TSO Pipeline Proposed Enhancements

The JWST (STScI) Pipeline, and in particular, the steps used by its TSO-mode are in constant development. In this page, we will keep track of the proposed enhancements to the pipeline by the TSO WG. Status of each of the proposed enhancements can be:

IDEA: the step is currently an idea; might or might not have an associated Jira ticket.
PROPOSED: the enhancement has been officially proposed to the CalWebb WG. Discussion ongoing as to whether move it to further study or not.
ONGOING: step currently under development by the Pipeline Development Team.
DONE: step has been implemented in the JWST pipeline.

Priorities are defined for the TSO WG:

1: Critical — will produce unusable/misleading results for TSOs.
2: High — will produce sub-optimal results for TSOs.
3: Low — will produce slightly offset results for TSOS.

Proposed enhancement	Brief description	TODOs	Relevant jira ticket (s)	Meeting links where this was discussed	Status	Priority	Update date
TSO3 Outlier Detection	Currently, the JWST pipeline does not properly use TSO information to reject outliers (e.g., jump step or current tso3 outlier rejection — see Jira tickets). Proposed to implement HST/STIS algorithm developed by Nikolay Nikolov on JWST TSO pipeline.	▪ Algorithm currently tried only on NIRCam data. Need to try it on NIRISS/SOSS, MIRI/LRS and NIRSpec/BOTs.	Problems with current TSO3 step: JP-1285 - Jira project doesn't exist or you don't have permission to view it. JP-1654 - Jira project doesn't exist or you don't have permission to view it. JP-1647 - Jira project doesn't exist or you don't have permission to view it.	CalWebb WG: 2021-05-04 Meeting Notes; 2021-01-05 Meeting Notes. TSO WG: 2021-02-24 TSO WG Meeting notes.	PROPOSED	3	December 2, 2021
Jump detection using TSO information	Similar to the above, but do it at the group-level; detect jumps using TSO information. This is an idea proposed by Michael Regan. Pros: you don't rely on the reference files.	▪ Need to do simulations to test algorithm.			IDEA	2	December 2, 2021

Correct jitter spectral movement prior to photom step	If jitter is an issue during JWST observations, then the photom step will incorrectly apply photometric flux standarization to the wrong wavelengths, introducing time-dependant systematics. TSO WG reps think turning off the photom step would be ideal, but not generic solution. Correcting those jitters prior to the photom step would be a solution.	▪ Need to write cross-correlation function to correct spectra. ▪ Apply it to spectra; and see if this can correct photom step jitter issues.	JP-2082 - Jira project doesn't exist or you don't have permission to view it.	CalWebb WG: 2021-06-01 Meeting Notes. TSO WG: 2021-06-02 TSO WG Meeting notes.	IDEA	2	December 2, 2021
Spectral tracing	Pipeline needs to perform spectral tracing of TSOs in order to (a) record trace movements (useful as external parameters to decorrelate lightcurves) and (b) extract the same portion of the spectrum at each time.	▪ Define tracing algorithms for each instrument. ▪ Test those. ▪ Implement on the pipeline.			IDEA	2	December 15, 2021
Photometric Centroiding	Same as spectral tracing, but for photometry.					2	
Pre-amp reset correction ("reference pixels for subarrays without reference pixels")	▪ Pipeline needs to take care of detector effects not taken care of by, e.g., superbias in the case where there are no reference pixels. ▪ Everett Schlawin proposes a "fast-by-fast", "slow-by-slow" substraction; give pipeline pixels that it can use on each mode (= "unilluminated pixels"). ▪ Algorithms like these are useful not only for subarrays with reference pixels: using noniluminated pixels is even better than "just" using reference pixels. ▪ Important for both spectroscopy and photometry	▪ Various codes to do this already at hand (Everett Schlawin for NIRCam, Nestor Espinoza for NIRISS/SOSS Unknown User (birkmann) for NIRspec). Significant improvement when doing this for NIRISS, NIRSpec and NIRCam. ▪ No tests for MIRI on this yet.			IDEA	1	January 3, 2022
Background substraction	▪ Background substraction is embedded on extract1d. But this is unintuitive. ▪ Need to allow for 2D background profile (e.g., we know NIRISS/SOSS has this!). ▪ This is particularly troublesome for MIRI/LRS.				IDEA	2	January 3, 2022
Optimal extraction	▪ Pipeline currently does box-extractions. Should perform extraction by weighting pixels by their SNRs/noise levels. ▪ Optimal extraction could include likelihood information on 1/f noise, for instance.	▪ Code for uncorrelated gaussian likelihoods is known (Nestor Espinoza has a C-implementation of this), but this assumes spectra go in the cross-dispersion direction (a-la-Marsh, 1989). ▪ Likelihood for 1/f noise is known.			IDEA	2	January 3, 2022

Pixel timing	▪ Due to the sequential nature of the pixel readout modes, every pixel does not have the same time-stamp. The pipeline should either report this, or have an utility function that corrects for this.	▪ Timing of the pixels is known, but uncertainties are not.	Discussion on this: JP-2330 - Jira project doesn't exist or you don't have permission to view it.		IDEA	1	December 15, 2021
Time-stamp accuracy	▪ Pipeline reports times in BJD — but are uncertainties associated with that? What is the precision on this? Has this been tested? (spacecraft position is known accurately probably, but this has a limit?) ▪ Important to note is that there are some Cycle 1 programs aiming to perform in-exposure timing measurements, but not inter-exposure accuracy measurements.				IDEA	1	January 3, 2022
Integration-level aperture positions	▪ Both for photometry and spectroscopy, a TSO aperture position is needed to be defined at each integration, because jitter might make the PSF /spectrum to change in position as a function of time (e.g., due to pointing errors, drifts, etc.). ▪ This can be "hacked" currently for spectrocopy, but doesn't exist for TSO aperture photometry (and should!). Can still be "hacked" for photometry though.				IDEA	3	January 3, 2022