Tutti-Frutti Device Configuration GUI in MATLAB

SLAC NATIONAL ACCELERATOR LABORATORY

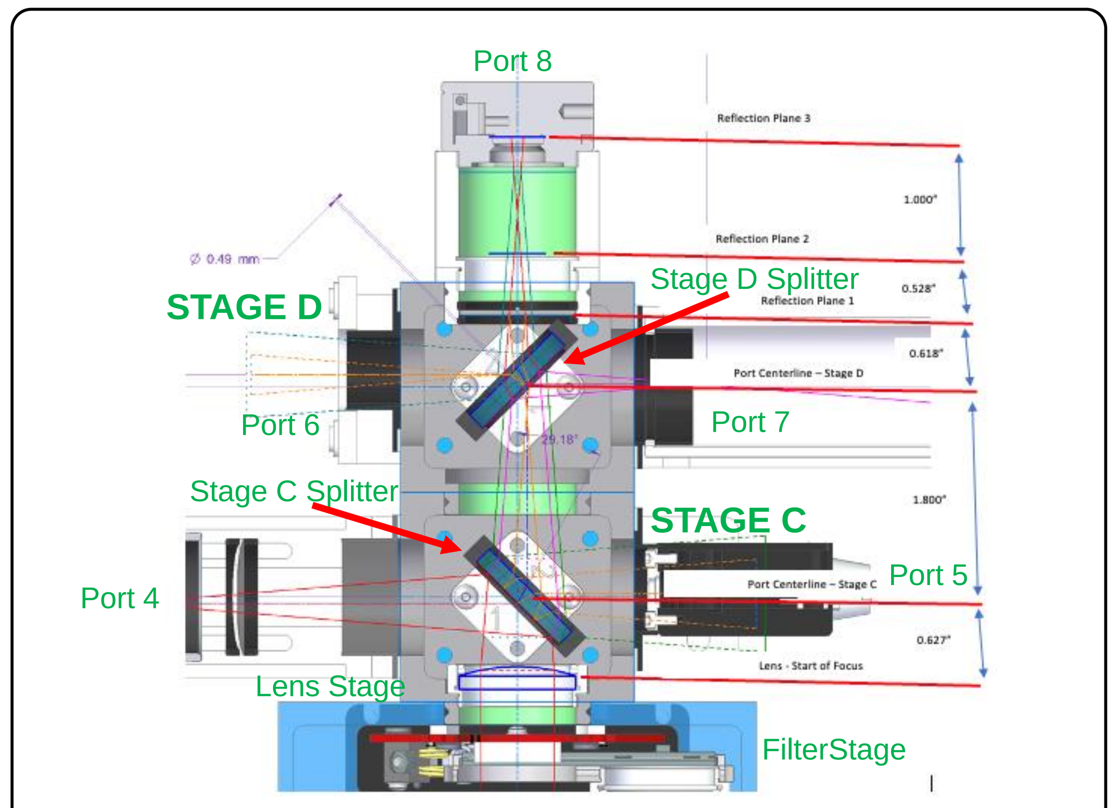
Aidan Kung

Introduction

- The Tutti-Frutti is a custom diagnostic tool that can be configured with filters, beamsplitters, spectrometers, cameras, and other diagnostics to manipulate and measure laser specs for various experimental and analytical purposes.
- As the primary diagnostic tool for experimenters both in the Near and Far Halls, it has seen widespread implementation at SLAC which will only continue to scale up in coming years.
- Because the Tutti-Frutti is so ubiquitous, it is important to be able to book-keep exactly which optics are in what nominal positions within the device, and exactly how these user-defined configurations affect key beam parameters like energy density and power output.

Motivation

- As it stands, their complexity makes them difficult to tinker with; as such, the current nature of Tutti-Frutti configuration relies heavily on back-of-the envelope calculations and general user intuition.
- This MATLAB GUI facilitates Tutti-Frutti user configuration by calculating and reading out wavelength, energy, and power-based instrument damage thresholds and beam parameters based on configurational presets provided by the user, saving hours of potentially erroneous hand calculation by experimenters as well as thousands of dollars of potential equipment damage.
- The goal will be to universalize Tutti-Frutti standard operating procedure and make it easy for any user to tailor their configuration for their particular experimental need, and to know everything about that configuration at the push of a button.



- The beam enters at the base (not shown) of the Tutti-Frutti assembly through one of 3 ports (user-determined) where they are reflected by a mirror vertically into a blue filter stage (shown) and then into a lens stage, where the attenuated beam is then passed through into beamsplitters and distributed into optics at each of 5 upper ports. As one can see, the beamsplitters are slightly offset with respect to one another, which can increase complexity and difficulty of hand calculation once variables like relative rotation of stages and different output instrument aperture sizes are introduced.

Some Equations Used

- Period and repetition rate are reciprocal:

$$\Delta t = \frac{1}{f_{rep}} \quad \text{and} \quad f_{rep} = \frac{1}{\Delta t}$$

- Pulse energy calculated from average power:

$$E = \frac{P_{avg}}{f_{rep}} = P_{avg} \cdot \Delta t$$

- Average power calculated from pulse energy:

$$P_{avg} = \frac{E}{\Delta t} = E \cdot f_{rep}$$

- Peak pulse power estimated from pulse energy:

$$P_{peak} \approx \frac{E}{\tau}$$

- Peak power and average power calculated from each other:

$$P_{peak} = \frac{P_{avg}}{f_{rep} \cdot \tau} = \frac{P_{avg} \cdot \Delta t}{\tau} \qquad P_{avg} = P_{peak} \cdot f_{rep} \cdot \tau = \frac{P_{peak} \cdot \tau}{\Delta t}$$

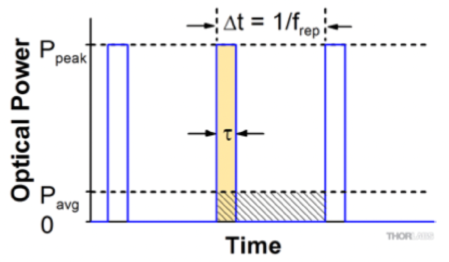
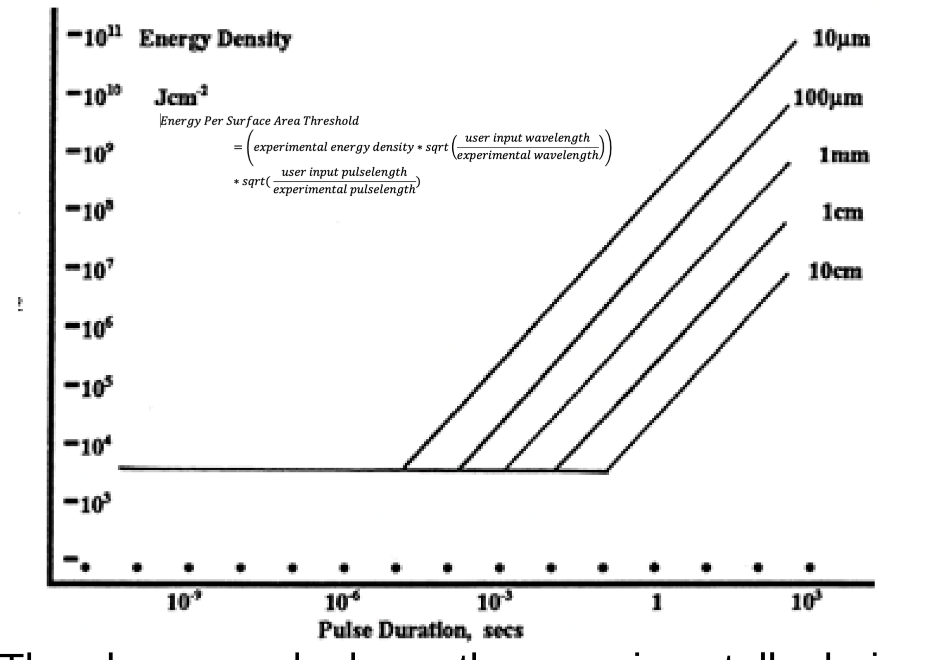


Figure 8. Laser Pulse Parameters

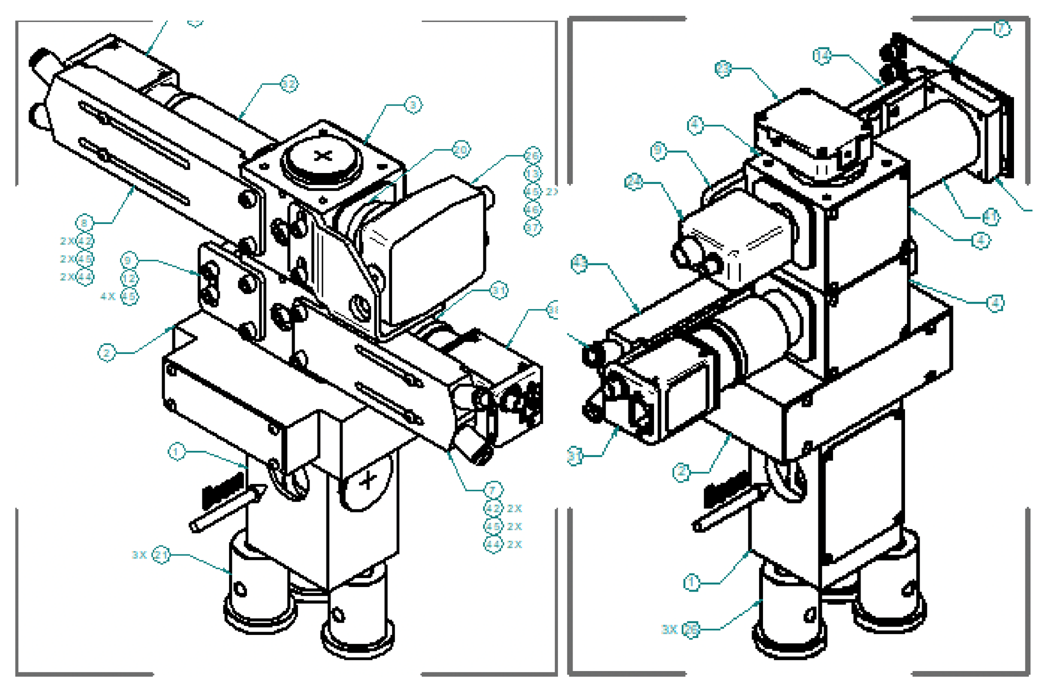
Δt	Pulse Period
E	Energy per Pulse
f_{rep}	Repetition Rate
P_{avg}	Average Power
P_{peak}	Peak Power
τ	Pulse Width

$$\text{Adjusted LIDT} = \text{LIDT Energy}\sqrt{\frac{\text{Your Wavelength}}{\text{LIDT Wavelength}}} \qquad \text{Adjusted LIDT} = \text{LIDT Energy}\sqrt{\frac{\text{Your Pulse Length}}{\text{LIDT Pulse Length}}}$$

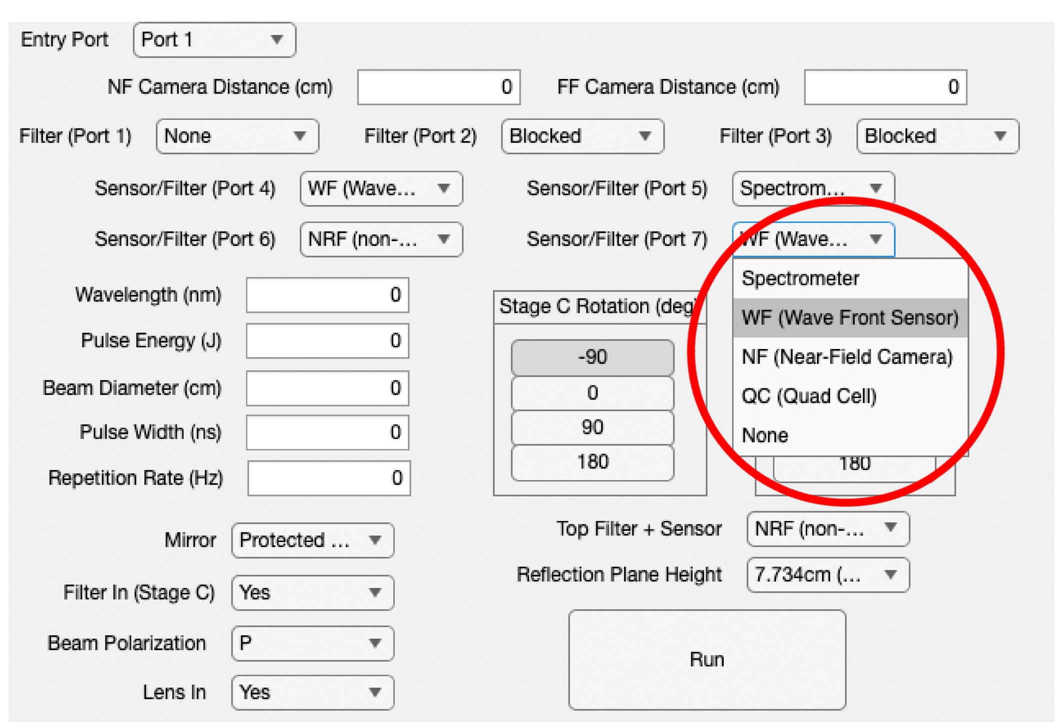
$$\text{Energy Per Surface Area Threshold} = \left(\text{experimental energy density} * sqrt\left(\frac{\text{user input wavelength}}{\text{experimental wavelength}}\right)\right) * sqrt\left(\frac{\text{user input pulselength}}{\text{experimental pulselength}}\right)$$



- The above graph shows the experimentally derived laser induced damage threshold (LIDT) for Thorlabs NDUV series filters which varies as a function of pulse length and spot size. The GUI script accounts for this differential using the formula for LIDT shown above based on user input wavelength/pulselength and experimentally used damaging wavelengths.



GUI/Code Screenshots



The user can determine whether certain lenses and filters are in place, the height of these optics, which kinds of sensors are located at which ports, input beam specs, rotation of stages relative to each other (and thus exact beam offsets), and more.

```
elseif mirror == "FS Wedged Window"
    if wavelength < 245
        fig16 = uifigure;
        uialert(fig16,'Input wavelength outside mirror reflection range','Invalid wavelength');
    elseif wavelength > 400
        fig17 = uifigure;
        uialert(fig17,'Input wavelength outside mirror reflection range','Invalid wavelength');
    end
    if ener_per_sa_initial > (5*sqrt(wavelength/355))*sqrt(pulswid/10)
        fig18 = uifigure;
        uialert(fig18,'Certification energy threshold exceeded; mirror may sustain damage','Warning');
    end
    mirror_transmission = 0.005;

end

pulsener_up = mirror_transmission*pulsener*attenuation_factor_up;
```

- The above code screenshot provides an encapsulation of the logic used to build the script; pulsener_upC refers to the pulse energy going into the stage C beamsplitter which is in turn dependent on user-determined mirror transmission, filter selection, relative positioning/angles of optics, etc.

Output

Parameters	Port 1	Port 2	Port 3	Port 4	Po
Output Energy (J)	0	6000	0	5.76	0
Output Beam Area (cm^2)	0	3.14159265358979	0	0.7654398655591	0
Output Peak Pulse Power (nW)	0	120	0	0.1152	0
Output Power Density (W/cm^2)	0	38197186342.0549	0	150501698.6747	0
Output Average Power (J/s)	0	300000	0	288	0

Port 1 Warnings	Port 2 Warnings	Port 3 Warnings	Port 4 Warni
	Beam diameter exceeds port 2 power meter apertur…		Input waveler
	Beam power density exceeds maximum measurable…		

- The current output state is a series of excel-styled tables which read out key beam parameters at apertures and internal devices. Also provided based on these calculations are potential damage and dimensional warnings based on incongruencies in the user input and internal GUI calculations, expandable on scroll-over.

Conclusions

- The tool is currently functional and will substantially benefit all those who use Tutti-Frutti diagnostics, and increase ease of use / accessibility to a device that will continue to increase in importance in the experimental halls.
- Potential future considerations include accounting for specific edge cases (e.g. differential offset of beam through beamsplitter based on wavelength), cosmetic improvements to UI, user customizability of optical tool options and specifications, increasing available functionalities, packaging and handoff of script.

Acknowledgments

I would like to thank my mentors, Corey Hardin and Connor Dixon, for their enthusiastic support and guidance this summer.