Exposure Time Calculator for the Roman Coronagraph Instrument

Corey Spohn[1] – SIOSlab & Sergi Hildebrandt Rafels[2] – Coronagraph Project Science Team
(1: Cornell University; 2: Jet Propulsion Laboratory, California Institute of Technology)

With support from: Vanessa Bailey, John Debes, Dean Keithly, Brian Kern, Bijan Nemati, Dmitry Savransky, Leah Sheldon

Roman Coronagraph Info Sessions
Day 2. 10/28/2021

CGI Perf: overview

- Excel spreadsheet[1,2,3] to calculate exposure time to reach SNR accounting for
  - Contrast stability
  - Coronagraph performance
  - Throughput
  - Detector QE
  - Albedo
  - Stray light
  - Exozodi
  - Observing scenario

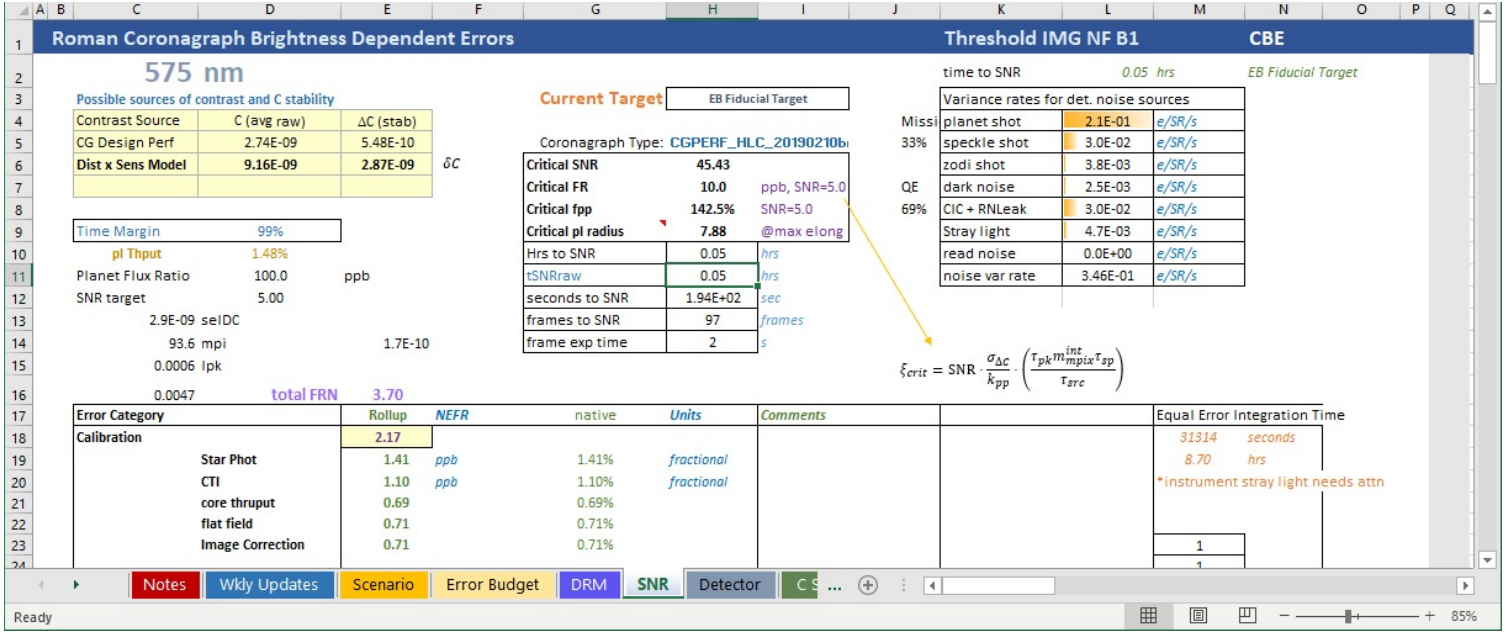
$$t = \frac{\mathrm{SNR}^2 C_b}{C_p^2 - \mathrm{SNR}^2 C_{sp}^2}$$

- t – Exposure time
- SNR –Signal to noise ratio
- C_b – Background electron noise count rate
- C_p – Planet signal electron count rate
- C_{sp} – Residual speckle count rate

[1] Bijan Nemati (2020) doi: 10.1117/12.2575983; [2] Bijan Nemati, H. Philip Stahl, Mark T. Stahl, Garreth J. J. Ruane, Leah J. Sheldon (2020) doi:10.1117/1.JATIS.6.3.039002 [3] Roman Coronagraph Instrument private repository

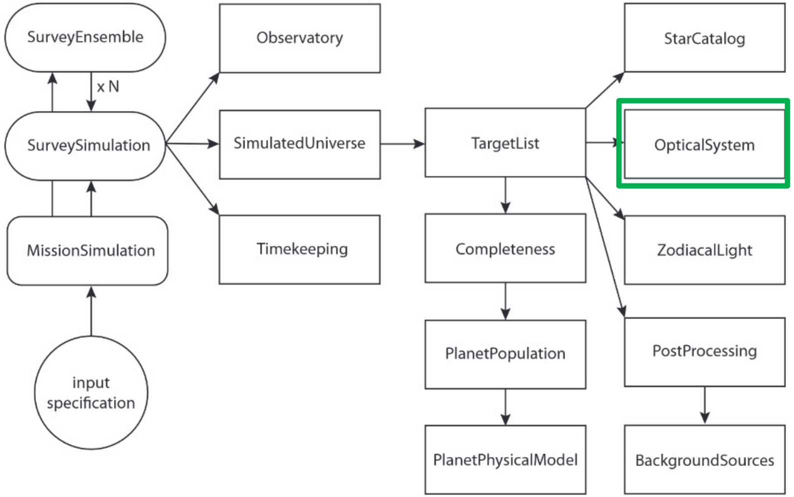
CGI Perf: overview

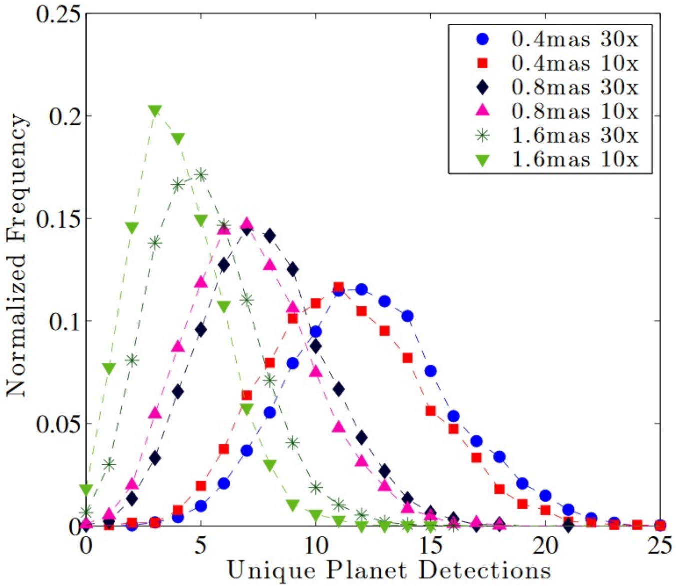
- Great for testing, but not very extensible



EXOSIMS: overview

- End-to-end simulations of space-based exoplanet imaging missions written in Python[1,2,3]
- Estimate detection and characterization yield by simulating hundreds of missions
- Extensible, modular, and open source





[1] Dmitry Savransky, C. Delacroix, D. Garrett (2017) 2017ascl.soft0610S [2] Dean Keithly, D. Savransky, D. Garrett, C. Delacroix, G. Soto (2020) doi:10.1117/1/JATIS.6.2.027001 [3] https://github.com/dsavransky/EXOSIMS

CGI Perf in EXOSIMS

- Goal: Calculate exposure times quickly for a many targets and many observing scenarios
- OpticalSystem module written that can calculate the CGI Perf exposure time calculations
- Works directly from CGI Perf csv inputs
- Methods[1,2,3]:
  - Calculate background, speckle, and planet count rates
  - Calculate integration time
  - Calculate Δmag for an integration time via numerical inversion

[1] Bijan Nemati (2020) doi: 10.1117/12.2575983; [2] Bijan Nemati, H. Philip Stahl, Mark T. Stahl, Garreth J. J. Ruane, Leah J. Sheldon (2020) doi:10.1117/1.JATIS.6.3.039002 [3] Roman Coronagraph Instrument private repository

EXOSIMS input

- A JSON file
  - Telescope and Coronagraph information[1]
  - Specify which EXOSIMS modules to use
  - Hard coded parameters such as planet orbital parameter ranges
  - For each starlight suppression system we load the csv files from the CG_Perf spreadsheet
- For more info see https://exosims.readthedocs.io/en/latest/userparams.html
- To create and use planets in code, use the TargetList module
- EXOSIMS uses the TimeKeeping module to track time

```
  starlightSuppressionSystems : [
   {  name :  HLC-565 ,
      lam : 575,
      BW : 0.10,
      Rs : 0,
      IWA : 0.13049,
      OWA : 0.45,
      Nlensl : 5.0,
      lenslSamp : 2.0,
      occ_trans :  $HOME/Documents/github/WFIRST-CGI-
      core_thruput :  $HOME/Documents/github/WFIRST-C
      core_mean_intensity :  $HOME/Documents/github/W
      core_area :  $HOME/Documents/github/WFIRST-CGI-
      core_contrast :  $HOME/Documents/github/WFIRST-
      core_platescale : 0.30,
          core_stability :  $HOME/Documents/github/WF
          core_stability_setting :  MCBE ,
          CG_Perf :  $HOME/Documents/github/WFIRST-CG
   }
 ],
  observingModes : [
   {  instName :  imager ,
      systName :  HLC-565 ,
      detectionMode : true,
          ContrastScenario :  2019_PDR_Update ,
      SNR : 5,
          tau_pol : 1,
          GCRFlux : 5,
          photons_per_relativistic_event : 250,
          luminescingOpticalArea : 0.7854,
          OpticalThickness : 4,
          luminescingOpticalDistance : 0.1,
          s_baffling : 0.001
   }
 ],
  modules : {
    PlanetPopulation :  KnownRVPlanets ,
    StarCatalog :   ,
    OpticalSystem :  Nemati_2019 ,
    ZodiacalLight :  Stark ,
    BackgroundSources :  GalaxiesFaintStars ,
    PlanetPhysicalModel :  ForecasterMod ,
    Observatory :  WFIRSTObservatoryL2 ,
    TimeKeeping :   ,
    PostProcessing :   ,
    Completeness :   ,
    TargetList :  KnownRVPlanetsTargetList ,
    SimulatedUniverse :  KnownRVPlanetsUniverse ,
    SurveySimulation :   ,
    SurveyEnsemble :
```

[1] https://roman.ipac.caltech.edu/sims/Param_db.html

Roman Coronagraph Exposure Time Calculator

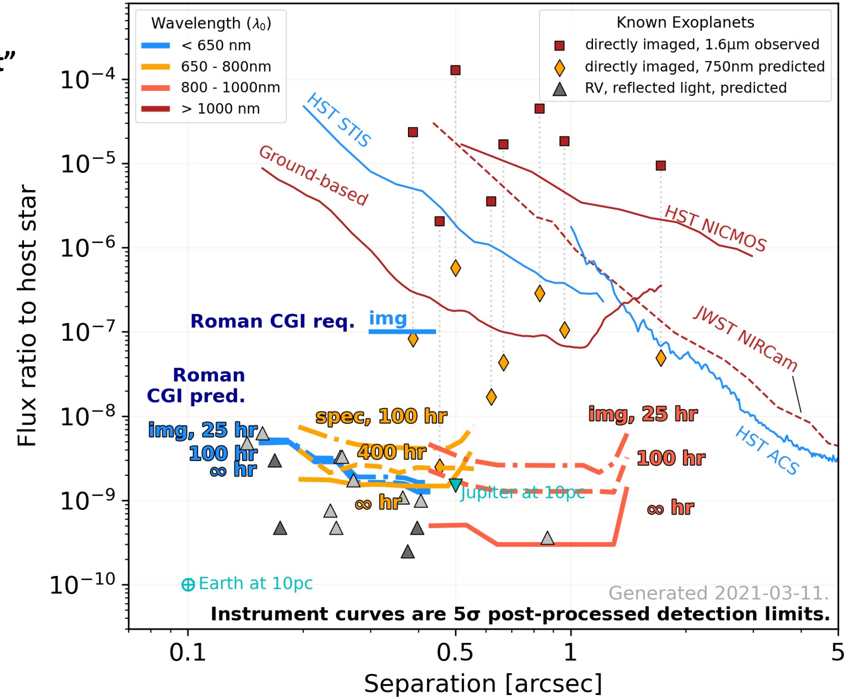
github.com/nasavbailey/DI-flux-ratio-plot/

See “**Observing with the Coronagraph Instrument**” talk on Day 1

Predicted detection limits are strongly speckle-limited at shorter wavelengths

Based on lab demonstrations as inputs to high-fidelity, end-to-end thermal, mechanical, optical models.

Most Model Uncertainty Factors set to ~1



Brian Kern (JPL)
John Krist (JPL)
Bijan Nemati (UA Huntsville)
A.J. Riggs (JPL)
Hanying Zhou (JPL)
Sergi Hildebrandt Rafels (JPL)

Roman Coronagraph Exposure Time Calculator

Allows the user to derive integration times for:

- Three types of **targets**:
  - Self luminous exoplanets
  - Reflected light exoplanets
  - Exodust
- Three different **observing modes**:
  - Narrow Field of View Imager (575 nm, 10% bandwidth)
  - Single Slit, Prism-based Spectroscopy (730 nm, 15% bandwidth, R=50)
  - Wide Field of View Imager (825 nm, 10% bandwidth)
- Two Instrumental and Mission **performance** levels:
  - **Optimistic**: most model uncertainty factors are set to 1
  - **Conservative**: current best estimates of model uncertainty factors (will be updated as ground tests happen)
- Fully **documented**
- Public repository on **GitHub**
- Written in python

https://github.com/hsergi/Roman_Coronagraph_ETC

SHR: acknowledges support by the Turnbull Roman CGI Science Investigation

Roman Coronagraph Exposure Time Calculator

Allows the user to derive integration times for:

- Three types of **targets**:
  - Self luminous exoplanets
  - Reflected light exoplanets
  - Exodust
- Three different **observing modes**:
  - Narrow Field of View Imager (5
  - Single Slit, Prism-based Spectroscopy (730 nm, 15% bandwidth, R=50)
  - Wide Field of View Imager (825 nm, 10% bandwidth)

- Two Instrumental and Mission **performance** levels:
  - **Optimistic**: most model uncertainty factors are set to 1
  - **Conservative**: current best estimates of model uncertainty factors (will be updated as ground tests happen)
- Fully **documented**
- Public repository on **GitHub**
- Written in python

Deep Dive Info Session on November 18, 2021, 9 am PT: Link

https://github.com/hsergi/Roman_Coronagraph_ETC

SHR: acknowledges support by the Turnbull Roman CGI Science Investigation

Questions