Nancy Grace Roman Space Telescope (Roman)
Technical Memo

Title: Identification of an Algorithm to Simulate Image Persistence of the Roman Wide Field Instrument	Doc #: Roman-STScI-000480 Date: January 31, 2023 Rev: -
Authors: Javier Sanchez, John F Wu, Tyler Desjardins, Andrea Bellini, Stefano Casertano, Eddie Schlafly Phone: 410-338-6537	Release Date: March 14, 2023

Abstract
The main goal of this work is to identify an algorithm suitable to simulate image persistence in Roman Wide Field instrument detectors, as required by image simulation tools that will be made available to the Roman community. In order to do so, we focus on the algorithm described in Lin et al. (2022). We compare the predicted persistence levels by this algorithm with early Roman data from the Detector Characterization Laboratory (DCL). We find that the model is compatible with the measured excess dark current for fluence levels (i.e., number of electrons per pixel) of 110 ke− or below, given our uncertainties, which is sufficient for the current stage of the project. We however find a significant underestimation of the amplitude of the persistence level for very-high fluence levels ($\sim$ 300 ke−). We propose a series of modifications to allow extra usage-flexibility, and improve the overall performance of the algorithm. The proposed modifications will allow us to increase the accuracy of the model without major changes in the code. We conclude our work including a Python snippet with our proposed algorithm.

1 Introduction

Image persistence, which we will refer to simply as persistence throughout this work, consists on an elevated dark current due to prior illumination, and has been known to limit data quality in HgCdTe infrared detectors (Smith et al. 2008a,b; Long et al. 2010; Tulloch 2018). The Roman Wide Field Instrument (WFI) will use 18 4k × 4k newly developed HgCdTe H4RG-10 detectors. The level of persistence on these new detectors is smaller than that of detectors from previous generations (Mosby et al. 2020). However, an accurate characterization of persistence is required in order to meet the ambitious scientific goals of the Roman mission, as persistence can bias the determination of positions, fluxes, and shapes of astronomical sources.

A recent study analyzed the impact of persistence in weak-lensing measurements using image simulations (Lin et al. 2022), and introduced an implementation of persistence using `GalSim` (Rowe et al. 2015) based on results from Mosby et al. (2020). The algorithm uses the Fermi model introduced at Long et al. (2015a), which was found to provide a good description of persistence for WFC3/IR images. Similar studies will be needed by the community in order to assess the impact of persistence on different probes, and to potentially devise mitigation strategies for this effect. This motivates the development of modeling techniques and algorithms to include persistence in image simulation tools.

A noteworthy effect related to persistence is the so-called burn-in (Mosby et al. 2020) or "inverse persistence" (Regan et al. 2012). This effect causes pixels exposed to large increase in the illumination levels to accumulate charge more slowly than expected, due to the presence of electron traps. For this study we will not address this effect and focus solely on persistence.

In this work we aim to identify an algorithm that enables the user to include persistence in simulated images generated by image simulation tools that will be available to the Roman community, as required by SOC-1036 and RSUBREQ-837.[1]

This document is structured as follows: In Section 2 we briefly introduce persistence and how it is typically modeled. In Section 3 we compare the model with detector characterization data and present an example implementation of an algorithm to include persistence in images using Python.

2 Image persistence

The concrete physical processes that give rise to persistence are not fully understood. However, the most widely accepted mechanism that gives rise to persistence is the so-called trap model (Smith et al. 2008a).

In this model, depletion region traps are the source of persistence. These traps are filled with charge as the depletion region shrinks due to illumination. Once the depletion region is reset to its original width the trapped charges are emitted.[2]

[1] Progress and details can be tracked in https://jira.stsci.edu/browse/RTB-344.

[2] Note that this model can also explain the burn-in effect, as the trapped charges make new charge to accumulate slower in subsequent illuminated exposures.

Persistence is typically negligible for fluence levels less than half of full well (Long et al. 2015a). However, for pixels close to saturation level it can exceed the dark current level. Persistence not only depends on the fluence level of previous exposures, but also on the total illumination history of a pixel, on the exposure time, and on how "quick" or slow the charge has been accumulating on the detector (Long et al. 2015a).

Based on this trap model and semiconductor physics, recent works try to develop phenomenological models for the persistence level as a function of time (Tulloch et al. 2019; Le Goff et al. 2022). For astronomical data, persistence is typically characterized by the empirical model presented in Long et al. (2015a), which has been observed to be a good approximation for different detectors (Long et al. 2015a,b; Tulloch 2018), including the laboratory data from the Roman detectors (Mosby et al. 2020). According to this model (time-dependent Fermi model) the persistence level, $P(x,t)$, can be written as:

$$P(x,t) = A \left(\frac{1}{1+\exp\left(\frac{x-x_0}{-\delta x}\right)} \right) \left(\frac{x}{x_0} \right)^{\alpha} \left(\frac{t}{1000} \right)^{-\gamma}, \tag{1}$$

where x is the fluence level of the image responsible for the persistence signal, t is the time passed (in seconds) since the illumination causing persistence ended and $A, x_0, \delta x, \gamma$ and α are free parameters. Note that, following equation 1, $P(x,t)$ is expressed as a flux at a given time t and is given in the same units as A. In the case of Mosby et al. (2020); Lin et al. (2022) the authors use the following values as typical: $A = 0.017$ e$^-$, $x_0 = 6.0 \times 10^4$ e$^-$, $\delta x = 5.0 \times 10^4$ e$^-$, $\alpha = 0.045$, $\gamma = 1$.

3 Simulating Persistence

For this work, we adapt the model from equation 1. There are, however, some open questions that we want to explore in this work:

1. Do the parameters in Mosby et al. (2020); Lin et al. (2022) provide a good description of the detectors?
2. How many previous exposures should be propagated/kept in memory to accurately describe persistence?
3. What should be done in the case of "overlapping" persistence events?
4. Should we consider spatial variations in the parameters of the model?

We try to address these questions in the following subsections.

3.1 Comparison with DCL data

For this work, we use the raw acceptance test data for sensors 21441, 21996, 22066, and 22067. In particular, we use the `pers` files. These files are described in detail in their respective acceptance tests reports. These files are organized as sets of consecutive exposures. The first file of the group contains data where a luminous source was illuminating the sensor for

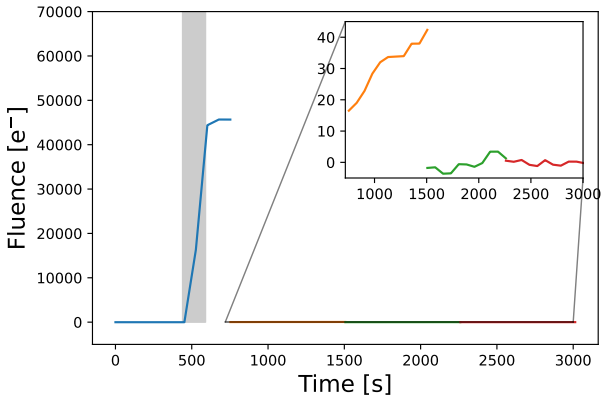

Figure 1: Clipped mean fluence level as a function of time for the persistence test data from sensor 21441. Each colored line represents a different exposure of the `11p3mv` set. The first exposure (blue line) is the only one where the sensor is illuminated. Then the second (orange), third (green), and fourth (red) subsequent exposures are not illuminated. The illumination lasts for $\sim$ 100s and it's shown as the shadowed region. The inset plot shows a zoom in to see the accumulated excess dark current in the subsequent exposures.

File group	Fluence [e^-/pixel]
`11p3mv`	50,000
`26p8mv`	110,000
`40p0mv`	170,000

Table 1: Summary table showing the relation between filename convention and fluence level for the data used in this work. Data at additional fluence levels are available, and were examined during the course of the analysis but are not discussed on this work.

100s [3]. The maximum fluence level is different for each of the groups, and is indicated in the file name as a voltage level. Table 1 summarizes the different files studied in this work, relating filename to maximum fluence level for a given group of files. Figure 1 illustrates one of of these file groups.

In order to characterize the persistence effect on the Roman detectors using these data, we need to first subtract the dark contribution. Thus, we construct a master dark file as the mean value of 18 dark exposures with the same exposure time and number of reads (10 minutes, 11 reads) as the persistence test data.

We subtract the dark contribution and compute the excess dark current as a function of time. We check the model for the $\sim 110,000$ e^- fluence level exposures, which correspond to the `26p8mv` files. The measured excess dark current is shown in Figure 2. In this figure we compare the excess dark current for the individual sensors (left panel) and their averages (right panel). We see that the model in equation 1 (represented by the black and gray lines) describes qualitatively well the data. Our best-fit model (gray line) is statistically compatible with those found by Mosby et al. (2020). Using only the observations in the `26p8mv` files the model parameters seem to be mostly unconstrained, except for γ. This is largely due to the

[3]The illumination for the first exposure of the group happens between $t = 440$ and $t = 540$s.

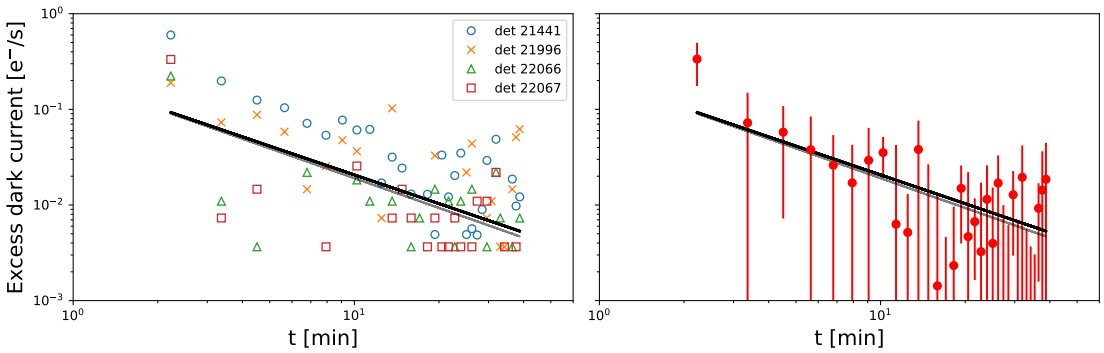

Figure 2: Left panel: Excess dark current level as a function of time for detectors 21441 (blue), 21996 (orange), 22066 (green), and 22067 (red). The model in equation 1 provides a good qualitative description of the data with the parameters in Lin et al. (2022) (black line) and our best-fit (gray line). Right panel: Average of the 4 detectors excess dark current as a function of time. The error bars are calculated as the standard deviation of the 4 detectors excess dark current as a function of time.

degree of degeneracy between $A, x_0, \delta x, \alpha$.

We now assess the performance of the model using these parameters for different fluence levels using the `11p3mv` ($\sim$ 50 ke$^-$) and `40p0mv` ($\sim$ 170 ke$^-$) data, focusing on the detector 21441. Our model seems to underpredict the dark current for the `40p0mv` data, but is statistically compatible with the data for the 50 ke$^-$ exposures as shown in Figure 3, when using the dispersion between different sensors as uncertainties. As we mentioned before, given that the parameters are largely unconstrained, simultaneously fitting data that are illuminated at different fluence levels (x) could, in principle, help break the degeneracies between the different parameters, and improve the fidelity of the model for the full dataset. However, even when fitting all of these data together, the best fit parameters still underpredict the persistence level for the `40p0mv` data, indicating a dependence of these parameters with the fluence level that might not fully captured by the model as it was shown in Long et al. (2015a)[4]. More recent studies, such as Gennaro (2018); Gennaro et al. (2019), also find this dependency of the parameters with fluence level in WFC3/IR data for a similar range of time after illumination as our study. Exploring a different model or set of parameters that can better describe the persistence level for different fluence levels is left for future work.

3.2 Time dependence and typical decay time

In this subsection we focus on the question of how many exposures need to be tracked in order to have an accurate description of persistence. This question is intrinsically related to the decay time of the persistence effect. As shown in Figures 2 and 3, the time dependence (i.e., the slope of the log-log plot) seems to be well described by our power-law model in equation 1. This means that, in most practical situations, the persistence level after 1 or 2 exposures is essentially negligible. However, we also see that for very high fluence levels,

[4] Long et al. (2015a) also find that these parameters depend on the exposure time.

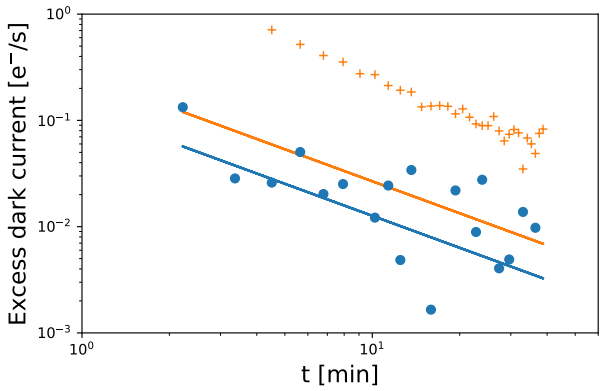

Figure 3: Excess dark current for the `11p3mv` (blue points) and `40p0mv` (orange crosses) data. The models using the parameters from Lin et al. (2022) are shown as the solid lines for the `11p3mv` (blue line) and `40p0mv` fluence levels. This choice of parameters seems to underestimate the excess dark current for the `40p0mv` data.

such as those from the `40p0mv` data, there is a persistence signal of > 0.1 e$^-$/s even one hour after illumination. As a consequence, it is difficult to set a maximum number of exposures to propagate through. Ideally, we want to keep track of as many exposures as possible, but this increases the memory usage rapidly if the full images are propagated through, as the WFI images are quite large. The current implementation in `GalSim v2.4` (Rowe et al. 2015) keeps in memory the full images, and lets the user select the maximum number of images propagated through. The persistence level of each prior image is computed at the time of the simulated image of interest. Most astronomical images however, will have only a small fraction of pixels with fluence levels that are high-enough to cause measurable persistence. We will explore how to take advantage of this for our suggested implementation if the following sections.

3.3 Overlapping persistence events

In this subsection we want to discuss the case when a pixel that is already affected by persistence is illuminated at a high fluence level. In Long et al. (2015a) the authors opted for only considering the persistence caused by one exposure (the one with the maximum persistence). In contrast, Lin et al. (2022) allows the user to select how many exposures to keep track of, and add together all of these contributions to the final image. Adding the persistence levels seems to reproduce well the behavior observed in different detectors (Tulloch et al. 2019). In our implementation, we take advantage that, in most situations, the majority of pixels will not be affected by persistence, and it is more efficient to track pixel number, fluence, and time since exposure of the pixels that are affected. So, we suggest to an implementation where we store three different lists or `numpy.ndarray` objects: one containing the pixel indices (16-bit integer), one with the fluence level (32-bit floating point), and one with the time since the image causing persistence (32-bit floating point). This approach results in a reduced memory usage with respect to the case of keeping $N_{\rm pers}$ previous images, as long as less than 40% of

the pixels of each of the $N_{\rm pers}$ previous images considered are stored in memory[5]. Further performance improvements can be achieved if pixels below a certain threshold (for example, below 10% of the read-noise level, but this threshold can be user-selected), are no longer propagated through.

3.4 Spatial variations

In previous works, the variation of persistence as a function of position was found to be non-negligible for different quadrants of the detector for WFC3/IR (Long et al. 2015b). Additionally, the authors in Mosby et al. (2020) find spatial variations in the WFI detectors, and that these variations depend on the illumination level. We focus on cross-checking these results by analyzing images with high levels of persistence (`73p0mv`) and examine the relative variation of the persistence level as a function of position, which we characterize as δ:

$$\delta(X, Y) = x(X, Y)/\langle x \rangle - 1, \tag{2}$$

where x is the persistence level and $\langle x \rangle$ is the average persistence across the image. These variations are illustrated in Figure 4. Here we see that these variations are significant. However, a model for these variations is out of scope for this work, and is left as for future investigations. Despite not providing a model for the spatial variations on the persistence, at implementation level it is simple to allow for these variations by simply making the coefficients in equation 1 be arrays with the same size as the original image (or of the list of pixels that are affected by persistence).

3.5 Algorithm overview

Given the discussion in the previous sections, we design our algorithm as follows:

- We use a time-dependent Fermi model (Long et al. 2015a) as described in equation 1.
- The user can provide a list of pixel indices, fluence levels, and time at which the exposure ended. This list is propagated through to the current exposure.
- We add the persistence level from overlapping persistence events.
- We allow for spatial variations of persistence by allowing the user to pass an image (of the same size as the original image) with different choices for the parameters of the model in equation 1.

Potential improvements to the algorithm described here include changes in the model to better describe the dependence of persistence with fluence level and exposure time. Additionally, further studies to characterize the spatial variation of persistence will prove valuable to improve the level of realism of simulated images affected by persistence.

[5] If the image causing persistence in the original algorithm is 16-bits instead of 32-bit, there is improvement in performance as long as less than 1/4 of the pixels are propagated through.

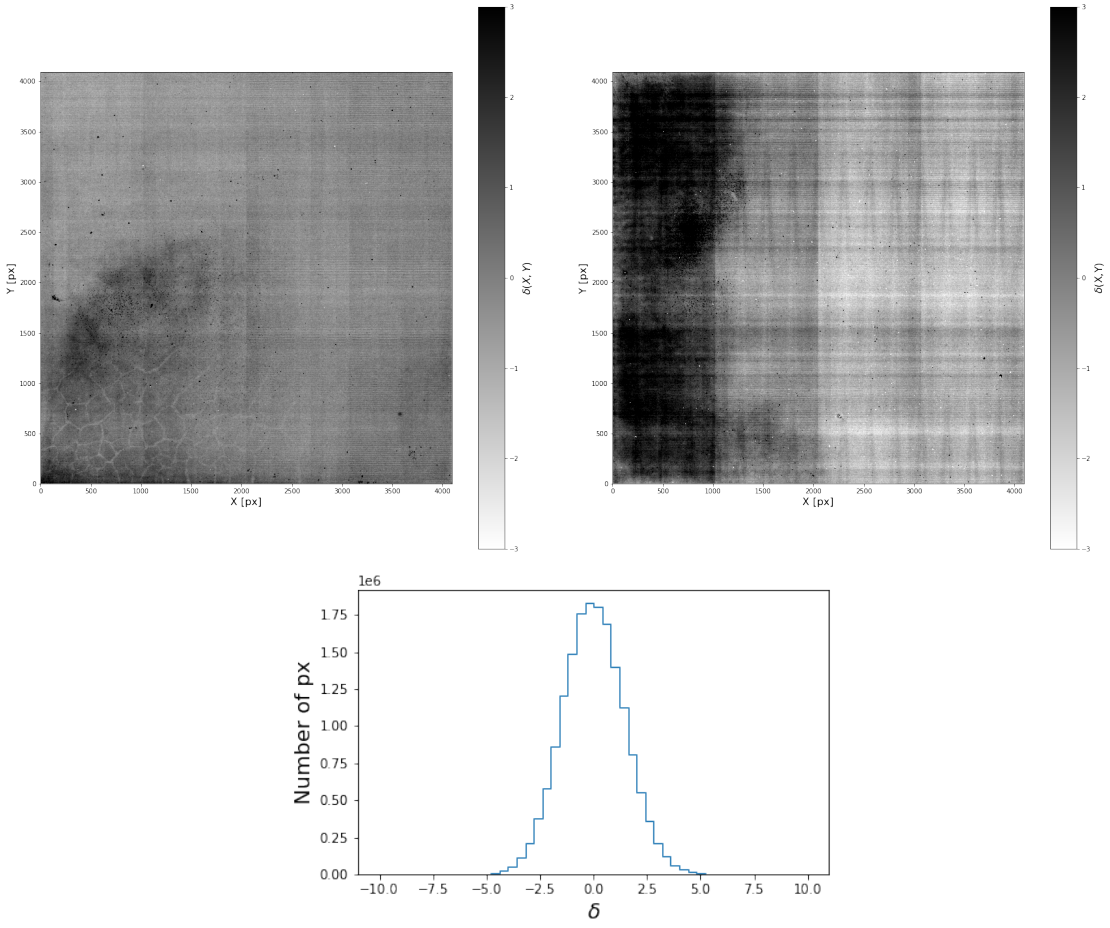

Figure 4: Relative spatial variation of excess dark current for detectors 21441 (top left panel) and 22067 (top right panel). We also include the 1D histogram for detector 21667 (bottom panel).

3.6 An example implementation in Python

```python
import numpy as np

def fermi(x, t, A, x0, dx, a, r):
    """
    The fermi model for persistence: A* (x/x0)**a * (t/1000.)**(-r) / (exp( -(x-x0)/dx ) +1. )
    For influence level below the half well, the persistence is linear in x.

    Parameters:
    -----------
        x:      Fluence level (in e-).
        t:      Time since exposure (in s).
        A:      Amplitude parameter of persistence (in e-)
        x0:     Pivot fluence (in e-).
        dx:     dx parameter (in e-).
        a:      Power of fluence level (no units).
        r:      Power of time decay (no units)
    Returns:
    --------
        The persistence signal of the input exposure x (in e-).
    """

    ps    = A* (    x    /x0)**a * (t/1000.)**(-r)/(np.exp( -(x-x0)/dx) +1.)
    return ps

def add_persistence(img, x, t, px, A_img, x0_img, dx_img, a_img, r_img):
    """
    Function that adds persistence to a given image

    Parameters:
    -----------

        img: np.ndarray, shape: (npix_x, npix_y)
        Image to which we add persistence (in e-).
        x: np.ndarray, shape: (npers,)
        Fluence level of pixels to be propagated through (in e-)
        t: np.ndarray, shape: (npers,)
        Time since pixel was illuminated at fluence x (in s).
        px: np.ndarray, shape: (npers,)
        Pixel index (in 1d) of the propagated pixel that was illuminated at fluence x, t seconds ago.
        A_img: float or np.ndarray with shape (npix_x, npix_y)
        Image with A coefficients. If a scalar is passed, the coefficient is considered constant across the image,
        i.e., no spatial variation of persistence is considered.
        x0_img: float or np.ndarray with shape (npix_x, npix_y)
        Image with x0 coefficients. If a scalar is passed, the coefficient is considered constant across the image.
        dx_img: float or np.ndarray with shape (npix_x, npix_y)
```

```
        Image with dx coefficients. If a scalar is passed, the coefficient is considered constant across the image.
        a_img: float or np.ndarray with shape (npix_x, npix_y)
        Image with a parameter. If a scalar is passed, the coefficient is considered constant across the image.
        r_img: float or np.ndarray with shape (npix_x, npix_y)
        Image with r parameter. If a scalar is passed, the coefficient is considered constant across the image.

    Returns:
    --------

        img: np.ndarray, shape: (npix_x, npix_y)
        Image including the persistence level induced by previous exposures. In the case of simulating reads the result needs to be multiplied by the time
        ellapsed between reads.
    """
    nx, ny = img.shape
    img = img.reshape(-1)
    if np.isscalar(A_img):
        # We use add.at so we accumulate repeated entries, if you use += only the last value is used (it overwrites)
        np.add.at(img, px, fermi(x, t, A_img, x0_img, dx_img, a_img, r_img))
    else:
        np.add.at(img, px, fermi(x, t, A_img.flatten()[px], x0_img.flatten()[px], dx_img.flatten()[px],
                  a_img.flatten()[px], r_img.flatten()[px]))
    return img.reshape((nx, ny))
```

References

Gennaro, M. 2018, WFC3 Short-term IR Persistence, HST Proposal. Cycle 26, ID. #15581

Gennaro, M., Baggett, S., & Bajaj, V. 2019, A Characterization of Persistence at Short Times in the WFC3/IR Detector. II, Instrument Science Report WFC3 2019-2, 11 pages

Le Goff, T., Pichon, T., Baier, N., Gravrand, O., & Boulade, O. 2022, Journal of Electronic Materials, 51, 5586, doi: 10.1007/s11664-022-09854-7

Lin, C.-H., Mandelbaum, R., Troxel, M. A., Hirata, C. M., & Jarvis, M. 2022, MNRAS, 512, 3312, doi: 10.1093/mnras/stac512

Long, K. S., Baggett, S., Deustua, S., & Riess, A. 2010, WFC3/IR Persistence as Measured in Cycle 17 using Tungsten Lamp Exposures, WFC3 Instrument Science Report 2010-17, 12 pages

Long, K. S., Baggett, S. M., & MacKenty, J. W. 2015a, Persistence in the WFC3 IR Detector: an Improved Model Incorporating the Effects of Exposure Time, WFC3 Instrument Science Report 2015-15, 18 pages

—. 2015b, Persistence in the WFC3 IR Detector: Spatial Variations, WFC3 Instrument Science Report 2015-16, 16 pages

Mosby, G., Rauscher, B. J., Bennett, C., et al. 2020, Journal of Astronomical Telescopes, Instruments, and Systems, 6, 046001, doi: 10.1117/1.JATIS.6.4.046001

Regan, M., Bergeron, E., Lindsay, K., & Anderson, R. 2012, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 8442, Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave, ed. M. C. Clampin, G. G. Fazio, H. A. MacEwen, & J. Oschmann, Jacobus M., 84424W, doi: 10.1117/12.927033

Rowe, B. T. P., Jarvis, M., Mandelbaum, R., et al. 2015, Astronomy and Computing, 10, 121, doi: 10.1016/j.ascom.2015.02.002

Smith, R. M., Zavodny, M., Rahmer, G., & Bonati, M. 2008a, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 7021, High Energy, Optical, and Infrared Detectors for Astronomy III, ed. D. A. Dorn & A. D. Holland, 70210J, doi: 10.1117/12.789372

Smith, R. M., Zavodny, M., Rahmer, G., & Bonati, M. 2008b, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 7021, High Energy, Optical, and Infrared Detectors for Astronomy III, ed. D. A. Dorn & A. D. Holland, 70210K, doi: 10.1117/12.789619

Tulloch, S. 2018, arXiv e-prints, arXiv:1807.05217. https://arxiv.org/abs/1807.05217

Tulloch, S., George, E., & ESO Detector Systems Group. 2019, Journal of Astronomical Telescopes, Instruments, and Systems, 5, 036004, doi: 10.1117/1.JATIS.5.3.036004