Commit History - CONTROLLER

January 12, 2026

v.20260112: Recovery corrections

- GUI: removed the v.20250912 static test type IF test in showEquations() that prevented recovery after system crash
- GUI: removed the v.20250912 .xml output options in UpdateLogs2() for static test types (.csv still OK)
- GUI: initialized self.*_update variables in UpdateLogs2() both in __init__() and beginUpdates()
- GUI: added 0.5 second wait during the recovery routine to prevent MainWin from reading UpdateMountThreadLoad() before beginUpdates() was called, causing a "data not found" error.
- GUI: incorporated limitations in EPFT specimen type to C(T) only
- GUI: deactivated a, K calculation options for static test selection
- GUI: required options in Measurement Output Controls dialog box to be sent to device by explicit SEND command instead of making them automatic (e.g. OP, cycle interval, crack interval, etc.)
- GUI: added format highlights to values in previous line item
- GUI: test type selection dropbox now deactivates while test is running
- GUI: changed comp_values_list[0] (cycles) format to INT from FLOAT
- GUI: enabled static test data recovery
- GUI: corrected C(T) PD with ref. solution (did not divide by W for a_init)
- BIN (ADX): initialized n_begin_counting to 1 for all dwell tests

December 10, 2025

v.20251210: EPFT load vs time plotter

- BIN (ADX): set pdcomp to 0 after PD / compliance averaging for EPFT if ai set to 0
- GUI: added load vs. time for EPFT
- GUI: automatically shut of soft start / stop for dwell / EPFT
- GUI: automatically set slope count per crack length measurement to below cycle segment for EPFT
- GUI: deleted some old script

December 5, 2025

v.20251205: EPFT, minor fixes

- BIN (ADX): built in EPFT capability into binary based on dwell cycle routine (see items below).
- BIN (ADX): modded dwell segments to check against EPFT flag
- BIN (ADX): changed location of recordLast() command to happen before (and independent of) softStop IF test.
- BIN (ADX): edited recordLast() to avoid logging "wrap up" average values and final point for EPFT tests.
- BIN (ADX): limited min n_begin_counting for dwell to 1. Only relevant for EPFT which is exclusive to ADX.
- BIN (ADX): moved setDwellOffsets_L() after setStepandRange() to properly set min load on first cycle of first segment
- BIN (ADG, ADX): set measurement start count N to prevent it from going to 0.
- GUI: fixed multiple stop criteria tab to prevent it from re-sizing itself out of the tab frame
- GUI: changed dwell hold time lower limit to 0.1 to allow for process time within the segment
- GUI: added E-P Fracture Toughness to menu / log file writer, and limited slope count and calculation values to prevent exceedence of cycle limit
- GUI: fixed static test time .csv output (int to float)
- GUI: edited log writer to include ramp and hold durations for dwell CG and EPFT

September 12, 2025

v.20250912: Static log / output file writer

- GUI: added Bnet to SEN specimen calculations
- GUI: changed log writer to create xml output and proper _segdata file for tensile tests
- GUI: added safety to K convergence check to prevent division by zero
- GUI: set up proper specimen selection menu for testing static specimens (CT, SEN-B, or smooth LCF)

August 11, 2025

v.20250811: Metric analysis fixes

- GUI: plot units are meters or MPa sqrt(m); dashboard and controller units are generally mm
- GUI: fixed dashboard and plot labeling to use m instead of mm

August 7, 2025

v.20250807: Metric units, eff. B, W correction

- GUI: Added metric unit capability to GUI for crack length and K calculations
- GUI: corrected "effective' B, W calculations based on inner R / outer R values

June 27, 2025

v.20250627: Time stop criterion fix

- GUI: added calculation of time_remaining when adjusting time based stop criterion in setConfiguration() to avoid premature stoppage.
- GUI: moved assignment of stop criterion values to BEFORE assignment of stop criteria types (again to prevent early stoppage).

June 23, 2025

v.20250623: minor GUI fix

- GUI: re-sized the stop criteria tab to fit properly
- GUI: corrected device number from 0x150 to 0x1 (default ADX)

June 30, 2025

v.20250619: multiple stop criteria

- BIN: added multiple stop criteria
- BIN: corrected time and cycle counters to properly track appended segments
- BIN: corrected curr_max/min_tx_load calculation
- BIN(ADX): incorporated "slow start" with point based PID correction instead of cycle based
- BIN(ADG): truncated peak averaging routine
- BIN(ADG): incorporated additional check on dwell fatigue / CG to prevent recordLast() - this may be refined later
- GUI: re-write of the Controls tab to incorporate multiple stop criteria
- GUI: minor code cleanup (deleted commented code segments)

- GUI: edited log writer to output list of stop criteria and values
- GUI: corrected stop criteria adjustment feature to prevent resetting of cycle / time values on every setConfiguration() command
- GUI: eliminated extraneous oddballValueCheck() command at the end of setConfiguration()

May 30, 2025

v.20250530: Mid-cycle stoppage, peak recall, disp exceedance

- BIN: set dynamic load adjustment to OFF by default
- BIN: adjusted test "wrap up" (softStop, recordLast) subroutines to be able to stop mid-cycle if frequency is under 1 Hz
- BIN: locked idx_max/min_disp to idx_max/min_load
- BIN: made peak indexing reset optional based on user selection (flag is new element in DATA_22)
- BIN: added displacement to checkCompletion(); will stop test if outside ADC range of 0-65535
- BIN (ADG): adjusted crack length calculator to send two records of data during recordLast()
- BIN (ADG): updated crack length and K interpolators to match ADX subroutines
- BIN (ADX): built in framework for "slow start", moving cyclic error control to point-by-point instead of cycle-by-cycle (NOT YET ACTIVE)
- GUI: changing frequency now un-checks "pause peak indexing"
- GUI: "remember indices" gives user the option to have the GUI remember the peak index value for load and displacement upon test restart.
- GUI: peak indices will always be remembered and re-applied after peak indexing auto-converges. Also works on recovery.
- GUI: warning built in to prevent / guide user from running K control at start of test (intent is to start under load control and then switch)
- GUI: added da/dN vs a; both internal and external plot windows

Apr 29, 2025

v.20250429: Reverted PID to 0326 (cycle based)

- BIN: due to control issues, the PID controller was reverted back to the cycle-based correction subroutine used 20250326 and before.
- GUI: k_p limit to da/dt lifted (was part of new controller approach)

Apr 28, 2025

v.20250428: PID correction, C(T) da/dt fix

- BIN-ADX: corrected the PID routine by setting lowdacval and amplitude to FLOAT from LONG, thus preventing smaller (< 1.0) changes from being filtered out.
- GUI: fixed error that prevented the da/dt option from being selected for C(T) specimens
- GUI: set upper limit of k_p to 0.25 for da/dt

Apr 24, 2025

v.20250424: Exceedance overload and PID fixes

- GUI: changed default segment overwrite selection to "No"
- GUI: adjusted color profile and fixed label for HUD

- GUI: controller will now erase queue and stop all queue updates on first sign of exceedance (goal is to prevent continued cycling seen in CSS tests; behavior seems improved but real life testing needs to confirm)
- BIN-ADX: moved cyclic fatigue PID control from start of cycle to spread over 20% of the waveform to prevent step loading behavior. (ADG will need to account for this procedurally)

Mar 26, 2025

v.20250326: HUD re-write

- GUI: changed all lineEdit entities in the HUD to labels, in hopes of preventing the "phantom digits" we keep getting.

Mar 24, 2025

v.20250324: Data "wrap up" subroutine fix

- BIN-ADX: fixed last-line output in logStatiistics() from "pdcomp" to either compliance or active PD (the pdcomp variable in DCPD was actually normalized and we don't want that)
- BIN-ADX: re-wrote the output "wrap up" subroutine on test completion to simply issue the logStatistics() command and get the current state instead of running one last crack length calculation (this was VERY problematic)
- BIN-ADG: re-wrote the above for ADG but did it by jumping to acalc_step 6 to 9 and then jumping out. Had to do this because the ADX way took up too much space.

Mar 20, 2025

v.20250320: Heads-up display

- GUI: first iteration of HUD incorporated, with run / stop controls linked to the main GUI UCP

Mar 19, 2025

v.20250319: HUD framework, LCF spec schematics

- GUI: introduction of the HUD; however given time constraints it is not enabled in this release
- GUI: added LCF specimen schematics (smooth, notched, compliance, A/RPD)
- GUI: all external plots now close through the menu, eliminated closing by "X"
- GUI: LCF cyclic SS plots now output even when "None" is selected for PD / compliance
- GUI: changed sig figs of current displacement values output from 5 to 3.
- GUI: added Debug entry for calculated amax to check no load increase behavior

Feb 27, 2025

v.20250227: Automatic static output writer

- GUI: added automatic SEGDATA write command for static tests
- GUI: added self. variable to log file for adc_mid and adc_resolution to allow access from the Log thread.
- GUI: fixed SS output file to calculate actual loads and displacements

Feb 24, 2025

v.20250224: Static fix

- BIN: added fix to ADX and ADG binaries to prevent the static test from "running away", i.e. overloading. Result of adding the "record last line" subroutine for cyclic tests.

Feb 21, 2025

v.20250221: Compliance ratio fixes

- GUI: initialized original idx_max_load values to prevent phase plot error on restart
- GUI: outputting rolling average CR value for compliance tests
- GUI: changed dashboard CR precision from 3 to 4
- GUI: command to initialize CR and compliance now uses instantaneous calculations, not the RA

Feb 7, 2025

v.20250207: Cyclic SS data output, new change list

- GUI: updates in compliance load % text field limited to K control, Kmax mode
- GUI: notch net thickness updates in the GUI when send command is pressed
- GUI: blanked out notch depth field for smooth test types
- GUI: blank out OP for cyclic LCF
- GUI: check on loading conditions from file when LCF specimen is selected (CG can't be selected for LCF specimens)
- GUI: free up None measurement option for LCF
- GUI: shut off XML file output write command when None selected
- GUI: displaying load ratio in Response Field in dashboard (same as controller value)
- GUI: updates K as well as load in the K / load calculator in "Calculators" tab
- GUI: "Send" and "Receive" buttons are now different colors to stand out
- GUI: added a "Clear Threads" option to see if that helps fix some of the hangups. We'll see.
- BIN / GUI (ADX ONLY): last 10 LCF cycles are captured in output csv file
- BIN: included crack length measurement from last set of slopes
- BIN (ADX ONLY): included last incremental measurement as independent line of data (must be manual for ADG)
- BIN: purged initial compliance and CR with purge() command

Jan 8, 2025

v.20250108: Cyclic LCF additions

- GUI: added four new specimen types: Smooth Cylindrical / Rectangular, Notched Cylindrical / Rectangular with lineEdit options de/activated accordingly
- GUI: expanded XML write / save feature to save LCF data; now has the same format as CG data
- GUI: changed color of RUN so it turns green ONLY while running.
- GUI: deleted some of the complianceCalc code that was commented out for a long time; better readability
- GUI: fixed error that prevented test recovery due to an improperly initialized variable (initial idx_max_load); resulted from checking peak index pause too early during recovery
- GUI: set scale for da/dN plot

- GUI: re-separated the buttons for updating E and initial compliance because you have to wait until the rolling average for compliance settles before initializing.

Jan 3, 2025

v.20250103: peak indexing fix

- BIN: set initial ADG peak load and displacement values equal to ADX (1/4 and 3/4 of total waveform length)
- BIN: edited to bypass the above and use previously averaged peak index if running an appended segment
- GUI: corrected peak index pause feature to 1) run while in cyclic LCF and 2) actually work properly, activating and checking only when a crack length measurement is made (this is also done in LCF but not displayed).
- GUI: corrected error in log file generator that did not shut off threadlock after completion (prevented static test from plotting)
- GUI: no longer plots peak index value for dwell or static cycles

Dec 18, 2024

v.20241218: GUI crash fixes, Quit button deactivated

- GUI: added numerous thread "FROM / TO" print statements to track what thread the program is in (all commented out for release)
- GUI: added thread wait command to beginUpdates() function in updateLogs()
- GUI: eliminated self.main_win.ui.textBrowser_log_file.clear() command because it wasn't needed and it crashed the GUI due to a sharing violation.
- GUI: fixed thread wait command in mountThreadLoad() run subroutine (was an "if" test not a "while" loop)
- GUI: isolated the getDataBlock() subroutine with its own threadlock value within mountThreadLoad(). Now GDB() runs with threadlock 4, resets to 0 when done, returns to MTL() and goes back to threadlock 1.
- GUI: fixed thread wait command in getConfiguration(); was an "if" test not a "while" loop
- GUI: fixed purgeData() threadlock value from 8 to 9.
- GUI: disabled the "X" close button on the main window. Now the user must select File --> Quit

Dec 16, 2024

v.20241216: GUI change request list

- GUI: fixed (again) no load increase on K-dec test to enable on a finite non-zero % difference between target and actual K
- GUI: changed significant figure count in load display
- GUI: program no longer quits during an invalid initial license check
- GUI: moved some of the commented out debugging code into another file
- GUI: more prominent, always visible “run/stop/purge” buttons
- GUI: change units for sy and E
- GUI / BIN: added UTS as passable parameter (not used for flow stress stop criterion yet)
- GUI: eliminated splash screen option from controller menu
- GUI: fixed splash screen to prevent user from hitting start before beginning the controller process
- GUI: peak indexing now locks and goes to manual after convergence
- GUI: peak index now shown on load and displacement plots

- GUI: controller metadata populated with default values on start
- GUI: initial compliance and CR values automatically updated with new modulus
- GUI: pre-test current option disabled for compliance
- GUI: OP_flag queried from the log file, user warned if improperly selected upon segment re-start / append
- GUI: specimen properties re-arranged in controller
- GUI: max stress / strain updated before compliance measurements to fix "infinite compliance" issue on restart
- GUI: specimen dimension files now includes geometry and will update when new dimensions are loaded
- GUI: hyperpd checkbox fixed to remain disabled at end of test for compliance specimens
- GUI: "tab-through" order fixed (can run through the buttons and options by hitting "tab" and it's in order now)
- GUI: set default log / data directory when user selects folder
- GUI: log window only updates when a new average crack length is measured, not for every crack length calculation
- GUI: fixed field highlight error for specimen dimension values
- GUI: fixed bug that prevented index maxima from updating after changing waveform length

Nov 14, 2024

v.20241114: settable R^2, new error handling

- BIN: added lower R (correlation) limit variable

- BIN: eliminated CheckSignals() subroutine in favor of license signal check during CheckDACSaturation()

- BIN (ADG ONLY): changed max_avg_load declaration to Float from Long (now consistent with ADX starting 0510)

- GUI: new error flag for correlation exceedance

- GUI: re-organized error flag popups so the dashboard thread ONLY triggers during the initial license check and the remainder triggers from the logwriter thread.

- GUI: NOTE: these fixes should also address the observed popup and log exceedance displays

- GUI: separate line item in dash for a/W

- GUI: swapped position of max and min loads

- GUI: added extra command to update log file after test stoppage to show user possible error message

Oct 14, 2024

v.20241014: updated license generator

- GUI: updated the license subroutine to insert information to indicate whether it is for an ADG or ADX. This will disable the device dropdown menu after the initial run to get the serial hash and prevent accidental operation using the wrong device configuration.

Oct 11, 2024

v.20241011: min compliance %, phase plot fix

BIN: correlation coefficient corrected to be consistent with FTA (eliminated extra sqr root)
BIN: initial compliance min % value set to 50 from 70. User must make sure to change any saved test conditions!!
GUI: corrected (AGAIN) phase and compliance offset plots. The phase plots were still coupled such that offsetting the peaks equally in load and displacement would result in a different phase relationship between the two.
GUI: fixed issue where the XML file generated post-recovery did not have the a or K lookup tables (generated at end of recovery algorithm.
GUI: fixed bug that allowed aux channel option to activate upon completion of an ADG test.

Sep 27, 2024

v.20240927: AUX channels, set / return load fix

- BIN, GUI: added two AUX channel inputs via IN3 and IN4
- BIN: fixed average cycle count to be integer only (necesary for data recovery file)
- BIN: added fix to amplitude and lowdacval for return load; LDV now follows the commanded return load, and amplitude goes to 0
- BIN: changed stop criterion based on low corr. coefficient (R) to only work if measured and commanded loads have converged.
- BIN: added NEW cycle counter that tracks cycles from the last start command. Needed to prevent premature compliance or DCPD calculations when ramping up to load after restart during the same segment.
- GUI: added live and CSV data updates of AUX channels on the Controller and Data tabs
- GUI: edited log file writer to better describe the line item outputs (crack length calculation vs. data output); also added N_total along with n_segment
- There is a known issue where the XML file generated post-recovery does not have the a or K lookup tables.

Aug 30, 2024

v.20240830: Fix to "Return After Complete"

- fixed return subroutine to prevent it from un-checking when done.

Aug 29, 2024

v.20240829: No load increase, /ADG freq. display fix

- GUI: fixes the frequency display for the ADG; it was multiplied by the VA mission_freq_factor that was not present in the ADG binary
- GUI: fixed check box or No Load Increase on K control to only work during a K control test and only after the K values match

Aug 27, 2024

v.20240827: VA input revision, error flags, 5V ADG output

- GUI: changed VA input file reader to eliminate the need to the NASGRO check box. It now automatically detects a comma or space delinited file.
- GUI: added back the license check to the the runaway_error_flag in AutoUpdateMount because it happens early enough in the program that there's no log file yet.
- BIN: set Digout(5) (CONN2 PIN 16) to ON for 458 switch power source in the ADG controller only

Jul 17, 2024

v.20240717: Updated log files, compressive fatigue

- GUI: reset loads and load ratios to 1000#, 0 for QTP subroutine
- GUI: edited log file to dump latest info from dashboard upon test completion
- BIN: changed binary to run cyclic fatigue using C(T) as reference and output compliance data (did not originally work in compression)

Jun 26, 2024

v.20240626: SFB a/W increase, LUT display fix

- max a/W value set to 0.6 (from 0.15) for SFB
- edited precision of LUT tables to properly display unique PD / complaince table values

Jun 24, 2024

v.20240624: Current correction, IACS added to get/set

- BIN: increased max waveform length to 20000 to allow for reasonable pulse times for TMF
- BIN: IACS added to controller for get/set commands
- GUI: corrected the current calculation for SFB (had incorrect numerator value and did not update with the correct value of VPRC denominator)
- GUI: also added a new subroutine to calculate VPRC specifically for current crack length value
- GUI: set max pulse width to variable (remains at 5000)
- GUI: added .txt format to waveform file loader options

Jun 14, 2024

v.20240614: K calcs for bending (C-C), longer HPD cooldown

- GUI: moved exceedence error messages from autoUpdateThread() to updateLogs() allowing it to send error messages AND write to the log file if there is an error that results in test termination
- GUI: addition to oddballValueCheck() to prevent positive loads being used for bending tests
- GUI: fixed Return command error that would measure current during the Return state (would yield an error if specimens were switched before returning to 0 load)
- BIN / GUI: added hpd_snapshot[] array to track values of the HPD pulse in the GUI (feature limited to ADX for memory considerations)
- BIN: K calculation changed for bending specimens to use min compressive load for Kmax
- BIN: extended hpd_cooldown wait criterion to 4X (from 1X) before capacitor charge error triggered

Jun 7, 2024

v.20240607: Compressive loading HPD fix

- eliminated full compression restriction in oddballValueCheck() to allow for SEN(B) / SFB tests
- moved the reference point for IH shutdown in ADX to idx_min_load for R > 1
- changed the starting point for HPD in the ADG to start at either idx_max_load or idx_min_load depending on R

Jun 4, 2024

v.20240604: Interpolator and crash recovery fixes

- changed location of generateWaveform() in GUI to only run at startup if not recovering from a crash; otherwise it runs at the end after everything is restored (this seems to be the cause of the load exceedences we were seeing on recovery)
- verified behavior of GUI in crash recovery for FALSTAFF (truncated), mission, short mission, spectrum, and arbitrary load point waveforms (constant amplitude) ~20x
- fixed interpolation routine for crack length and K; had an error that wouldn't go back to the first interpolation point for small variations in PD or a/W (would be locked at aw_i or kp_i >= 2)
- set condition on completed segment cycles to only reset for a NEW segment; if appending an existing segment it will resume from the current cycle count

May 24, 2024

v.20240524: GUI initialization streamline, PD load fix

- BIN: commented out the purge() command in startup ... didn't seem to do anything and it would make things worse if it did
- BIN: ADX BUG FIX - in the previous version the max and min loads used as a basis for telling the code when to measure PD was set once at the start of the test. This could eventually prevent PD measurements from being taken if the load migrated (through K control) to lower values and falling outside the initial comparison load range. A new update command was created to execute once per cycle.
- BIN: re-wrote the defines.inc file for ADG to match ADX; also added mission_test variable
- GUI: moved the UCPDropdown() command into getConfiguration(); this is needed if the GUI starts in a static test since the dropdown menu technically doesn't get "changed" if you stay with the default settings
- GUI: changed the order in getConfiguration() to get the test type value and then immediately change the menu selection
- GUI: changed gain menu subroutine to reset the menu selection along with the value
- GUI: fixed log restore command to not search for a/K LUTs for cyclic fatigue or static
- GUI: edited the log file writer to include LUT info even for cyclic fatigue and static tests (to prevent search errors)
- GUI: added array for G, n, c into binary for storage; will be used for the generation of the LUT plot / table widgets
- GUI: overall streamlining of GUI initialization process, various fixes to recovery subroutine; now the recovery tool does not send anything to the device; it only reads.
- GUI: re-wrote the LUT plot and table generator to allow for a user commanded "preview" based on the GUI text fields and then an automatic "actual" set of tables based on the lookups sent to the controller immediately prior to test.

May 21, 2024

v.20240521: Charge flag reset, new return location

- BIN: re-wrote the interplation function for crack length to check the limit states first and then loop through the LUT after
- BIN: significantly restricted the runaway error flag for crack length; should never get into that case unless there is an error in calculation.
- BIN: fixed the charge_flag variable; it had been set to 1 for debugging and was not shut off
- BIN: moved return_after_done variable assignment to immediately before actionComplete() to see if it prevents it from shutting off (ongoing investigation)

May 18, 2024

v.20240518: Revised HPD, recovery subroutine

- BIN: re-wrote entire HPD subroutine to better coordinate IH shutdown for cyclic and dwell testing
- GUI: modified software recovery routine to restore mission cycle information in the block builder
- GUI: shortened a/W limit for SFB to 0.15 from 0.6

- there is an outstanding issue with spectrum loading that causes exceedance upon restart

May 10, 2024

20240510: Unified mission / spectrum loading (NO EXE)

- GUI: oddballValueCheck() changed; error flagged if abs(R) > 1 and load > 0 but should only be R (not abs(R))
- GUI: updated K plotter to allow for da/dt, dwell CG
- GUI: eliminated the OBVC that prevents negative static load rates for nonpositive min loads
- BIN: changed the checDacSaturation() subroutine for static testing to shift the threshold value to BELOW the min target if min is positive
- (the two above changes allow for the required step loads at a user specified rate)
- BIN: set max load to abs(max load) in error check in cycleControl()
- BIN: fixed compliance segment reader (in ADX and ADG) to properly cycle around if counter goes above or below waveform range
- BIN: added check in the ADG MUX to switch from PD to load channel when setting up the test and going back and forth from PD / compliance / none
- BIN: changed the criteria for n_begin_counting for mission cycles, basing it on new variable total_subcycles_completed
- BIN: set a lower point limit for compliance calculations to 5% of the total waveform size
- BIN: because we are (for the time being) using a "V" waveform for missions I had to add another idx_max_load for the first half of the cycle to properly bookkeep the compliance segment
- BIN: moved the rx_snap[] readers to outside the linear compliance subroutine to get them to keep reading regardless of whether the code is calculating crack growth
- BIN/GUI: completely re-wrote the mission block subroutine to use the half-cycle approach I used with the spectrum loading
- BIN/GUI: checked validity of compliance and PD using spectrum, long/short missions, truncated FALSTAFF, and standard constant amplitude loading

Apr 22, 2024

v.20240422: Spectrum / mission ok, compliance, PD ok

- again, another placeholder with no executable
- spectrum and mission loading built in
- verified PD and compliance calculations

Mar 25, 2024

v.20240325: Draft of mission block VA (NO EXE)

- GUI / BIN: incorporated basic mission block capability with transition enabled by linear variation of lowdacval and amplitude
- NO UPDATED EXECUTABLE for this release. This is just a placeholder / restore point so I can start looking at spectrum loading

Mar 12, 2024

v.20240312: Fix to import / export load sequence

- GUI: the command to export load sequence saved the file with point count and interpolation type saved as float, not int, causing a read error when trying to re-import it.

^^^^^ VERSION 1.2 BRANCH (VARIABLE AMPLITUDE) STARTS HERE ^^^^^

Mar 11, 2024

v.20240311: Fixed init_PDcomp, % stop criterion

- BIN: algorithm to initialize and reset PD0 has been changed; original formulation didn't work as intended, making parts of the code redundant and possibly leaving it open to an errant initial crack length calculation.
- BIN: fixed DCPD algo to allow simcrack to work (finally)
- BIN: fixed ADG error in the cyclic DAC command that allowed the initial value of the waveform multiplier x to take on an initial value of 0, thus preventing any form of cycle endpoint control. Why this wasn't a problem in the ADX with literally the same logic is beyond me but ok. It's fixed in both now (xx instead of x to differentiate from the utils argument of the same name, and it's initialized in init() and setStepandRange())
- BIN / GUI: moved reset_pd0 from INT to FLOAT (it is now the new PD0 value instead of a flag so we can specify where and when in the crack length calculation the update takes place).
- GUI: added "normalized PD" under the active and reference PD update values (act / ref / pd0)
- GUI: fixed the log writer (again) to better inform user of changed values and initialized PD
- GUI: added feature to import / export spectrum load points into the load point tab.
- GUI: changed load point feature to only use numerical values for interpolation (instead of radio buttons)
- GUI: changed the % PD / compliance stop criterion to % OVER, not % of original
- GUI: fixed dashboard display of normalized PD to only display when refPD has a value (otherwise it divides by 0)
- GUI: added rolling avg of CR within the code (does not display)

Mar 6, 2024

v.20240306: QTP and soft stop cycles

- GUI: fixed QTP test sequence (unneeded loops)
- BIN: reverted soft_stop cycle count to 5
- BIN: commented out an unneeded DIGIO command for both ADG and ADX that simulated a 458 signal

Mar 1, 2024

v.20240301: da/dt crack growth, HPD capacitor check

- GUI: fixed log file writer to incorporate other crack growth types (dwell, da/dt)
- GUI: corrected plot / data tab labels from "Compliance" to "PD / Compliance"
- GUI: fixed LOG file to update in GUI before start of data collection
- GUI: added pre-run capacitor charge check; controller will not complete run command if capacitor not charged.
- GUI/BIN: incorporated da/dt crack growth as test type 5
- GUI/BIN: incorporated "% of initial PD / Comp" as stop criterion for cyclic fatigue tests
- BIN: fixed SUSPEND TEST command so it works for both dwell and cyclic tests, but only if freq < 1 Hz.
- BIN: changed HPD to prevent pulse prior to collection of voltage artifact
- BIN: incorporated capacitor charge exceedance timeout limit of one period

Feb 22, 2024

v.20240222: Dwell crack growth, compressive PD

- GUI/BIN: incorporated dwell crack growth as test type 4
- GUI: changed calculation of process frequency in the "Device Status" window to use actual process freqency rather than freq * wf_length
- GUI: added initial slope count to saved parameter file
- BIN: fixed DCPD subroutine to allow for PD measurements during compression-compression testing (automatic polarity switcher for load range check)
- BIN: changed calculation of HPD cooldown time to incorporate the difference in process frequencies of each dwell segment

Feb 19, 2024

v.20240219: New log writer

- GUI: revised the log file writer to include crack length and cycle count interval sizes / flags
- GUI: additional changes to UpdateLogs to streamline output formats, making it easier to read larger volume of crack growth measurement data
- GUI: added specimen_index to complianceCalculator(); this sets the foundation for being able to change the order of the specimen dropdown menu without throwing everything into chaos.
- GUI: fixed label error in SFB specimen interface to read "Diam." instead of "Radius"
- GUI: fixed mis-spelling of "Inner Raduis"
- GUI: added non-alt text for pulse width description
- GUI: check against SFB specimen geometry to prevent XML writer from generating lookup tables in the output
- BIN: added logdata_flag to every crack length measurement command; all crack length measurements made within an interval are now recorded, provided they are calculated at a rate of 2 Hz or slower because that's the refresh rate of the dashboard.

Feb 9, 2024

v.20240209: Added SFB, fixed FBH, slope limits

- GUI: added bending surface flaw specimen option
- GUI: added SFB to the QTP list
- GUI: fix oddballValueCheck() for Hyper capacitor (no longer stops user from sending conditions if capacitor is not charged)
- GUI: change to setConfiguration() to prevent a non-empty oddballValueCheck() result from resetting all pending changes back to their original values (i.e. I commented out the getConfiguration() command at the end)
- GUI: added comments on y1, y2, PD gain and eliminated the perpetually problematic "alt text" that never ever worked.
- GUI: fix to segmentRecord and updateLogs to record outer and inner radii for FBH and SFB specimens
- GUI: selected dimension labels re-named on selection of the SFB specimen (B, Bnet, outer/inner r, SOW, etc.)
- GUI: re-labeled "SF" to SF (Tension) to match SEN
- GUI: changed warning in updateMountThreadLoad ("a/W exceeds 0.xx") to account for possibility that a/W limit may not be the same for all specimens
- GUI: check box to allow crack length interval measurements during cyclic fatigue testing now disabled
- GUI: swapped the "type" and "number of cycle points" labels ro correspond with the correct menu options
- GUI: added auto E modulus correction action to the log file
- BIN: applied limits of 1-4000 to crack length calculation slope count

^^^^^ VERSION 1.1 BRANCH STARTS HERE ^^^^^

Jan 11, 2024

v.20240111B Version 1.0 Hotfix

- changed location of run/stop exceedance check
- changed message for run/stop exceedance

v.20240111 Version 1.0 release
- GUI: truncated the K LUT plot to a max of 500 for easier reading
- GUI: divided ref PD plot by gain to match the dashboard readout
- BIN: placed additional ADC assignment command for the ADX in the setRXVals() subroutine to track updating of refPD gain
- BIN: moved run/stop error check to updateCycleControl()

Jan 8, 2024

v.20240108: HPD voltage fix, ADX defaults

- updated min and default pulse strength to 20 from 10
- changed text for PD gain dropdown menus
- added HPD DAC command to set voltage back to ADC_MIN if HPD flag = 0
- set default splash menu selection to ADX from ADG

Dec 22, 2023

v.20231222: DAC12_1 LED signal output

- sends a signal to DAC12_1 (Pin 27 in ADX) when LED 1 is on

Dec 20, 2023

v.20231220: K plot, spec config file fixes

- corrected time calculator for K control plots (was giving lower elapsed time than actual test time because the refresh rate is actually slower than 0.5 sec)
- added header to test queue files
- fixed bug that did not update max load and R when switching from K to load control (may still be buggy; we'll see)
- fixed the ALUT coefficient file to no longer overwrite n or C coefficients (they don't get read in anyway for the crack length calcs)
- completed ALUT, KLUT library check
- changed device number to work with ADX on install

Dec 8, 2023

v.20231208: Load / Save test queues, QTP

- added feature to load / save test queues
- completed LUT library QTP check (IDE ONLY)

Dec 6, 2023

v.20231206: Min load calc, M(T) and NR K fixes

- fixed specimen configuration file for crack length lookup table to no longer set A coefficients (for K calculations) to zero.
- fixed Johnsons reference PD calculation for M(T) which divided by the wrong width dimension (W, not W/2)
- fixed graphic for SEN pin loading K solution (actual subroutine equation OK)
- fixed graphic for SF FTA K solution (actual subroutine equation OK)
- changed display of G, A (a calc and K solution) coefficients to allow for more significant digits
- fixed Newman Raju K solution f(W) term; supposed to be 2B, not B
- changed Newman Raju phi value to pi/2 from 0.
- min load display fixed after soft stop
- initial a, K LUT library check subroutine built in; can only run from IDE at this time.

Nov 27, 2023

v.20231127: Fix in auto queue

- added command in the queueing subroutine to set ctrl_k0; otherwise it would use the current value of K in the dashboard
- corrected instances of ESEN(T) to ESE(T)

Nov 10, 2023

v.20231110: Live updates, thread consolidation

- added user-controlled live updating of data plots
- consolidated the queueing thread with the dashboard thread (dashboard calls everything now)
- introduced "blankflag" variable which tells oddBallValueCheck() to only check blank values when you're submitting something to the queue or if it's running.
- added feature in the text field format updater to insert a zero value for all fields that are left blank after editing

Nov 6, 2023

v.20231106: BFS fix, GUI formatting update

- added default closure code (2) to the xml postprocessor input file writer
- added Params.crack_interval to the list of test items with 4 sfigs for GUI formatting check
- eliminated the check on test / spec type that would multiply compliance by W for back face strain (now taken care of through IN2 calibration factor)

Oct 30, 2023

v.20231030: K calc after a calc, logging edits

- moved the K calculation subroutine so it is only called after a new crack length is calculated
- edited logging subroutine for controller to make the recording of condition edits consistent with the original formatting so it can be read by the postprocessor log file importer
- edited the logger to eliminate duplicate parameter names to simplify searches
- corrected thickness calculations for C(T) specimens (ONLY) in binary and GUI to use sqrt(B * Bnet) instead of just B
- corrected initial LUT generator values for SF / SEN PD solutions to account for larger LUT size and thus smaller alpha increments
- corrected FMID variable assignment that caused GUI to create log / data file with the segment value in the "Test Control Options" text field instead of the test queue field
- corrected time request expression for static test condition to prevent negative values during compressive loading
- changed setConfiguration() so the K control values (K0, R0, a0) are set BEFORE the command to change control type to K-control. This prevents an occasional exceedance glitch in which the controller receives the command to switch to K control from the GUI but doesn't have a K0 value to work with since the GUI hasn't gotten to that command yet. This happens more in the ADX since it's faster but will happen in the ADGold too.

Oct 6, 2023

v.20231006: Interval based K updates

- K now updates in the GUI based on the crack length or cycle interval, not after every crack length measurement.

Oct 2, 2023

v.20231002: PDcomp calc interval fix, soft stop fix, etc.

- fixed bug that allowed the slope count to continue accumulating after post-interval adjustment even though compliance was reset to zero upstream
- fixed bug in the way the controller treats soft stop (it still automatically set to -1 for static and dwell, then it got re-set to 0 during setConfiguration(). Had to check against test type.
- got rid of the development tab in the menu bar
- got rid of some of the "debugging" features that are no longer relevant in the DEBUG tab
- added "target load" plot variable to show new loads in cases where control is shut off
- added try: / except for majority of subroutines to make debugging easier
- added descriptive comments for all subroutines in win_mainwin.py
- restored CR (compliance ratio) rolling average

Sep 27, 2023

v.20230927: Divide by zero fix for LR compliance

- added check on number of points flagged for compliance calculation to avoid division by zero.

Sep 26, 2023

v.20230926: Data record option, soft_stop / return changes

- edited "soft stop" and "return after complete" so they are no longer instant and require user input to send to device
- added "previous_N" so user can properly set the cycle interval real-time
- set and fixed the GUI window dimensions (975 ht)
- added DAC status indicator and changed label for command voltage to "Last" when not issuing DAC commands in the wait state
- fixed specimen type selection to disable opening load when going to PD / none for crack length measurement

Sep 22, 2023

v.20230922: Compressive static, state cycle fix

- controller can now run static testing in compression
- added "Min Load" text field to facilitate compressive loading; only active during static testing, otherwise updates automatically.
- enabled 3-pt averaging ONLY for cyclic tests; otherwise disabled and set to 1 for dwell / static
- fixed RETURN to use tx_avg instead of tx_load which didn't work when rolling avg was shut off
- fixed log file data index tracking error (did not reset when appending new data to existing file)
- fixed log file date / time display for test stop / restart (you can actually read it now)
- re-fixed soft stop amplitude calculator (had been accidentally reverted in the 0619 release)
- added fix to prevent controller from auto-running if there were outstanding issues with the input
- removed the debugging command from the cycling subroutine that was accidentally left in the previous release
- compliance % now updates during CG cycle testing regardless of control method (load, K)
- fixed reset of compliance % values on shutdown of cyclic PD cycles

Sep 19, 2023

v.20230919: Interstitial compliance for dwell

- edited log writer to only send changed parameters on a "send to device" command during test
- set up waveform offset function to set 0, 25, 50, 75% offset from mid load
- get rid of the interpolation for regular waveforms (can still do it in the arbitrary waveform builder)
- re-wrote NOTES section and put it below the wave generator
- fixed initialization of pulse width to not show a format change on initialization
- incorporated the non-iterative compliance calculator to dwell for ADX and ADG
- fixed dwell cycles so that they may use time as a stop criterion (used dwell_time_remaining as new variable to measure time remaining for each dwell segment)
- corrected the HPD cooldown period, basing the wait time to sum of ramps and holds in one cycle
- corrected incrementation of counter for dwell based compliance measurements (was off by 1 in the initial calculation)
- changed the equation for elapsed time to use 1 / PF for dwell

Sep 13, 2023

v.20230913: OP for ADX, segment appending, etc.

- opening load capability built in for Adwin-X; placeholder code also available for interstitial compliance in dwell (not incorporated yet)
- fixed data formatting error that would prevent multiple appended segments from being properly formatted in the CSV data tab
- fixed LUT generator subroutine to prevent exceedance upon GUI recovery
- added soft_stop, dyncal, and return_after_done to test parameter save file
- changed hpd pulse strength spin box from instant to send-on-command, and will now change format when edited
- set PD0 to automatically update if the PD values are within a certain threshold
- lowered ADX PF limits to account for the new compliance and OP calculators; OP is now 100k, PD is now 250k
- changed interpolator subroutine to assume x[1] is always < x[max] for compliance offset values
- changed phase adjustment commands in the GUI to show ACTUAL index max values (should be easier to interpret)
- re-ordered GUI tabs to allow user to move to adjacent text fields

Sep 6, 2023

v.20230906: PD artifact correction

- PD artifact had been set to zero in previous releases starting in 0825 (interstitial compliance) due to an oversight after test simulation; it is now restored to the correct calculation.
- adjusted the average of the secant compliance so slopes are calculated individually and then averaged
- added op_count_missing to prevent OP from being calculated in cases in which one or more segment slopes is not available due to low (< 0.8) correlation value

Sep 4, 2023

v.20230904: Interstitial OP (initial)

- re-incorporated opening load capability into ADGold based on same strategy as interstitial compliance.
- max PF for OP is now only 15000 to prevent signal "wobble"

- frequency of OP calculations is based on slope count interval / 4 as long as slope count interval > 8
- compliance, OP calculations, interpolation all verified using a comparision of ADbasic calculations and Python scripts applied to user data
- fixed the load-displacement phase spin box (sign of phase difference was reversed)
- incorporated "3 strikes" rule to compliance, shutting down test if R < 0.9 3X in a row

Aug 27, 2023

v.20230827: PD slope count, peak indexing fixes

- moved PD0 initializer check into crack length calculator; pdcomp_count now resets when slope count is reached even if no crack length is calculated (e.g. when PD0 = 1)
- shuts off / deactivates 'pause peak indexing' every time controller enters wait state, preventing controller lockup.

Aug 25, 2023

v.20230825: Interstitial compliance

- new compliance subroutine using nested IF tests and non-iterative linear regression. (AD Gold only)
- same approach used for secant compliance, eliminating the need for the comp_offset[] arrays
- crack growth calculations also done interstitially allowing same-cycle crack length results after reaching the slope limit
- overall speed limit decreased to 370 Hz @ n=100 in ADGold
- replaced static loading subroutine; can now handle multiple tensile loading segments, returns, etc. (compression still WIP)

Aug 24, 2023

v.20230824: Recovery / return fixes

- initialized peak index values at start of every segment (fixed controller lockup when changing waveform lengths by large amounts)
- fixed recovery of cycle interval, crack interval, refPD, and hyperPD flags on restart
- fixed "return after complete" flag to no longer de-select when finished returning
- added formatting changes to inform user either control or recording is paused
- changed DCPD selection options to prevent loading / unloading check boxes from being de-selected; also de-activated OP check box for DCPD

Aug 17, 2023

v.20230817: Removed speed optimization

- removed speed optimization changes from 0811 code (it had caused an occasional freeze in the crack length calculations)
- restored original OP PF limit to 20k

Aug 14, 2023

v.20230811: Speed improvements, csv table fix

- fixed the in-GUI da/dN calculation to restore multi-segment plotting / listing capability
- decreased linear-regression based compliance calculation time by factor of 2X (for 10 Hz, n=1000, no OP time decreased from 32 sec to 15 sec)
- as a result of the changes implemented to speed up the compliance calculations, a fix was required to the SoftStop() subroutine to shut off compliance calculations during the last 5 cycles.
- dropped OP PF limit to 35000 because of workload exceedance

Aug 9, 2023

v.20230808: Cycle averaging correction, pulse width fix

- added process frequency / limit to the dashboard
- changed the formatting of the pulse width values if out of bounds
- allowed the GUI to display invalid pulse width values to show the proper calculations, but retained the restriction against submitting them to the device.
- added checks against non-numerical values related to the creation of the a / K lookup tables
- fixed the runaway accumulation of cycles for DCPD tests (relocated summation variable n_pdcomp from a compliance-only location)
- fixed current switching cycle averaging accumulation; when current switching, pdcomp_count only increments when measurements in BOTH current directions are complete, thus making it smaller by a factor of 2x
- reset n_pdcomp at the start of a new segment (corrected initial cycle counts)
- there is a known bug that prevents multiple-segment data from displaying in the CSV tab; this is WIP

Aug 4, 2023

v.20230804: Frequency selector update

- added new security features to the frequency selector and preliminary value checking subroutine to prevent process frequency exceedance
- fixed charge flag in ADX to read digital input (had set to 1 for debugging)
- changed default test mode to cyclic crack growth
- restored original K updater
- set field updater to work during cycling
- updated dashboard with new PF values

Aug 3, 2023

v.20230803: Slope adjustment DEBUG window

- created new DEBUG window to track crack length data, calculation count, interval value, and code location within the slope modifier subroutine to track root cause of slopes not changing for rkk
- modified "additional options" checkboxes to make all of them both "real time" and recoverable, i.e. checking the box will immedately change the value of the global variable within the device and will also show up in the GUI upon program recovery.
- same for cycle interval and crack interval selections. HOWEVER there is a known bug in which the queueing system will not change the cycle or cracklength selections on its own, instead defaulting to the user selection in the dashboard (e.g. if user selects 1000 cycles and .001" for N and a respectively but has "cycle interval" de-selected, the binary will not record data based on cycle interval).

Aug 1, 2023

v.20230801: U_x backsolver, da/dN calc fixes

- re-wrote lookup table subroutine for C(T), M(T), and ESEN(T) to backsolve for u_x in the crack length equation. This allows for the creation of regular crack length intervals in the lookup table sent to the binary and makes them identical in functional form to the PD LUTs
- changed da/dN calculator in the GUI to use a[i] - a[i-1] vs a[i+1] - a[i-1] (which is what the numpy.diff() command does), thus making it identical to the secant da/dN calculator in the binary
- changed data type of n_pdcomp (variable that accumulates the cycle count for every compliance measurement) from Long to Float because it resets when working with high cycle counts (> 2.E09).
- fixed crack length averaging routine bug (was reset to 0 on on reset instead of the last measurement made during the previous interval)

Jul 28, 2023

v.20230728: Average of averages

- first full implementation of "average of averages" technique to smooth crack growth response.
- increased size of lookup tables from 200 to 2000 to reduce "stepping" in the da/dN calculations (temporary measure)
- now calculating the average cycle count along with average pd / compliance for individual crack length calculations. This prevents the da/dN curve from "jumping" if the slope interval changes.

Jul 21, 2023

v.20230721: Return fix (again) , minor fixes

- hard coded convergence criterion for cyclic load matching to 1%
- changed default load exceedance to 50% from 2%
- precision of output data (to csv widget and XML) increased from 3 to 5
- deleted A, G, n, C coefficient tree widget
- disabled the compliance / HPD / calculator option when running a static test
- changed extrema checking of the static test subroutine to match the cyclic test routine
- controller now outputs data for cyclic fatigue (either with or without compliance selected)
- "Return" command fixed so the "Soft Stop" user selection is preserved
- GUI check to prevent static load rate from exceeding load range
- outstanding issue with static loading: trips exceedence on unload (i.e. static load from max to zero); still a work in progress.

Jul 7, 2023

v.20230707B: K updating, current switch fix

- fixed error that did not properly update target Kmax prior to the start of a test
- re-zeroed PD bookkeeping variables when selecting / de-selecting current switching
- shut off K calculations during soft stop
- created option to clear rolling average

v.20230707: Act/RefPD calculation fix

- eliminated the absI() commands from all active and reference PD calculations (it zeroed them all out).
- disabled TEST PAUSE option (pending fix)

Jul 6, 2023

v.20230706: Fixed softStop subroutine

- the SoftStop() subroutine as previously written broke the static, dwell, and post-static return functions
- this routine is now hard-coded to run only during cycling.

Jul 5, 2023

v.20230705: Fixed PD calculator

- fixed the PD averaging calculator to no longer swap the ADC polarity before taking the average.
- also using the absolute value of the PD value when averaging.

Jul 4, 2023

v.20230704: Updated rolling avg, exceedance check

- added rolling average selector for load peak finder / exceedance check (spin box)
- completed migration of A, G, n, C coefficient values to test fields
- changed "Commanded" load values in the laod displacement plot to the DAC value instead of applying the load and R value to the device waveform array.
- fixed soft stop for CCPC loading (the mean value was incorrectly calculated)
- fixed load exceedance check for CCPC
- load averaging is now applied to peak finding and load exceedance checking.

Jun 29, 2023

v.20230629: Rolling average for exceedence

- uses 3 point average to check for load exceedence in RX and TX values.
- framework for lineEdit A, G, n, and C values instead of tree widget (not yet functional)

Jun 22, 2023

v.20230622: Modified rxval snapshot algo

- added feature that takes snapshot of last analog input / digital output cycle before exceedence (using the saturation check subroutine in binary). This is separate from the "data grabber" feature in the GUI as it guarantees capturing the last cycle regardless of frequency.
- changed data grabber subroutine to automatically stop upon control exceedence so user can "set and forget"

Jun 21, 2023

v.20230621: Added DAC to analog output script

- added the command output to the data grabber
- added Stop button to halt collection on command
- corrected subroutine to check for presence of PD and ignore if running compliance only

Jun 20, 2023

v.20230620: Soft stop, cycle interval fixes, analog input

- fixed the soft stop option so it automatically re-checks after the return command
- fixed the cycle interval to properly update after a positive crack growth rate increment under the cycle interval criterion
- added csv output generator for analog input variables load, disp, and actPD

Jun 19, 2023

v.20230619: updatable cycle and crack length intervals

- code now updates the crack length and cycle interval values during a test
- edited softStop() to ignore returns to load
- changed the PD comp calc interval adjustment routine to use a percent based adjustment (25%) instead of a direct calculation
- adjustment routine checks to see if the crack length exceeds the lower bound before adjusting the slope count.
- added the ADX bootloader

Jun 14, 2023

v.20230614: SEN(B) Pmax correction

- fixes target load calculation for SEN(B) in the "calculators" tab
- increased max frequency to 10000 Hz
- removed k_i and k_d debug values from binary

Jun 12, 2023

v.20230612b: HPD debug - new tracking parameter

- added a tracking variable to follow the binary's way through the crack length / k calculator

v.20230612: HPD Debug Release 2
- Adds K calculation in the debug output
- resets the debugging text fields before updating to try to track which variable is updated last

Jun 9, 2023

v.20230609: HPD Debug Release

- adds new information in the DEBUG tab to track progress and internal variables related to the rolling average plotter during HPD

- changes the state variable query for the "state" global variable from the device controller to the auto_update thread
- first to add the ADwin-X binary but since this is an interim debug release this part of the code is not ready for use.

Jun 7, 2023

v.20230607: SEN(B) current calc fix

- SEN(B) current calculator updated to latest version. No other changes.

Jun 6, 2023

v.20230606: Initial ADX, soft stop fix, comp-comp fix

- fixed log file writer to incorporate FBH specimen type
- added math for "average of averages" approach (not yet active)
- edited splash screen to add the ADwin-X option (will boot but binary not yet active)
- limited RefPD gain to 2X for ADwin-X (device limitation)
- combined load displacement plots into a single plotting subroutine
- fixed exceedence check to allow for compression-compression tests
- fixed compliance ratio % value correction to only apply when |R| > 1.
- fixed slow stop to converge to proper load value

Apr 28, 2023

v.20230428: Data output fix, new waveform load/save

- changed waveform load / save to use one column of data instead of one row. Makes it easier to read.
- edited soft stop routine to restore original "pre-stop" load ratio (the sub sets the load to P_avg and then R = 0.999 to wind down)
- WAIT frequency changed to 1000 Hz from 200 to get a better pic of the PD signal
- shut off device query fix until the PD signal issue is sorted out (it also caused data output issues)

Apr 24, 2023

v.20230424: Dashboard refresh rate fix

* had to go back and restore the dashboard refresh rate to 0.5 seconds
* did this to prevent the queueing system from thinking the test was done before it even had the chance to update its status from "wait" to "cycling".

Apr 19, 2023

v.20230419: ESEN(T) fixes

- fixed the "specimen type" menu and log file entries, which had previously been adjusted to account for the fact that we had not yet implemented a working ESEN(T) specimen
- background: ESEN(T) was built into the original code but omitted when it was not yet put into the same format as the other specimens. It was not taken out completely, rather it was "skipped" so that the menu

options went from 0 directly to 2. This was done so we could easily restore the option when it finally was built in properly. Unfortunately this was not done the right way when we put it back in, so this is the fix.
- corrected crack length calculation coefficients for ESEN(T); if user switched from front to back face, they took on the incorrect values
- added reference PD capability for ESEN(T)
- corrected K solution menu option for ESEN(T) from C(T) to ESEN(T); this would only cause a problem if the user switched K solutions within the same specimen geometry.
- enabled DCPD radio button when crack length lookup table selected for all specimen that also use compliance.

Apr 6, 2023

v.20230406: ESEN(T) and dashboard freq change

- incorporated ESEN(T) into controller
- changed dashboard timer to update at rate of 1 Hz (from 2) to check PD behavior

Mar 23, 2023

v.20230323: PD0 update, minor GUI fixes

- fixed log file reference error for "refpdgl"
- fixed bug in controller that did not update PD0 when it was set for the first time (!= 1)
- created groupBox to more effectively lock compliance / pd / hyperPD selections during test run

Mar 15, 2023

v.20230315: 3-wire, soft stop, max/min detector

- converted SF and SEN ref PD solutions to 3-wire from 4-wire
- equations updated for 3-wire solutions
- all references to PG gauge length changed from GL or y to y1 to make it consistent with reference PD equations
- displacement calibration field set to 4 decimal places
- soft stop set as default
- "hard" stop changed to force tests of ALL stop criteria to stop by cycle coutn when complete to force the load to stop at mid-load.
- soft stop fixed to make sure load is properly set to mid-load (was not always happening)
- created max / min detector for entire test run with optional reset.

Mar 9, 2023

v.20230309: Spec images, SF back face, PD fixes

- specimen images available along with the a/K equations, plots, and lookup tables.
- back face correction incorporated into SF solution as a standalone selection
- PD hangups caused by large changes in load or target K values fixed by putting a check in the number of PD measurements being made to get the average values (> 1).
- Error in setting target K for C(T) fixed (eliminated what was basically a circular reference).

Mar 3, 2023

v.20230302: OP / interval fixes, rolling da/dN for min

- both crack length and cycle intervals after either target hit
- implemented rolling avg (n=5) for min da/dN but NOT for max da/dN
- changed order of da/dN calculations so they are in the same place and reference the same crack length
- fixed OP calculator; had been dividing by 2x the OP calculation count

Feb 22, 2023

v.20230222: n_start lower bound, check box disabling

- added lower bound to the number of cycles required to start compliance calculations for cyclic tests to 50 (fix for opening load)
- deleted temporarily repurposed debugging variables accidentally left in the previous code
- disabled "no load increase" check box before initial compliance measurement
- disabled RUN check box when running
- added proper OP load range calculations in the case of inverted cycles.
- put in crack length / cycle interval da/dN calculations for RKK

Feb 17, 2023

v.20230217: Fix to life stop criterion, OP

- live updates to stop criterion / stop value fixed for all test types (did not update properly)
- OP waveform length change error (system crash when waveform length changed between segments) fixed by re-initializing the load range array prior to every segment.

Feb 13, 2023

v.20230213: Batch get, soft stop

- put the GUI option switches into getConfiguration command
- set up soft stop option
- organized GUI to use batch queries
- reorganized the global vars to transfer the spinbox values for pulse width and pause time
- set the PF limit on HPD to 2000; otherwise it outpaces the mux switching and we don't get accurate ADC values.
- added feature to highlight text fields upon parameter and HW configuration loading (i.e. they load to the GUI but are not sent to the device)

Feb 1, 2023

v.20230201: Init. V/Vref, FBH, UserPoly

- repaired broken K LUT for SEN
- incorporated "initial V/Vref", which is identical in functional form to PD0 but for tests that run with reference PD. Capability added to all specimens.
- updated images to represent use of initial PD/PDref

- "user specified polynomial" for crack length calculation removed; user still has the option for those geometries that have a polynomial as their crack length equation.
- added flat bottom hole (FBH) specimen type
- added r_outer and r_inner for FBH calculations
- changed "user defined" to "determined from PD lookup table)
- disabled y1 calibration (did not yet remove from GUI ... waiting for RKK buy-in)
- reset response fields for better formatting

Jan 25, 2023

v.20230125: SEN(B), C(T) PD, global var arrays

- moved some (FLOAT) dimensional variables to an array to free up global variables (B, Bnet, W, E, syield, notch depth, alpha, initial compliance, initial compliance ratio)
- moved some (LONG) "check box" flag variables to an array for same reasons.
- generalized the Johnson's equation to use "2b" from literature, but ...
- ... the displayed equations in the controller now use W only to avoid confusion (so it's W for MT specs, 2W for single edge geometries)
- Gauge length adjustment feature (i.e. the PD analog to auto-emod correction) is fixed and can be used for SF and SEN specimens
- waveform limit set to 5000; process frequency restrictions unchanged.
- SEN(B) K solution available
- Johnson's C(T) crack length solution only available for PD0 type tests
- New FEMM based crack length solution for C(T) with reference PD available (only for PDref tests)
- Reference PD MUX gain (1, 2, 4, 8) available

Jan 13, 2023

v.20230113: HPD add-ons, C(T) PD, etc.

- k_p and frequency text fields rounded properly
- fixed bug that prevented dashboard from updating at fractional frequencies
- Johnson's solution replacing Hicks Pickard for C(T); reference PD feature for this geometry is functional (i.e. does not throw an error) but is not calculationally correct
- Villari effect "artifact" calculated
- OUT2 channel now charges HPD capacitor; spinbox for power percentage on HW configuration page
- status field in dashboard lets user know HPD capacitor charging status
- current calculation feature; works with the PD Gain dropdown menu in HW configuration page
- M(T) crack solution fixed (errant factor of 2 resulting from wrong file being loaded)
- PD and compliance calculation status now shows for cyclic and dwell fatigue

Jan 3, 2023

v.20230103: Period to Pulse, OP ADC

- introduces manual period to pulse ratio calculator, allowing the user to set the time in between pulses (default set to 10000x)
- changes ADC measurement for opening load from READADC12( ) to ADC( ) for better accuracy in offset calculations.

Jan 2, 2023

v.20230102: Dcharge, Dfault

- added two flags to the code for HyperPD:
- Dcharge: turns on to alert ADWIN when capacitor voltage is within a tolerance of target voltage.
- Dfault: digital fault output. Normally low. High indicates fault/failure in pulse switch circuitry.
Breakdown of DIGIN values (CONN1)
- DIO_0 (Pin 1 bit 0): input signal
- DIO_2 (Pin 2 bit 2): Dcharge
- DIO_3 (Pin 15 bit 3): Dfault

Dec 23, 2022

v.20221223: HyperPD, reference PD

- first phase of hyper PD incorporated into PD system (does not yet measure current)
- pulse width spinbox allows for user o determine length of HPD pulse
- reference PD built in for SF, SEN, and M(T) specimens
- gauge length correction feature for SF and SEN (at this point only works for SEN as it's a direct calculation; the other needs some work)
- plotter fixed to account for intermittent reading of PD vs. load

Dec 17, 2022

v.20221217: PD / Set_Mux fix

- framework for HyperPD built in (GUI) but subroutine not available.
- Set_Mux and channel switching fixed for cyclic PD. Previous version caused interference in the load and PD outputs making the signals very noisy. Now switches on a per cycle basis to avoid the "teeth" observed the load and PD outputs.
- Dwell PD fixed to switch Mux every 5 *data points* (not cycles) to maintain control for every cycle during dwell. I don't think we can get away with a whole dwell without PID.
- Plotter adjusted to account for the fact that PD dwell outputs are missing some points as a result of the channel switching during hold. Both load / disp and PD are filtered based on the Set_Mux counting system internal to the binary.

Dec 9, 2022

v.20221209: PD for dwell

- adds PD measurement for dwell fatigue cycles (does not yet incorporate crack growth)
- OP check box disabled for dwell
- cycle data read in before Case() statement to determine state; allows command to only be run once, saving program space (166-> 159 kB).
- "hold" cycle eliminated; not used.

Dec 7, 2022

v.20221207: M(T) geometry

- introduces the M(T) geometry with compliance and PD consistent with ASTM E647.
- this is preliminary and not yet tested against actual data; however output files seem to load into the postprocessor with no problem.

- correction added to "Load Parameters" to eliminate loading the modulus (done elsewhere) and requested _cycles (unneeded).

Dec 2, 2022

v.20221202: Added C_ry optional K gradient

- incorporated C_ry as option for K gradient. Dropdown menu provides option between this and E647 standard.
- fixed bug that did not provide proper load comparison for compliance in the unloading direction when load calibration < 0.
- global fix to GUI to minimize the number of device queries that have been identified as the root cause of the GUI-side crashes. Monitoring thread now reads in a number of global variables that are used more than once into a set of variables, and then are made available for other subroutines to access them instead of the device.

Nov 30, 2022

v.20221130: Speed optimization

- spread out calculations for linear regression based compliance to take place over more cycles, but with less overhead.
- re-established secant compliance as a separate subroutine using the original "idealized input" approach for calculating compliance index offsets.
- reset the speed limits for various measurements: OP and linear compliance = 40k, secant = 60k, PD = 40k, and max speed limit is 65k
- fixed frequency dial to seamlessly transition from linear compliance to secant and back again if the PF passes through applicable threshold.
- added status indicator showing the type of measurements being made (linear compliance, secant, PD, none)

Nov 21, 2022

v.20221121: New compliance calculator

- re-wrote compliance and OP calculation routine; no longer uses the compliance index offset calculator in the GUI based on an idealized sine wave, but rather the actual loads to determine the proper "data windows" for compliance and OP calculations.
- Fixed the adaptive compliance window calculator to force a "buffer" in the lower load limit if the min load is very near - but not quite equal to - the min compliance percentage (think R=0.699 and comp% = 70%)
- secant compliance ("quick compliance") now integrated into the linear regression subroutine as an option based on process frequency. Basically it will grab the data, figure out where the compliance endpoints are, and then skip down to the secant calculations without doing all of the linear regression. It will still calculate CR though so you can get ACR with secant compliance.
- edited dwell to use the same adaptive compliance as cyclic testing. Dwell CG not yet built in.
- "cycle with hold" cycles have their extrema calculated slightly differently, as the compliance index offset array is no longer being used. Now uses the "interp_points" and "invert_flag" global variables to indicate the presence of a hold as well as its relative position in the cycle (max or min)

Nov 9, 2022

v.20221109: OP based on load percentages

- incrementation of OP segments based on LOAD percentages, not on indices
- OP range limited to 5% - 95% total load range (down from 1% - 98%)
- fixed min crack length limit formatting
- fixed bug that prevented output for cyclic and dwell fatigue
- fixed bug that didn't purge the previous list of data segments during PurgeAction()

Nov 4, 2022

v.20221104: Test pause fix, LUT plots

- fix to the "pause test" feature that resets the amplitude of the original control cycle. This is necessary to prevent overload when resuming. (We set min = mean during pause, so when we resume, we immediately go to mean + amp instead of original min + amp. That's bad.)
- added preview window to show plots and tables of the equations used to calculate a and K

Nov 2, 2022

v.20221102: Hicks Pickard, PD outputs

- built in Hicks Pickard PD C(T) crack length solution
- dashboard / live plotter updates to include active and reference PD
- changed plotter windows to avoid updating if not selected
- check on alpha lookup tables to prevent non-monotonically increasing behavior
- corrected specimen index values so they properly translated to the postprocessor (C(T)=0, SEN=2, SF=3; E(T)=1 but is not in the menu anymore); this created a problem in the postprocessor.

Oct 28, 2022

v.20221028: CSV waveform output (RP)

- interim release for RP use
- changed wording of SUSPEND TEST to something a little less dramatic
- added smoothed accel to acceleration plot
- added option to output displacement / velocity / accel / smoothed accel to csv
- corrected a/W limit to 0.7 from 0.75
- allowed user to turn on PD current BEFORE test
- no live PD update yet

Oct 26, 2022

v.20221026: Adjusted LUTs, license script

- re-wrote LUT generator subroutines for SF and SEN to allow for direct calculation of normalized PD where available.
- program will now warn user that a/W exceeds 0.75, and will issue a STOP command when the end of the LUT is reached.
- binary now uses the actual LUT size, not the default limit size
- new plot to GUI showing acceleration based on displacement (calculated in multiples of g)

- warning provided in the licensing subroutine to tell user that the count is either >100k or <2 (connection error)
- count is still limited to 999 so we'll adjust this if needed.

Oct 16, 2022

v.20221016: Default directory correction

- corrected default load / save directories to avoid saving files to wrong location or overwriting existing files.
- corrected name of crack length calibration for SF specimens

Oct 10, 2022

v.20221010: Test pause feature

- test pause feature added; PID set to control on mid load
- line edit formatting re-incorporated to let user know of edits that need to be sent to the device.
- added formatting feature to waveform generator
- waveform invert bug fix (triangle waveform wouldn't invert at certain sample sizes)

Oct 5, 2022

v.20221005C: Peak Indexing Fix

- default peak index values corrected
- option to pause peak indexing deactivated until there is at least one compliance measurement
- option available to pause compliance / PD calculations
- bookending algorithm now works for crack lengths that go below lower bound

v.20221005B: Live feed of CR data
- allows a live feed of the compliance ratio calculation data
- shortens the license calculator to one loop only.

v.20221005: Hash License Key
- sets the "start process" button to enable only once after "Boot" is pressed; otherwise it's disabled and greyed out.
- K gradient editing is enabled (when applicable)
- the loop counter licensing system is set up
- all parameters related to the compliance ratio will be displayed if and only if it hits infinity. (I think. The manual says infinity is pegged to some ~10e38-ish value but I'm not 100% sure if it's still interpreted as a number I can use for comparison. This may change.)
- now using only one binary instead of two based on device hex values. They seem to work locally as long as the proper device is specified in the "device_number.txt" file.

Sep 30, 2022

v.20220930: Unloading OP, da/dN stop criteria

- binary calculates OP for loading and unloading ... but still compares to unloading standard compliance ONLY.

- min da/dN stop criterion corrected to prevent a check on the value if there's nothing there yet
- also prevents either a check or data record action if the value for the interval da/dN <= 0
- changed label for load stop criterion to "max load" for clarity.

Sep 28, 2022

v.20220928: Consolidated Code

- consolidated cyclic test subroutines to free up memory
- last point in da/dN is different color for better visibility

Sep 27, 2022

v.20220926B: Updated Installer

- updated the installer for latest version

Sep 26, 2022

v.20220926: Commercial PyQt5 / Qt5 Builds

- now uses the commercial versions of Qt5 and PyQt5; no other changes.

Sep 22, 2022

v.20220922: Adaptive index point adjustment, etc.

- compliance measurement window adjusts itself based on current stress ratio, calculated explicitly using trig functions for const. Kmax / var DK
- new compliance offset percentages sent to the device and updated in the GUI with user message if they fall outside load range OR if they violate ASTM E647 for OP
- waveform generator and send / receive disabled while test running
- checkbox for no load increase persists on recovery
- data plotting displays for combined OP / non-OP segments fixed
- data recovery properly parses data from all segments in current device test history; records kept in log files
- "data snapshot" csv file properly formatted based on OP / non OP status; only shows the last segment
- R squared evaluation of compliance for linear regression
- legend for K control plot available, COLORS SWAPPED TO BE CONSISTENT WITH OTHER PLOTS
- intermediate save to XML feature
- a vs. N plot available
- per-calculation compliance offset plot added

Sep 1, 2022

v.20220901: Kgrad debug window, compliance fix

- fixed bug in the compliance range calculator that by default included the point BELOW the min compliance percentage range.

- fixed bug in the rolling peak index averaging calculator that used AbsI(), which always rounds DOWN to the nearest integer no matter its value, to Round() which rounds either up or down to the nearest integer value. The combination of these two fixes puts the compliance data windows in the right place.
- added output in the DEBUG tab to show the values of K0, r0, a0, and current a for K gradient bug checking.

Aug 31, 2022

v.20220831: Cable lock, more verbose errors

- incorporates a larger number of / more verbose error messages for GUI subroutines to better track the I/O error (and others) that's still popping up.
- popup window for process frequency exceedence
- log and data folder group box disabled during test
- cable lock built into binary with backdoor in GUI.
- color palette menu on splash page

Aug 26, 2022

v.20220826: OP subincrement fix

- fixed OP subincrementation error that would average compliance over incorrect number of data points.
- improved offset interpolation speed for OP calculations to bring speed limit up to process frequency of 20k

Aug 23, 2022

v.20220823: Fixed previous release (GUI)

- apparently the location of the .qss file for the custom GUI was in the wrong place for the executable to properly find it. It's better now.

Aug 18, 2022

v.20220818: Initial OP, new color scheme

- Incorporated .qss file to create customizable visual template
- Added opening load capability (needs testing)
- New CSV parser to account for multiple segments with / without OP selected
- Corrected error in static test logging that prevented csv output updates during test.

Aug 5, 2022

v.20220805: Initial DCPD, back face strain

- C(T) specimen geometry now has back face strain for compliance calculations
- initial DCPD capability incorporated, can select reference PD, current switching
- still needs testing to check quality / formatting of output.

Jul 28, 2022

v.20220728: Fixed stop criteria tracking

- fixed a bug in the code that was using the wrong values for tracking the stop criterion based on crack size or da/dN values.

Jul 27, 2022

v.20220727: Incorporated PyArmor and expiry

- Same code as the 20220725 release, but compiled using PyArmor to obfuscate the code to complicate decompiling and set an expiration to 12:00:00 AM on Friday 7/29.

Jul 25, 2022

v.20220725: Add'l thread consolidation, etc.

- code reads channels IN5, IN6 concurrent with IN1, IN2.
- fixed updateMountThreadLoad (the now-primary GUI thread) to use the device value of the load calibration factor instead of the text field.
- converted the log thread (updateLog) to a class, and is now referenced by the mount thread.
- added features to properly change stop criterion mid-test and record correct control info in the segment record if a change is made.

Jul 15, 2022

v.20220715: Recovery fix

- Uses the log file to get all of the general specimen info (FMID, contract, etc.) so we don't have to worry about copying over the header file during the recovery process when moving from one version of the executable to another.

Jul 14, 2022

v.20220714: Threadlock correction, etc.

- fixed threadlock to always shut off at the end of a test (caused lockup otherwise)
- fixed the LUT layouts so they scale properly during specimen selection
- changed LUT file selection to deactivate when test is running
- additional error traps for subroutines to better inform users

Jul 13, 2022

v.20220713: Thread locking, error notification

- consolidation of three threads (updateThread, updateMount, updateLoad) into a single thread to avoid conflict
- incorporation of "threadlock" variable in device to prevent more than one thread from accessing data at the same time

added thread status in GUI, along with the option to start them by hand if restoring the program by hitting "start" versus "recover"
- put in additional error checks for data recovery; should still go to the GUI even if there's no data to recover.

Jul 12, 2022

v.20220712: Add'l safety features, fewer threads

- placeholder in the event my messing around with combining threads goes terribly wrong
- de-activated dropdown menus for spec type, a cal, k sol during testing to prevent changes during tests
- added information to the log file for the postprocessor to read during recovery
- deactivate the LUT "clear" button when running.
- eliminated the lockReturn() thread, placed in updateThread() instead.

Jul 6, 2022

v.20220706: Phase Plots, Disabled Text Field Formats

- disabled text field re-formatting that indicated field value changes to see if this prevents code crashes
- edited splash to better guide users to proper process flow
- replaced fi with cyclic_max_load (temporarily) to give some insight for RKK exceedence issues
- added phase plot and manual shift of max index when peak indexing is paused
- made default and starting value of waveform points equal to 1000

Jun 23, 2022

v.20220623: Rolling avg. based Auto E

- uses a rolling average calculation to provide a more stable basis for calculating a corrected E value
- same concept applied to PD to get PD0
- incorporated list and chart for PD / compliance tracking
- LUTs will update if when a@PD0 changes,
- reset some static test variables during purge() to potentially address RP issues with "bump" during second return.

Jun 16, 2022

v.20220616: Dynamic correction adjustments

- changed location of dynamic correction to run only during compliance / pd averaging.
- fixed dynamic correction to update even if not applied
- eliminating E(T) option from GUI (for now)
- limit C(T) crack measurement option to compliance only, SF / SEN to PD only.

Jun 15, 2022

v.20220615: Dynamic calibration, log fixes

- PD lookups built in and have been verified with external data

- fixed bug that prevented dwell value fields from opening up when changing test type
- multiple segment data now formatting properly
- compliance value calculation corrected for dynamic calibration; now includes phase value that accounts for the range represented by the max and min compliance percentages.
- fixed error in data logger that prevented updating.

Jun 7, 2022

v.20220607: Minor fixes for compliance

- edits to prev_a / next_a bookkeeping to better follow calculated crack length
- fixed the check on K-dec. test to prevent load increase (load may still vary based on rx load but the command will not increase)
- incorporated return to set load after test
- PD libraries built in for SF and SEN (not complete yet)
- KNOWN ISSUE with plotting data after testing static and then cyclic. Work in progress. For now test them separately after purging data in between.

May 27, 2022

v.20220527: Corrected dynamic calibration, etc.

- streamlined method for getting K target value from the GUI
- set convergence criterion to load only (not convergence on K)
- set exceedence exception flag based on change in target load due to change in target K
- fi calculations more efficient; will trigger exceedence exception flag only once based on difference between previous and current fi.

May 25, 2022

v.20220525: Small error trap fixes

- set "startup error" check based on magnitude of initial load (prevents very small but unchanging error from triggering exceedence) This may go away if we add a "dead man switch" into the system ... we'll see.
- added a check on whether the control is paused to determine whether switch to turn back on error trapping is triggered.

May 24, 2022

v.20220524: Cleaned up / added new error traps

- fixed bug that output crack length interval data regardless of user option.
- fixed crack interval update (again) - start and end interval wouldn't update after correcting the modulus
- fixed bug in which code would start calculating changes in DK without having actual crack lengths yet.
- added error trap to prevent runaway loads immediately upon startup
- fixed bug that stopped static test prematurely because it tripped the static IIR filter; changed logic.

May 20, 2022

v.20220519: Autonomy / Recovery

* autonomous device based crack length calculations and K control
* crack length measurements split up over multiple cycles for time efficiency
* PD, K/P lookup tables created for SF, SEN
* new recovery mode allows user to restart / recover test data
* dynamic correction fixed; option to show load cell load response in GUI
* improved (hopefully fixed) exceedence checks
* K gradient field disabled during test
* added "elapsed time" field
* KNOWN ISSUES:
* cycle 'lag' at frequencies over 100 Hz.
* cannot run constant Kmax test after any other test; premature test stoppage
* static loading test will stop because it will trip the static IIR filter.

Apr 27, 2022

v.20220427: Fixed "no load increase" for K-dec.

* fixed issue in K-dec subroutine that led to no change in load depending on selected conditions.
* also has (commented out) calculations for device-based crack length and K

Apr 25, 2022

v.20220425B Same but with installer

* oops

Apr 24, 2022

v.20220425: Error trapping, recovery

* confirmed K mode control can go between load control and K-inc/dec control.
* error trap to stop command when load exceeds user specified error percentage
* enabled recovery from overload without complete restart.
* user option for no load increase during K-dec. testing (based on a_max, RKK will test)
* data plotting during state_wait and state_cycle
* added (placeholder) PD calibration factor entries into GUI
* fixed output display for B, Bnet, W, CRI, EVBPI
* fixed gauge length field shutting off for compliance testing (unnecessary for computation)

Apr 15, 2022

v.20220415: da/dN Plotter fix

* fixed da/dN plotter (was deactivated)
* fixed log file writer to prevent writing when there is no crack length calculation
* changed the rate of updates based on subincrement calculations
* fixed "stop Action" button to work for cycling without CG

Apr 13, 2022

v.20220413: Crack length subincrements, etc.

* corrected K gradient formulation to match ASTM E647 8.6.5.1
* new method of bookkeeping crack length intervals based on subincrements.
* both crack length interval and cycle interval counts will be checked at the end of each subincrement and will update Data[] appropriately
* direct calculation of "calculations per subincrement" value to limit between 10-20 per crack length interval
* GUI now supplies instantaneous da/dN calculations based on subincrements
* various variable renaming for clarity
* fixed output to log file to avoid repetition
* fixed the da/dN v DK plot to get the proper loads applied (was using Pmax, not pmax[i] based on the assumption of a changing load for K control)
* static test no longer starts at some small nonzero load after return
* DCPD framework set up, not tested

Mar 30, 2022

v.20220330: Emod low freq fixes, etc.

* changed name of external windows to "display"
* fixed Emod display to return to normal formatting
* new da/dN window
* new load / disp and command / received phase plots (internal and external)
* user can now send post-Emod compliance values over to GUI
* Auto Emod butons deactivated until the crack growth calcualtions are updated
* removed notch length from Auto Emod *only using ai, which includes notch)

Mar 28, 2022

v.20220328b: Added 0x190 Bootloader

* Earlier version didn't include this ...
* ... plus some updated GUI details.

v.20220328: Ext. Windows, Highlights, etc.
* User can select external windows for Load / displacement, K plots.
* Changes to text fields and dropdown menus are now highlighted until sent to ADwin
* Option to automatically send Auto Emod results directly to Emod field
* fixed display for cycles / crack length remaining
* load decrease only for negative C
* load calculator for given K value
* User can shut off command control (cycle but no change in condition)

Mar 22, 2022

v.20220322: Auto E

* incorporates iterative modulus calculator to get proper Ai (please test)
* send / receive parameters on front page now functional.

* other minor formatting fixes

Mar 21, 2022

v.20220321: Multiple Fixes

* added setConfiguration comments, crack measurements into the log file
* added check box to PurgeData
* fixed error in log file writer that didn't reset the tracked number of data points in Data[] when reset
* Moved call to sub that saves default folders to GUI, fixing the Quit menu bug
* fixed cyclic fatigue -> cyclic CG - crack interval checkbox activation bug
* option to shut off data recording
* added frequency spin box with user specified resolution
* log file now records shutting off data recording, pausing index averaging, send to device, and updating frequency with the spinbox
* fixed error that kept min load from being calculated when changing max load or R during a test
* "pause index" deactivated after Send to Device

Mar 17, 2022

v.20220317: Kmax output fix

* fixes output to Kmax that prevented K control from working
* also has initial framework for DCPD (not yet funcitonal)

Mar 15, 2022

v.20220315: Crack Growth Intervals

* user may now specify either cycle or crack interval (or both) for data output.
* checked K gradient; appears to work properly.
* fixed segment letter update in controller page during testing
* fixed "I/O Error in closed file" error
* added check on zero load rate for static testing
* added error catching for leaving crack and cycle intervals unchecked.

Mar 10, 2022

v.20220310: Corrected Data output page

* moves the formatting of the cyclic tests to the device, allowing a single format to be used in the GUI for cyclic and static test results.

Mar 8, 2022

v.20220308: Static Ramp Fix

* fix to static ramp function due to problem with CalculateADCFromLoad() not giving proper results when passing a sum (a+ b)
* also fixed the time_remaining to match the updated ramp duration resulting from maxload-minload

* fixes the data segment selection dropdown menu.

Mar 7, 2022

v.20220307: Partial "revert" to max load

* restored the original max_load static equation
* eliminated min_load calculation
* using ramp_up_duration again
* does this work? If not, then there's something else I did to break the static test.

Mar 4, 2022

v.20220304b: updated optional min load

* fixed due to internal update subroutine resetting min load.

v.20220304: optional min load
* now has a check box for option inclusion of min load
outputs min load on debug tab

Feb 25, 2022

v20220225: Corrected Commit

* Same stuff as below but the last commit didn't have everything (??)

v.20220225: Improved Static (DEBUG)
* Static now starts at current (mount) load.
* switched to cyclic return command for all return functions (versus a separate one for static)
* fixed cyclic return (wouldn't work if waveform was larger than 100)
* return gives non-zero load option (and you can just set the load this way)
* dynamic correction language streamlined (internal)

Feb 21, 2022

v.20220221: DEBUG version of below

* allows for debugging of the static loading test.
* also fixes the max load lineEdit to represent commanded load, not dynamically corrected.

Feb 10, 2022

v.20220210: Installer script, bug fix

* Now has an installer executable so all you have to do is download that and you're all set.
* also fixed a default folder bug (missing backslash)

Feb 9, 2022

v.20220209: Unified Control UI

* New GUI, completely re-arranges options and command flow.
* Major re-write of the test segment command manager.
* replaces the old cycle command list and incorporates all test types: static, cyclic fatigue, dwell, and FCGR
* test data now outputs log files (for all) and XML files (for FCGR postprocessing)
* fixed K control to use correct loads; now the PID works much better

Dec 24, 2021

v.20211224: End of year dev placeholder

* DO NOT USE
* this is just a placeholder to make sure the work done so far is saved to the server.
* features will be described in the official commit next year.

Dec 2, 2021

v.20211202b: Okay ONE check box

* puts back the dynamic calibration check box to allow user to see its value before deciding to turn it on / off

v.20211202: No more check boxes
* all of the monitoring options are now on
* same for dynamic calibration; if the user wants it deactivated, they can set the inertial weight to zero.

Nov 23, 2021

v.20211123: Optional index suppression

* allows the user to stop max_load index averaging for compliance measurement.
* it will still calculate internally, but not update until user re-selects it.

Nov 15, 2021

v20211115: No Amplitude Reset

* eliminates the amplitude reset that forces the PID to re-establish the cyclic amplitude every time a new cycle block is started.
* adds a flag to ignore the 1.1 * max_load exceedence check for cyclic loading until the new amplitude is stabilized.

Nov 10, 2021

v.20211110B: this time with the proper executable ...

v.20211110: Additional Completion Criteria
* adds min da/dN, max da/dN, and crack length to FCGR completion criteria.
* this is NOT in the same section as the fatigue testing tab and is therefore not part of the list / queue maker (yet)

Oct 30, 2021

v.20211030: Dwell, Newman Raju

* Adds dwell cycling capability plus corresponding compliance calculations
* Adds Newman Raju K solution for surface flaw crack type

Sep 23, 2021

v. 20210923: Fixed State Return, Load Exceedance, XML Parser

* Fixes State_Return() to return based on the values of txvals, not rxvals. (RP request based on FBH work).
* Last column of controller output now has max load for the output block (replaced f_i for now)
* Completed XML postprocessor, but it's not compiled yet; will probably generate the first executable in an independent git release.

Aug 31, 2021

v.20210831: Re-compile using Python 3.8 32-bit

* No other relevant changes. Will be using only Python 3.8 32-bit for Win7 and WIn10 compatibility.

Aug 30, 2021

v.20210830: Increase to load tolerance, 32- and 64-bit .EXEs

* Increase of load variation tolerance to 1.10 from 1.08; this is the only functional change from the previous version.
* There are also 32- and 64-bit versions for different OS versions.
* Introduces GUI options for SEN and SF specimen plus all of the crack length calibration / K solutions that go along with them (not yet functional)
* Does NOT have controller interface for DCPD yet.
* Also does NOT have da/dN - dependent compliance average variation.

Jul 22, 2021

v.20210722: Fix to State_Return()

* Change to state_return() function to allow a return from negative load.
* Minor cleanup of extraneous code
* Change to initialization of compliance / OP variables before state_cycle() ... functionally identical for standard compliance calculations.
* changed sum and avg loads to long from float
* both 0x190 and 0x150 are for USB devices

Jul 16, 2021

v.20210716: K control restart after overload

* Allows restart to K control tests after overload / user stoppage.
* Properly incorporated Purge routine into K control tests
* Blackout of GUI buttons depending on test state to avoid confusion.
* Gives TX output for static tests in "Monitor" section
* Fixed a load calculation error when starting new cycle block

Jul 10, 2021

v.20210710: Fix to static output and max load threshold

* Static load output was incorrect because of leftover debugging code.
* overload threshold increased to 1.05 from 1.01.

Jul 9, 2021

v.20210709: Improvements to overloading

* Binary will now enter state_wait when load exceeds max for static and cyclic testing (versus state_stop, which stopped the binary altogether).
* This only works for fatigue testing; K control testing still requires restart.

Jul 2, 2021

v.20210702: Minor fixes; kp_static, rolling avg, CR fix

* adds kp_static to the hardware config import / export file
* makes the max index rolling average weight equal to compliance_cycles_avg.
* properly initializes compliance_ratio to avoid overflow on start

Jun 30, 2021

Improved / corrected compliance calculations

* code now gets p, d data from the same cycle to minimize scatter during compliance calculations
* floating point based rolling average with weight of 1000 to settle on average index for max load
* corrects compliance calculation with proper fi
* incorporates compliance ratio calculation and outputs it in Data[]
* adds kp_static for static loading

Jun 14, 2021

Overload protection (initial)

* 1st draft of fix to prevent load from exceeding commanded "maxload" value by 1% (does nothing for min load).
* has exception for when user drops target Kmax value in K control tests

Jun 11, 2021

New compliance calcs to fix frequency lag

* Compliance calculation now done inside stateCycleAction() on a per-point basis to avoid a significant end-of-waveform lag.
* (Actually there's multiple stateCycleAction()s now, depending on where you are in process_freq space)
* Performs linear regression at lower frequencies (process freq < 50k) and secant at higher values (>50k, < 75k)
* OP capability removed for now.
* Corrected dynamic calibration to subtract |fi| from max, add to min in waveform.
* Two-device capability still intact for now.

May 13, 2021

First draft of OP capability, corrected device number options

Created two binaries: one each for 0x150 and 0x190 (don't use any other device number). User must change device_number.txt in the dist\FLcgs\adwin folder.

Incorporates a first draft of the opening load calculations.

May 7, 2021

User Specified Device Number, OP calcs (commented out)

Provides a text file "device_number.txt" with a default value of 0x150, but can be changed by the user as needed. If no file is provided the code will default to 0x150.

Also contains the framework for calculating opening load but it's all commented out right now. Just mentioning this for reference if we need to revert back to this version for some reason.

Apr 26, 2021

Linear regression compliance

controller.bas now uses linear regression to calculate compliance.
R squared is present but commented out (goes > 1.000 so that's WIP)
GUI has additional K controller features: plot of current vs. target K, options to run P-ctrl, K-ctrl with C active on Kmax or DK
Does NOT include closure calculations yet.

Apr 13, 2021

K control, 1st draft of core postprocessor

Edits to GUI to introduce K control to "Controls 2" tab.
Dyn. calibration compliance correction moved to ADWin.
Factor "fi" applied real time, tracked in Data[] array.
Correction to K coefficients.
Default waveform starts at mid cycle
First draft of XML file parser / processor;
C(T) compliance and SEN PD OK - checked against FTA software.

Mar 2, 2021

Fixed time based Stop command

Additionally, the waveform starts at the first instance of mid load.

Mar 1, 2021

Set Fi to 0 and re-wrote stop function

Changed the stop request for time to run to 1/freq PLUS the remainder of whatever cycle it's currently running. Best I can do since issuing a command to change control mid-stream doesn't do anything.

Also Fi defaults to 0, not 1.

Feb 27, 2021

Attempt at Compliance and Request Stop Fix

Sends same information to plot and CSV for compliance, and applies remaining_cycles / remaining_time criteria to stop_request.

Feb 22, 2021

Fixed Compliance, improved load / disp

Code now sends B and W to the device upon edits to the "Specimen" tab; this corrects the compliance calculations.

Improved load response from earlier version; gets to within 0.03% for a 100# load

Feb 20, 2021

Fix to Return, Stop, and Compliance Plot

The compliance plot is fixed.
Edits have been made to address the "dump load" and "stop test" issues, but they have to be verified by FL.

Feb 17, 2021

FIX: live monitoring for dynamic calibration

Earlier version updated the voltage even when dynamic calibration was shut off; this is fixed. Also loads and displacements properly update when dynamic cal is checked / unchecked.

Purge data now works.

Live plots now properly display loads regardless of whether dyn cal is on or off.

Feb 15, 2021

Constant live update of Dynamic Calibration

Now has a constant live update of the dynamic calibration percentage, even if it's not applied. The voltage, displacement, and load will update if the box is checked.

Also addresses the R / compliance min ratio issue; this is now resolved within the compliance offset calculator.

Feb 11, 2021

same as last update but with no "pops"

The prior update did not address the 'pop' resulting from WaitAction() within the CycleAction() subroutine.

This was removed and a smaller set of commands was put into the compliance calculator that does the same thing.

Feb 10, 2021

Dynamic Calibration (1st pass)

Adds dynamic calibration factor to loads with new GUI.
Stress ratio fix for compliance (will auto-correct compliance min ratio to R if R is larger)
Default waveform size set to 100.
Auto updates queue frequency to match Hardware settings page

Feb 3, 2021

Fixed live update for p, d

Live load and displacement updates only activated once every (freq) cycles, thus significantly lowering processor load.

Also updates the frequency spinbox based on what's in the queue (bonus!)

Jan 27, 2021

Live update for state_cycle; not dwell yet

Gives a 0.5 second update for load / displacement / voltage during cycling. Only useful info here is max and min since they're averaged over the entire waveform. Average is too so it's just the midopint betweeen max and min.

Jan 20, 2021

Fixed p-d display, limited dwell compliance calc

Properly displays load vs displacement, adjusting for the addition of time as a recordable parameter.
Also calculates compliance for SHORT dwell cycles. (still WIP for general use)

Jan 6, 2021

Voltage readout and time column

Readout modded to include live voltage plus voltages associated with set Pmax and Pmin.
DATA[ ] array changed to include time for static load tests (now has 5 data entries versus original 4 for cycle tests)

Dec 31, 2020

Compliance calculations for arbitrary waveforms

Can now calculate compliance for any given waveform (pre-packaged, user defined, or imported from file).
User has the option to selet calculation for loading, unloading, or both; also, the max and min load levels based on percentage (default 99% and 70%)
Tolerance for max points needs to be checked against actual test specimen results; may change accordingly.

Nov 25, 2020

Proper live load and plot readouts

Fixes the adc polatiry swap issue in the live load output subroutine that gave results of the incorrect sign.

New executable provided.

Nov 14, 2020

Same source, updated executable

The last push did NOT have the executable compiled to reflect the changes from Bryan's last push; this one does. Also updated FLcgs.spec to my directory path. Everything else (*.bas, and *.py) should be the same.

Nov 6, 2020

Mike's first commit (minor upgrades)

Calculates compliance offset index for triangle, sawtooth, sine (non-dwell)
Adds re-zero load function to GUI (under actionReturn in controller.bas)
Adds plotter and CSV output for load and displacement
Adds real-time update of loads / displacements for wait / static / re-zero controller states
Adds placeholder for Data_180 / 181 purge (not yet functional)