# GUI

The graphical user interface (GUI) design was formulated based on the initial design suggestions and project requirements. The GUI is the perspective through which users will interact with the TCM, from performing measurements on a sample to data analysis.

First, the interface was designed to be clean and user friendly. This was implemented by designing each window with a minimal number of on-screen controls while maintaining full functionality. Common buttons, such as “OK” and “Cancel” were placed in the bottom-right corners, which is where users would expect to find such controls. Equipment statuses are provided via real-time feedback, appearing as green or red indicators with matching “On” or “Off” text. The GUI also updates in response to available actions by disabling controls associated with such actions. When the program is not connected to the TCM hardware then: 1) a red indicator lights up with matching “Off” text, and 2) the data collection and hardware utilities actions are disabled. Connecting to the TCM hardware causes the GUI to switch to a green indicator light, “On” text, and enable buttons associated with hardware interaction.

The video feed from the TCM’s cameras are always displayed in the upper-left corner, providing an extra sense of constancy and uniformity to the program. Any window that displays a camera video feed also includes stage controls. The sample location can be linked to or separate from the camera selection; when linked, the stages will automatically reposition so that the same sample location is centered under the newly-selected camera.

Basic stage and camera functionalities are implemented in the Controls.fig and Controls.m files. These two files are designed to be extended as needed to provide a specific set of actions. Currently, two other file sets extend the Controls file set:

* Control\_PositionSample: Set the sample position under the cameras
* Contols\_CollectData: Collect data at the current sample position

# Analysis

The analysis aspect of the program is built based on the previous work done on the TCM. The previous functionality has been extended with a GUI for configuring the analysis settings. A real-time visualization window was created to provide progress feedback during analysis, including graphs of the current solution and an estimated percent completion.

The data analysis routines are split into three files: FitProperties.m, FitTCMData.m, and ThermalWaveNumbers.m.

* FitProperties.m
  + A simple enumeration class
  + Provides named-type access to the properties that are calculated by the analysis
* FitTCMData.m
  + Accepts user-specified analysis options
  + Loads the data files
  + Performs data pre-processing
* ThermalWaveNumbers.m
  + Contains the actual data analysis routines
  + Uses minimization methodologies to:
    - Iteratively solve for the fit properties
    - Converge on a solution matching the measurement data

The estimated percent completion is based on past observation of average data analysis performance and convergence indicators. However, the nature of the solution does not enable an absolute prediction of analysis progress. Instead, the reported progress should be seen as a generalization that is given with the intent of providing user feedback and demonstrating that the program is still active and working. This is specifically designed to prevent a false impressions that the program has frozen because nothing appears to be happening.

# Resources

The TCM software uses two text-based configuration files, Settings.ini and Preferences.ini. The first, Settings.ini, is designated for containing the options related to hardware/software interactions. These include items such as hardware addresses for communications, stage limits, and device-specific configuration options. The second, Preferences.ini, is used to store softer items and provide a unified user experience between sessions. These include items such as the most recent data file name, window sizes and positions, and the previous data analysis configuration.

An Excel spreadsheet, Database.xlsx, is used to store constants that are essential to program operation. A spreadsheet format was chosen so that the data it contains will be modifiable. One important set of data it contains are material properties for films and a few example substrates. Although not necessary for generating a correct solution, the analysis duration can be shortened by selecting a starting substrate with properties similar to the material be analyzed.

# GPIB Interface

The GPIB interface was designed around a single base class, contained in GPIB\_Interface.m, which implements the function to perform the raw communications with GPIB devices. This class is extended by device-specific classes to provide functions that represent the capabilities of the corresponding device. The child classes are:

* DS345\_Control.m
  + DS345 function generator
  + Provides controls such as changing the frequency and pulse amplitude
* ESP300\_Control.m
  + ESP300 stage controllers
  + Provides controls such as setting stage limits, stage travel speeds, and stage positions
* SR830\_Control.m
  + SR830 lock-in amplifier
  + Provides controls such setting time constants, sensitivity, and reading phase difference
* ProbeLaser\_Control.m
  + Controls the probe laser on/off status via the SR830 control class
  + Uses the output port of the SR830 to send an analog control signal to the laser control box