrument.query('*IDN?'))
```

“GPIB0::14::INSTR” will be the name of a GPIB enabled instrument. “GPIB0” indicates the device uses GPIB interface, “14” is the instrument’s address on the GPIB.

The K6485 address is currently set to 17. K685 address is 15. We send SCIP commands to the instruments using the DAQ software described in later section.

The commands can be found on **page 3-5** of the K6485 manual, which is attached in the manual folder.

Operation of fiber testing box

Note about the system

1) Shut off iris when opening the black box to prevent damage to PMT.
2) The laser has a violet indicator light at the back that shines steadily when the laser is on. I have covered it with a black tape but some light may still escape – so the black box is not totally dark. Consider completely erase the light.
3) When the platform moves, the detector signal may continually drop. The baseline signal established right now is about 4.6 μA. The signal could drop to as low as 3.8 μA. When the detector signal drops, the PMT signal measured at the same point also drops. I do not know exactly what's causing the issue. Some possibilities are
    1) Power to laser drops decreasing intensity.
    2) Platform motion causes beam to point away from the diffuser – less light illuminates the fiber.
4) Shaking the laser or the photodetector by hand don't affect the detector signal. So I tend to think the problem is in the laser. I have not tinkered with the laser power yet. Maybe it could help.
5) The problem can be "corrected" by instructing the platform to move towards the far side when it's already at the far side. This forces the bell to keep moving and "shake" the platform. I have observed doing so will cause the detector and PMT readings to go back up.

Software Instruction

Log on to **FiberTest**, open a terminal and navigate to DAQ folder. Inside there are a couple python packages:

1. fibertest.py
2. ft_arduino.py
3. DAQ.py
4. am_reader.py
5. RunDAQ.py
6. shakeitup.py

fibertest and **ft_arduino** are the original softwares used to control the platform.

am_reader is a simple utility to read the picoammeter signal output.

DAQ took functionalities from **ft_arduino** to improve the control of the platform, **am_reader** to read signal and has codes to export readings to csv files.

RunDAQ is a standalone python program to perform fiber measurement and automatically saves output to predefined directories.

Do

>> ./RunDAQ.py -h

will return

```
Process Fiber Test
optional arguments:
  -h, --help            show this help message and exit
  --id ID               Set fiber ID, default to -1
  --suffix SUFFIX       Suffix to file name prefix_id_suffix.csv, default toNone
  --prefix PREFIX       Prefix to file name prefix_id_suffix.csv, default to fibertest
  --save_dir SAVE_DIR   Saving directory, default to ./data/{today's date}/
  --rep REP             Number of rail passes default to 5
  --n N                 Number of accumulated measurements default to 5
```

For example

./RunDAQ.py --id 20

will measure a fiber with id 20, run 5 rail passes, and at each point read ammeter 5 times and report the average. The output will be saved to ./data/20210612/fibertest_000020.csv

The tester can control the exact save directory, the prefix and a suffix in case a need arises.

The output file records data in columns of:

pmt01, pmt02… pmtNN, det01, det02,…,detNN, pmt_avg, pmt_stdev, det_avg, det_stdev

where 01, 02.. NN indicates the number of passes and each row contains the measurement at a point along the fiber. In order of furthest towards the PMT. So the first row record the measurements furthest from the PMT and the last row is the point closest to the PMT.