EXOSIMS: The Open Source Exoplanet Imaging Mission Simulator

Yield Modeling Tools Workshop 2023

Dmitry Savransky for the EXOSIMS Development Team

Cornell University

Carl Sagan Institute

SIOS Laboratory

Jet Propulsion Laboratory
California Institute of Technology

June 8, 2023

Predicting Exoplanet Yield: Summed Completeness

Expected number of exoplanet detections for n target stars:

$$E[\text{detections}] = \eta \sum_{k=1}^{n} k \sum_{j \in {}_nC_k} \prod_{i \in j} p_i \prod_{i \notin j} (1 - p_i) = \eta \sum_{i=1}^{n} p_i$$

Planet Occurrence Rate

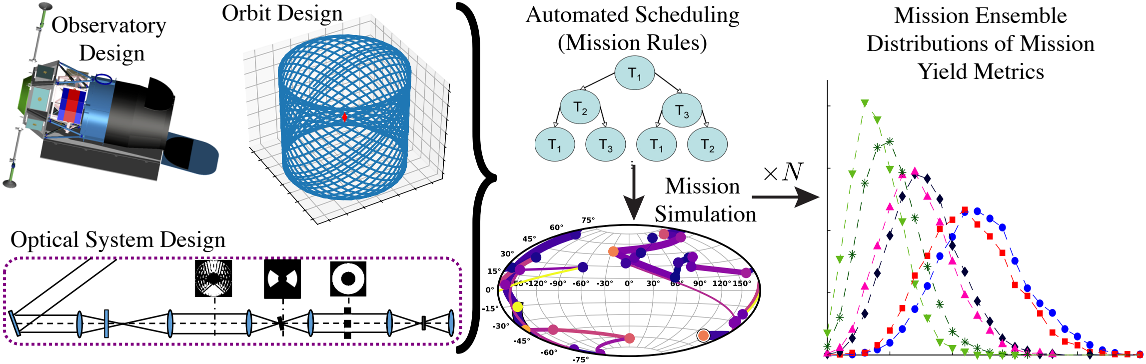
Combinations of $\{i\}_{i=1}^{n}$ Taken k at a Time

Probability of Planet Detection at ith Target

- Pro: (Relatively) Straightforward to compute
- Con: Standard deviation and other metrics are more complicated to get right
- Pro and Con: Can get a result without actually scheduling observations

See: Brown, "Single-visit photometric and obscurational completeness", 2005; Garrett and Savransky, "Analytical Formulation of the Single-visit Completeness Joint Probability Density Function", 2016; Garrett, Savransky, and Macintosh, "A Simple Depth-of-Search Metric for Exoplanet Imaging Surveys", 2017

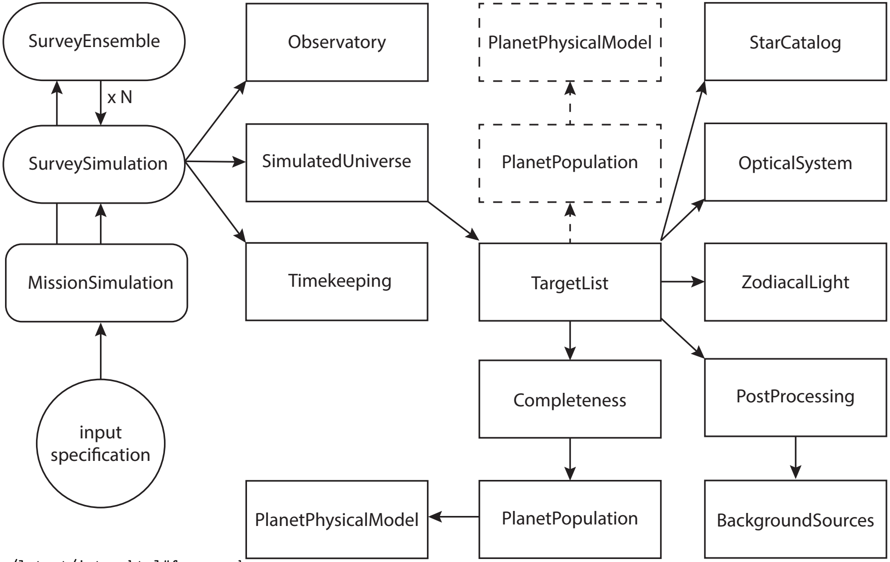
Predicting Exoplanet Yield: Monte Carlo Mission Modeling



- Pro: Can extract effectively *any* metric of performance with errorbars
- Con: Computationally costly
- Pro and Con: Requires a mission schedule

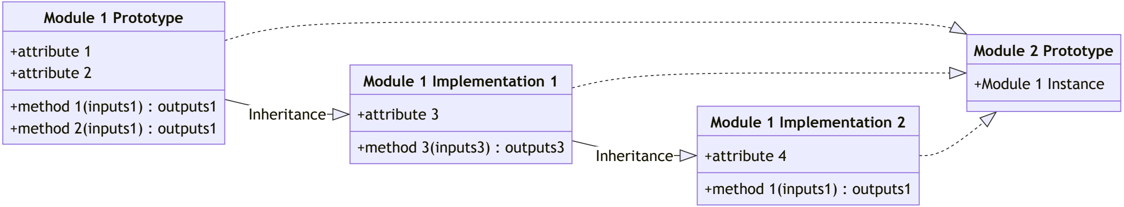
See: Savransky, Kasdin, and Cady, "Analyzing the designs of planet finding missions", 2010; Savransky and Garrett, "WFIRST-AFTA coronagraph science yield modeling with EXOSIMS", 2015

EXOSIMS: A Framework for Monte Carlo Mission Modeling



https://exosims.readthedocs.io/en/latest/intro.html#framework

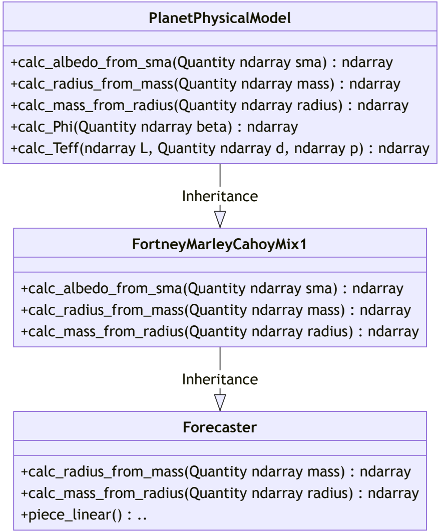
Maximizing Code Re-Use Through Inheritance



- Module Prototypes provide *all* attributes and methods required by all other prototypes and set input/output specification for all required methods
- Module Implementations may add additional attributes/methods and/or overload existing methods (so long as input/output remains unchanged)
- Internally, module objects are referred to only by their module type (e.g. any implementation of `TargetList` is called as `TargetList`)

https://exosims.readthedocs.io/en/latest/exosimsmods.html#module-inheritance-and-initialization

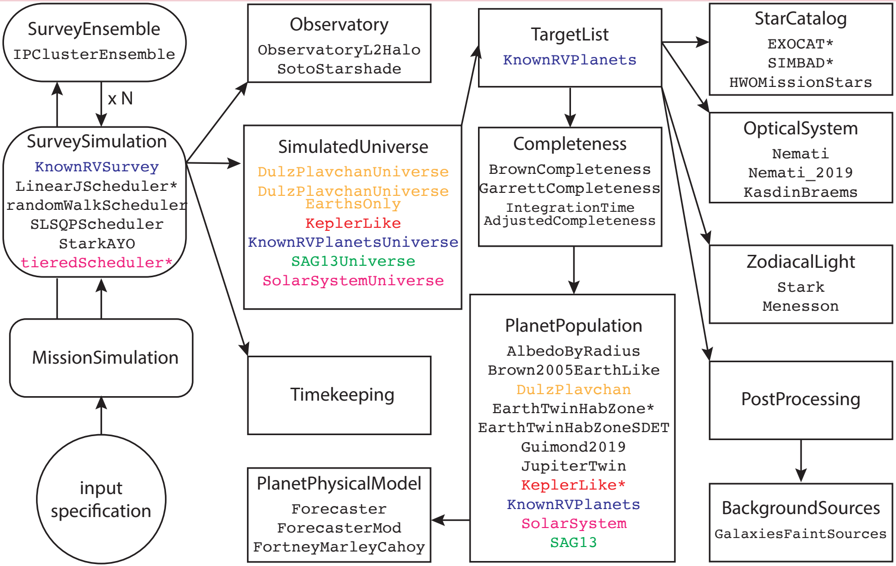
An Inheritance Example



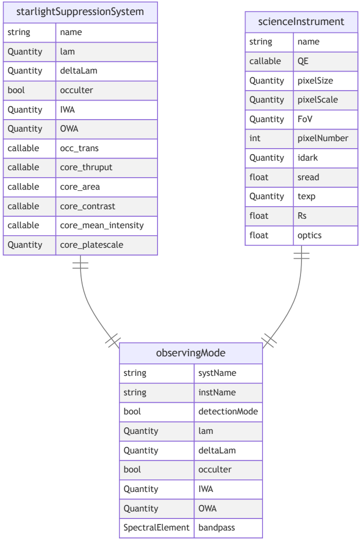
When `Forecaster` is being used as the planet physical model:

- `PlanetPhysicalModel.calc_Phi` calls the Prototype method
- `PlanetPhysicalModel.calc_albedo_from_sma` calls the method from `FortneyMarleyCahoyMix1`
- `PlanetPhysicalModel.calc_mass_from_radius` calls the method from `Forecaster`

(Some) EXOSIMS Implementations

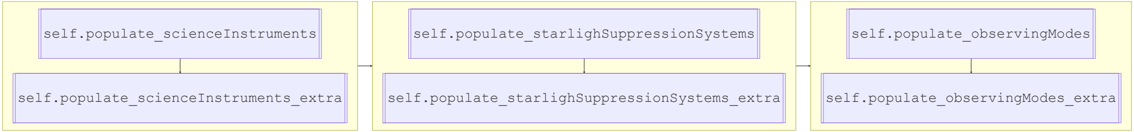


Optical System Encoding



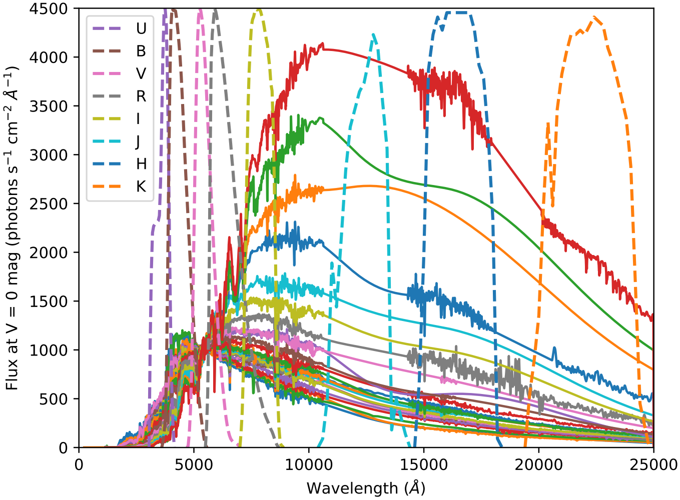
- An optical system is defined as collection of:
  - Science Instruments
  - Starlight Suppression Systems
  - Observing Modes
- Science instruments can be imagers or spectrometers
- Starlight suppression systems can be (internal) coronagraphs or (external) occulters (starshades)
- An observing mode combines a starlight suppression system with a science instrument and can override certain parameters (e.g. wavelength range and IWA/OWA)

Optical System Initialization and Inheritance



- All `*_extra` methods are empty in the prototype optical system
- The prototype populates all science instrument attributes required to describe a conventional detector
- Implementation `Nemati` overload method `populate_scienceInstruments_extra` to add photon-counting detector-specific attributes

Stellar Photometry Handling

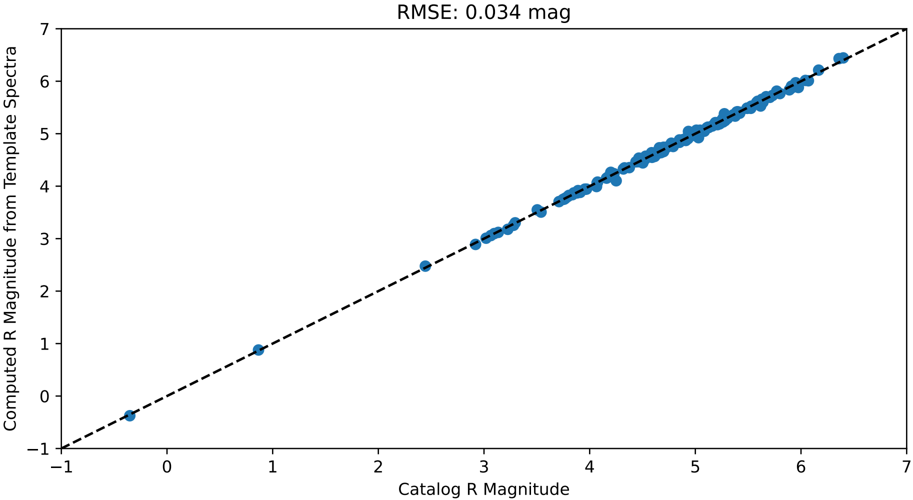


HWO Mission Star List Spectral Templates

Stellar flux is computed as:

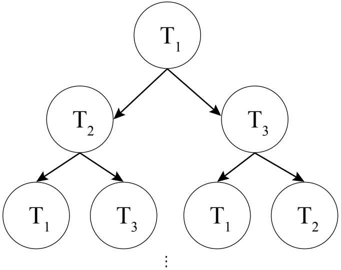
1. Match the closest available catalog spectrum to the target's spectral type
2. Identify the closest (in wavelength) band to observing band, for which the original star catalog provided a magnitude
3. Re-normalize template spectrum to target star's magnitude
4. Integrate the spectrum over the observing band

Stellar Photometry Across Bands

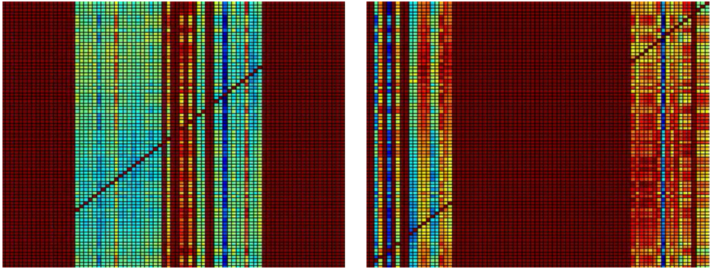


R mag computed from V mag-normalized template spectra for HWO Mission Stars

Traveling Spacecraft Problem



Visit graph for 3 target pool.



Adjacency matrices at two different times

The cost of transitioning from target i to target j is:

$$A_{ij} = \frac{\sum_k a_k m_k}{\left(1 - B^j_{keepout}\right)(1 - \delta_{ij})}$$

- m_k: Cost/benefit metrics/heuristics
- a_k: Weights
- $B^j_{keepout}$: 1 if target j is in keepout, else 0

See: Savransky, Kasdin, and Cady, "Analyzing the designs of planet finding missions", 2010

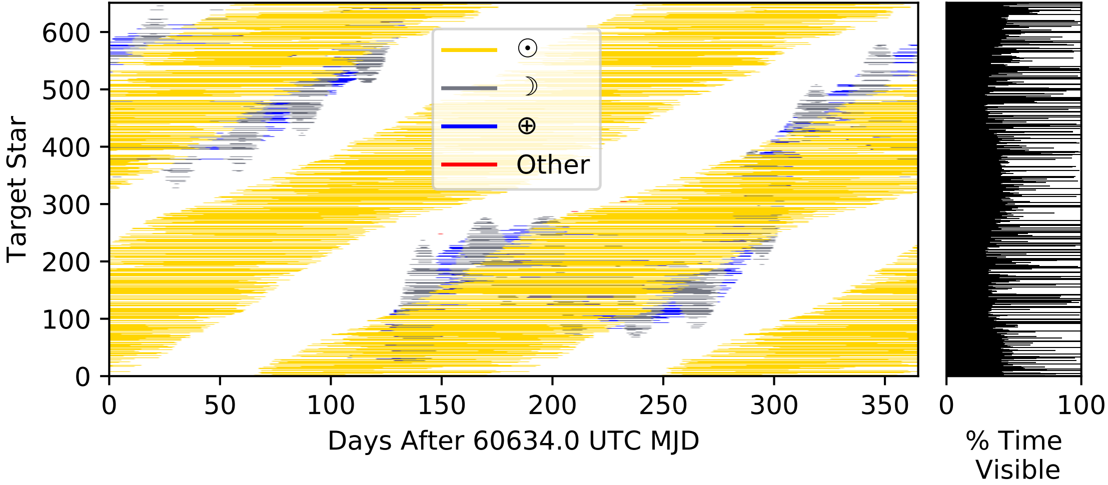
Costs and Benefits

- Penalize for long slews (if using starshade)—angle between look vectors is an acceptable heuristic, but actual fuel/time costs are better
- Reward for accumulating completeness
- Penalize long integration times (minimize known astrophysical noise sources)
- Reward for repeat observations of prior detections (up to some maximum, penalize after)
- Reward observations of hard to schedule targets (large fraction of time spent in keepout)
- Penalize targets likely to give more false positives

$$A_{ij} = \left[a_1 \frac{\cos^{-1}(u_i \cdot u_j)}{2\pi} B_{inst} + a_2 \mathrm{comp}_j - a_3 e^{t_c - t_f} B_{unvis} + a_4 B_{vis}(1 - B_{revis}) \right.$$
$$\left. - a_5 B_{revis} \left(\frac{N_j}{N_{req}}\right)(N_j < N_{req}) - a_6 \frac{\tau_j}{\mathrm{vis}_j}\right] / (1 - B_{keepout})$$

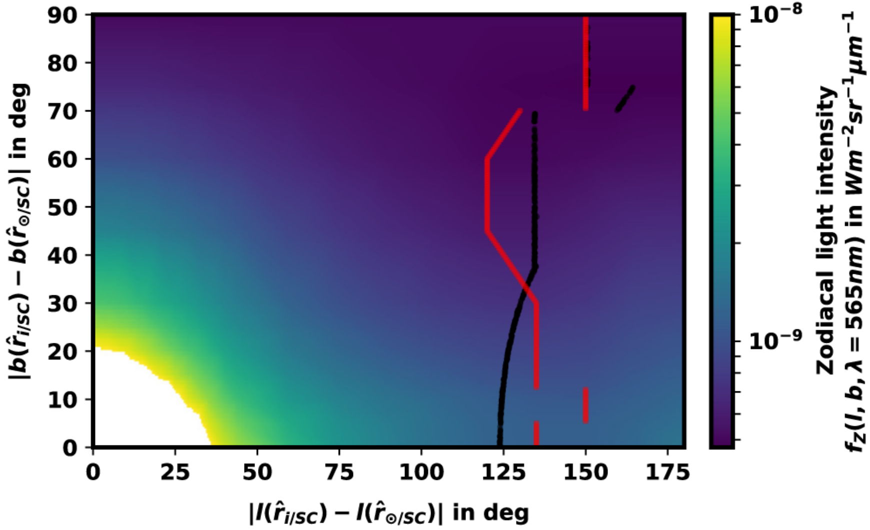
See: Savransky, Kasdin, and Cady, "Analyzing the designs of planet finding missions", 2010

Keepout Constraints



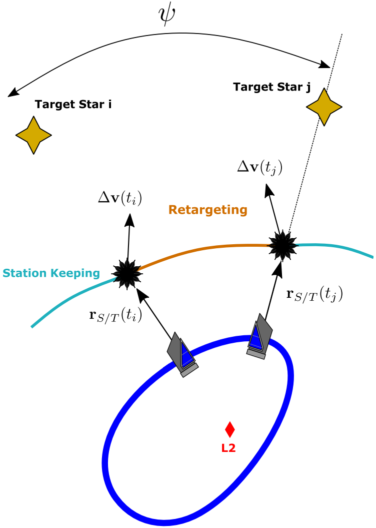
Targets are observable in white regions of the graph. The sun keepout may be due to direct sun avoidance, starshade glint avoidance, or solar panel pointing restrictions.

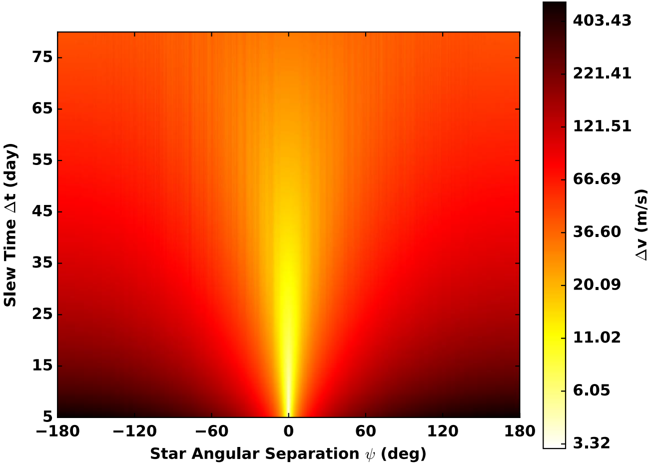
Local Zodiacal Light Minimization



From Keithly et al., "Optimal scheduling of exoplanet direct imaging single-visit observations of a blind search survey", 2020 based on Leinert et al., "The 1997 reference of diffuse night sky brightness", 1998.

Starshades Make Everything Harder

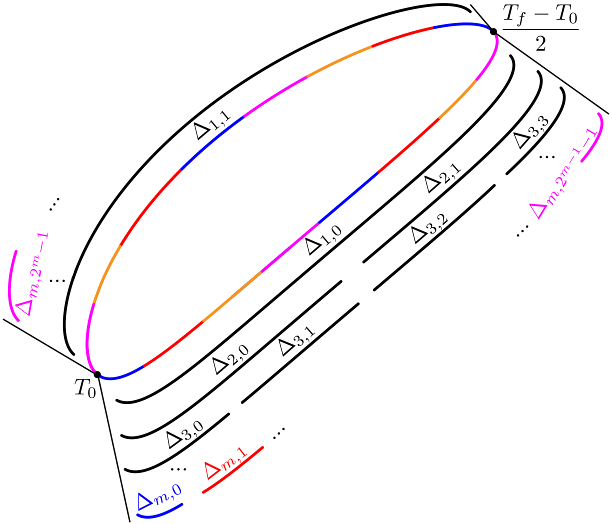




Required slew Δv for impulsive burn model

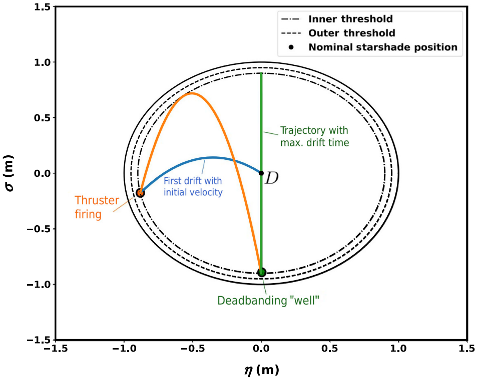
See: Soto et al., "Parameterizing the Search Space of Starshade Fuel Costs for Optimal Observation Schedules", 2019

Continuous Thrust Slews Make Things *Even* Harder

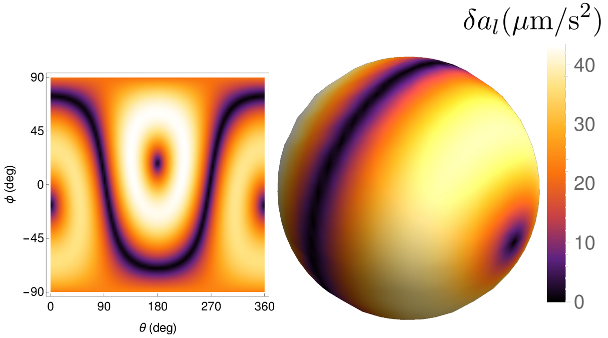


See: Kulik, Clark, and Savransky, "State Transition Tensors for Continuous-Thrust Control of Three-Body Relative Motion", 2023

Stationkeeping Requires Additional Optimization



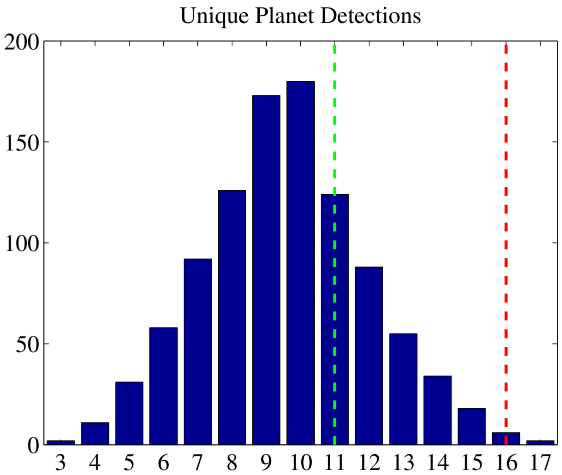
Starshade stationkeeping schematic. σ axis aligned with lateral differential acceleration.

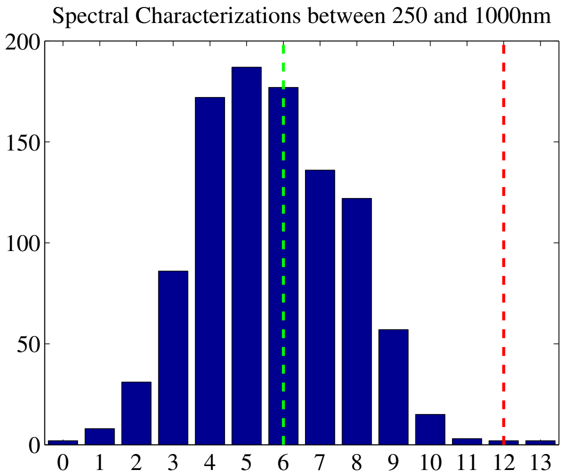


Minimum lateral differential acceleration occurs for targets on poles and equator of unit sphere about the observatory

See: Flinois et al., "Starshade formation flying II: formation control", 2020; Soto, Savransky, and Morgan, "Analytical model for starshade formation flying with applications to exoplanet direct imaging observation scheduling", 2021; Kulik, Soto, and Savransky, "Minimal differential lateral acceleration configurations for starshade stationkeeping in exoplanet direct imaging", 2022

Schedule Validation via Random Walks

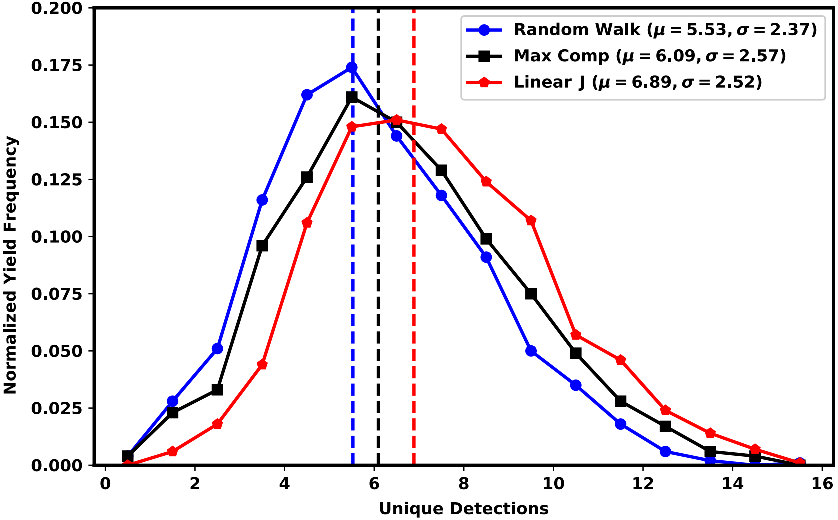




Comparison of yield from randomized visit order (blue), choosing the next highest completeness target (green) and automated scheduler (red).

See: Savransky, Kasdin, and Cady, “Analyzing the designs of planet finding missions”, 2010

More Random Walks



From: Soto et al., "Parameterizing the Search Space of Starshade Fuel Costs for Optimal Observation Schedules", 2019

Validation via Multiple Planet Populations

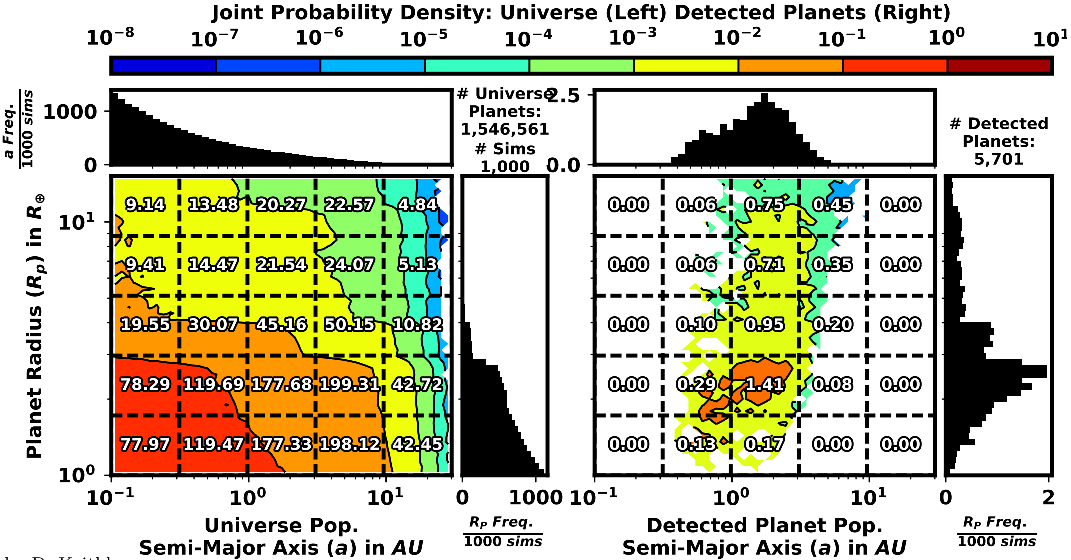


Figure by D. Keithly

Validation via Multiple Planet Populations

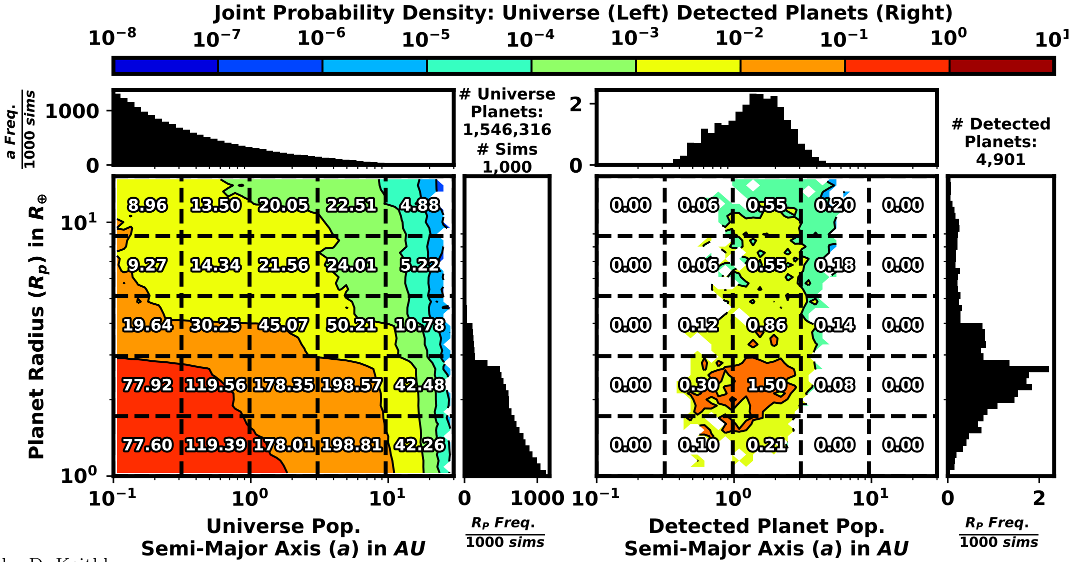
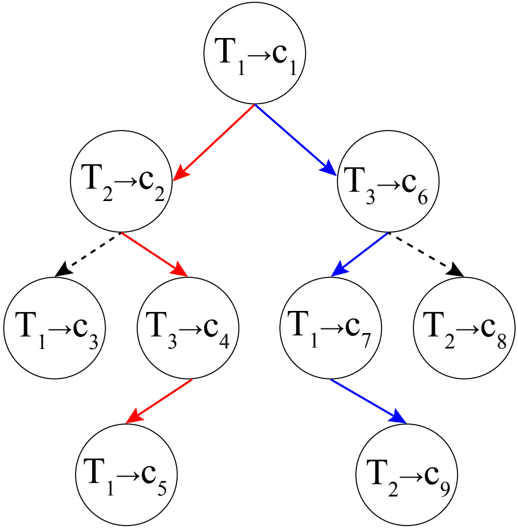


Figure by D. Keithly

Validation via Exhaustive Search



- Exhaustive search may be possible for a limited target list *if* we can prune equivalent branches
- Equivalency is determined by ignoring target order and tracking accumulated completeness from the same set of targets
- For example: red $\equiv$ blue iff

$$c_2 + c_4 + c_5 = c_6 + c_7 + c_9$$

- Stay tuned...

See: Blum and Furst, "Fast planning through planning graph analysis", 1997

Ensuring Reproducibility

- EXOSIMS generates a complete record of all inputs and all defaults filled at runtime
- Intermediate products are cached with filenames based on hashes of the full inputs


```
{"obscurFac": 0.1,
 "shapeFac": 0.7853981633974483,
 "pupilDiam": 4.0,
 "intCutoff": 50.0,
 "stabilityFact": 1.0,
 "use_char_minintTime": False,
 "texp_flag": False,
 "scienceInstruments":
 [{"name": "imager",
   "QE": 0.9,
   "optics": 0.5,
   "FoV": 10.0,
   "pixelNumber": 1000,
   "pixelSize": 1e-05,
   "pixelScale": 0.02,
   "idark": 0.0001,
   "CIC": 0.001,
   "sread": 1e-06,
   "texp": 100.0,
   "Rs": 1.0,
   "lenslSamp": 1.0,
   "Nlensl": 5.0,
   "focal": 103.13240312354819,
   "fnumber": 25.783100780887047}],
"starlightSuppressionSystems":
[{"name": "coronagraph",
  "occ_trans": 0.2,
  "core_thruput": 0.1,
  "core_contrast": 1e-10,
  "core_mean_intensity": 1.0e-11,
  "core_area": 0.0,
  "optics": 1.0,
  "occulter": False,
  "lam": 500.0,
  "deltaLam": 100.0,
  "BW": 0.2,
  "koAngles_Sun": array([  0., 180.]),
  "koAngles_Earth": array([  0., 180.]),
  "koAngles_Moon": array([  0., 180.]),
  "koAngles_Small": array([  0., 180.]),
  "core_platescale": None,
  "contrast_floor": None,
  "IWA": 0.1,
  "OWA": inf,
  "ohTime": 1.0}],
"observingModes": [{"detectionMode": True,
  "instName": "imager",
  "systName": "coronagraph"}]}
```


- Instantiating an optical system with no inputs (all defaults) generates (minimally) this set of inputs.
- Cached products based on this optical system will have a filename including the string `9c437d0035943d38e9abce629bf9bc61`, which is the full MD5 hash of this dictionary.

Ensuring Reproducibility and Avoiding Regressions

Reproducibility is a key challenge in Monte Carlo

but remember that pseudorandom is not truly random

- Random number generator seeds are saved along with simulation outputs, and can be used to replicate random draw sequences
- EXOSIMS allows for the dumping/loading of all randomly generated values when creating synthetic universes

Continuous Integration is required for actively developed projects

EXOSIMS uses both unit tests (run in CI) and end-to-end tests (run offline) to avoid regressions

Some Final Thoughts

- Monte Carlo Mission Modeling is an enormously powerful approach to yield modeling, but requires equally enormous validation efforts
- We are making progress on answering the extent to which differences between summed completeness and MCMM yields are due to real constraints or scheduling inefficiencies, but more remains to be done
- Implementation of an MCMM framework such as EXOSIMS produces numerous useful tools (keepout map generators, exposure time calculators, etc.)
- Open Source Science is great - we should all be doing it

Thanks!

Many Thanks to all EXOSIMS contributors:

Christian Delacroix, Daniel Garrett, Dean Keithly, Gabriel Soto, Corey Spohn, Walker Dula, Sonny Rappaport, Michael Turmon, Rhonda Morgan, Grace Genszler, Patrick Lowrance, Ewan Douglas, Jackson Kulik, Jeremy Turner, Jayson Figueroa, Owen Sorber, Neil Zimmerman, William Balmer, Mario Damiano

Join Us!

`https://github.com/dsavransky/EXOSIMS#contributing`

The EXOSIMS team gratefully acknowledges support by NASA via grants NNX15AJ67G and NNG16PJ24C, and JPL via the SURP program.