An approach to modeling linear and non-linear self-shading losses with pvlib

Will Hobbs[1], Kevin Anderson[2], Mark Mikofski[3], Madison Ghiz[3]

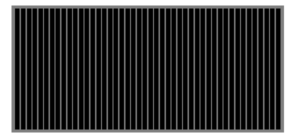
[1]Southern Company, [2]Sandia, [3]DNV

2024 PV Performance Modeling Workshop, May 6, 2024

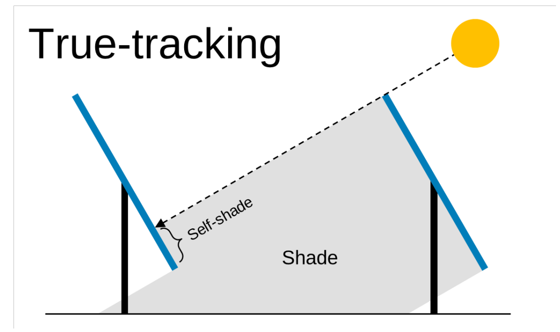
Self-shade (row-to-row shade)

- About 40% of US utility-scale solar has self-shade losses
  - 25% thin-film tracking
  - 18% fixed tilt
    - 2/3 (12% of solar) crystalline → has *non-linear* losses

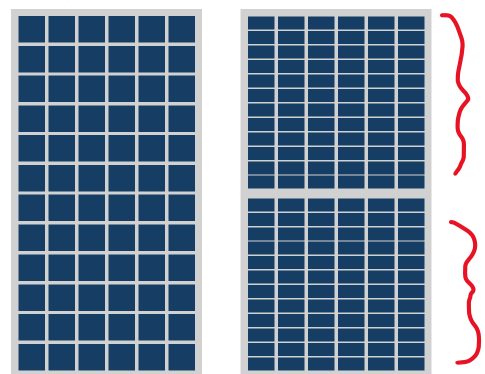
Thin-film



Linear

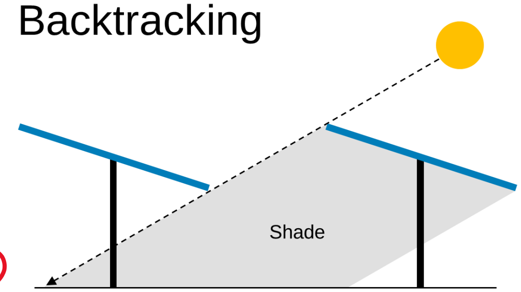


Crystalline | Crystalline, twin



2-in-1

Non-Linear | Non-Linear



How do we model all of this?

Non-linearity
Twin modules (w/ half-cut cells)?
Fractional backtracking?
Tracker issues?

Solar stats: Berkely Lab, Utility-Scale Solar 2023, at https://emp.lbl.gov/utility-scale-solar/

For example…

Easy-ish to model:

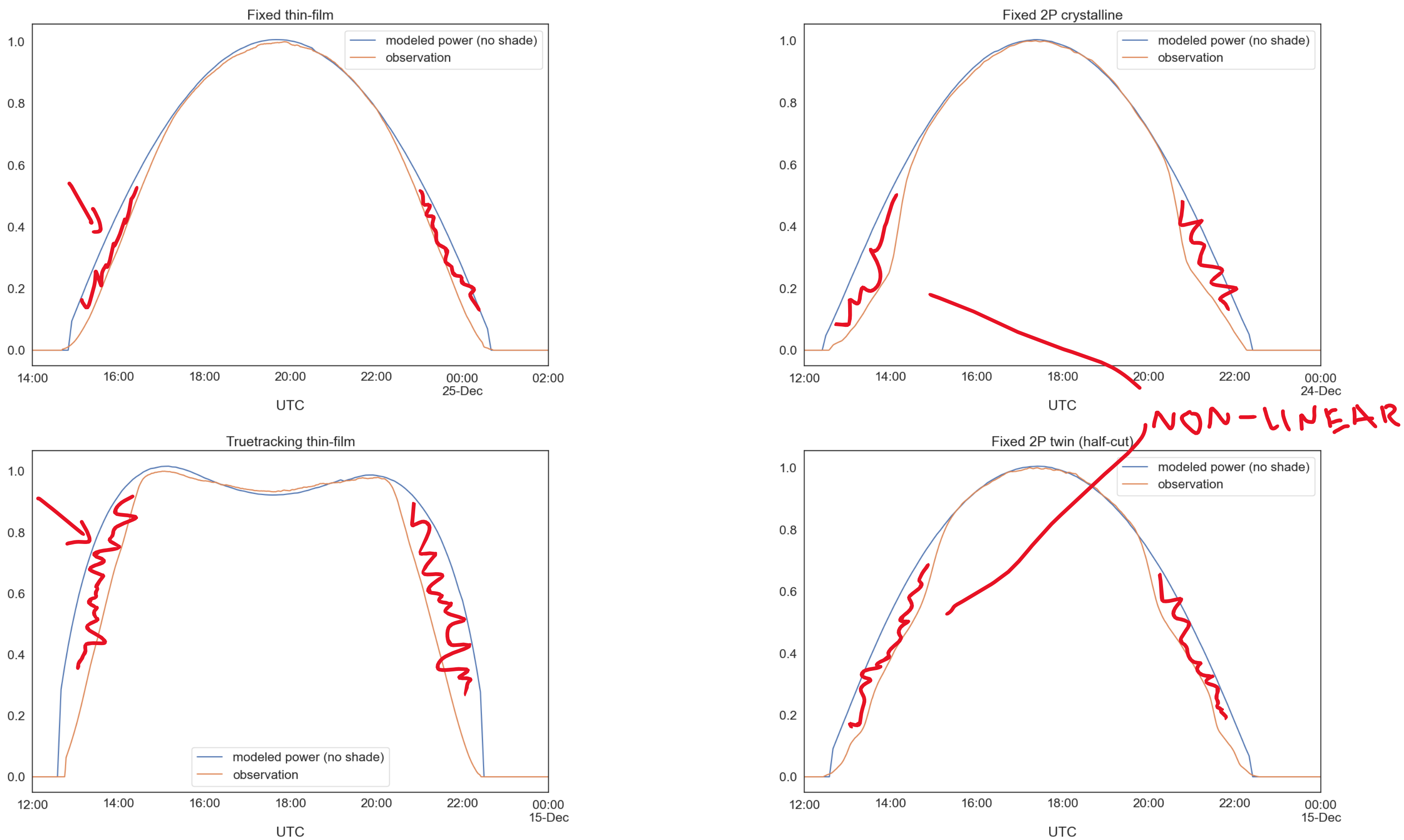
- Fixed crystalline w/ no shade

Hard to model

- Lower sun elevation = shade
- Non-linear losses
- 2P, 2P twin, etc…

What does this look like for power?

Overestimation at low sun angles:



Solution using pvlib

Steps:

1. Calculate shaded fraction*
2. Calculate power loss**
3. Done!

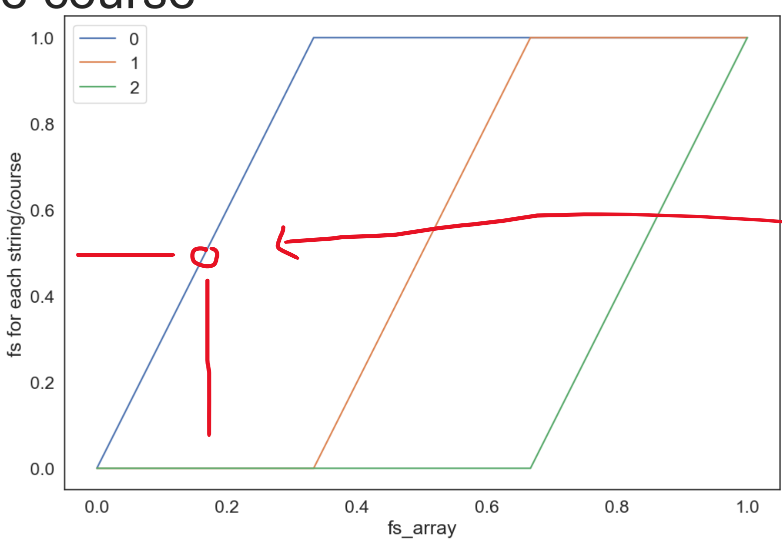
* Start with array shaded fraction, then calculate for each “course” (i.e., string) of modules

** actually, effective irradiance instead of power loss

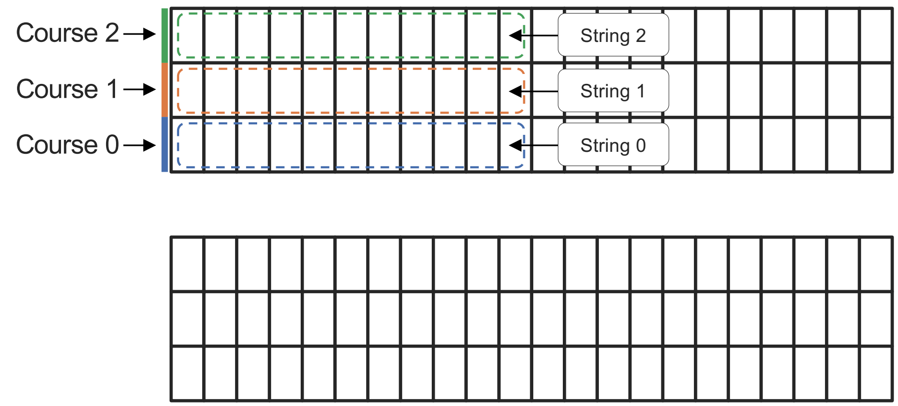
Shaded Fraction

- Use *shaded_fraction_front* from infinite_sheds.get_irradiance() for array shaded fraction
- Then calculate shaded fraction in each course*

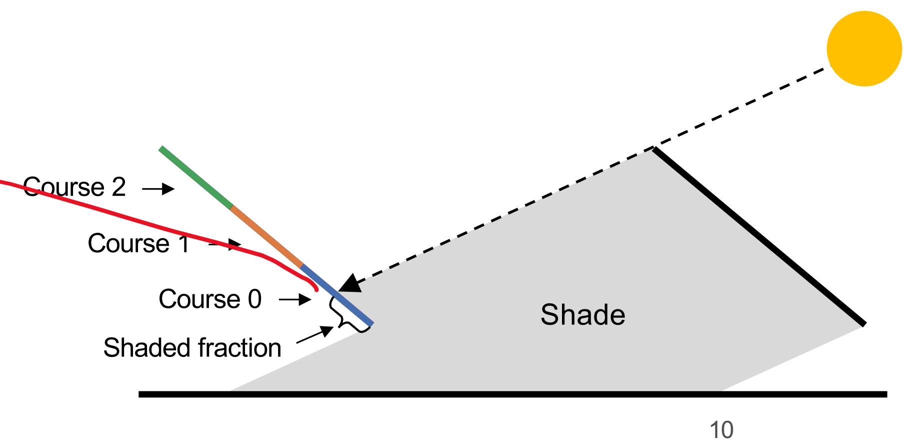
* For our model, each string can only be in one course



Top-down view of 3P system (2 rows)



End view of 3P system (2 rows)



Twin modules?

- Double the number of effective modules up the side of a row
- For example, 2P twin → 4P effective

Calculate power (effective irradiance) loss

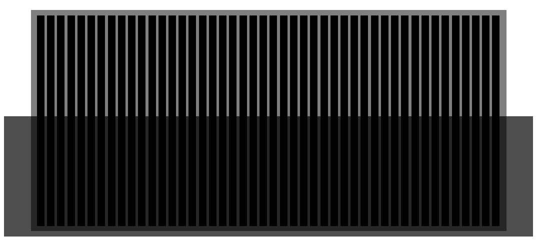
- Thin-film? Loss of direct irradiance linear with shaded fraction, *fs*
- Crystalline? Loss of direct irradiance *across the first row of cells*

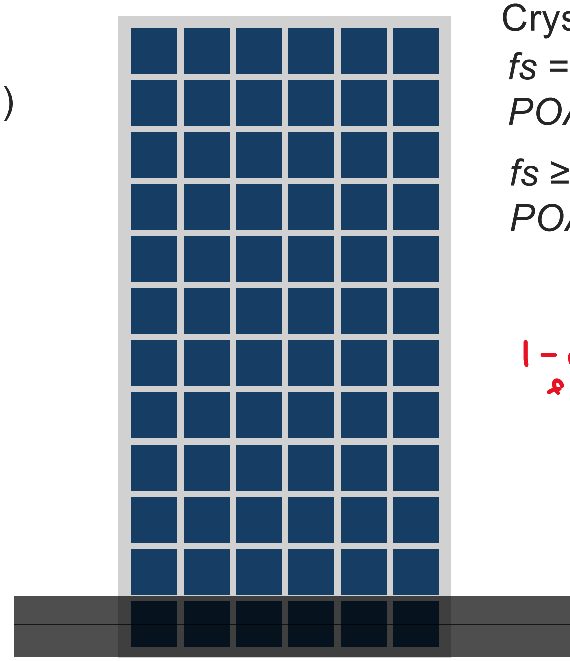
(and treat twin modules as 2 modules)

Thin-film (linear):

fs = 1/2

POA_{direct} reduced by 1/2





Crystalline (non-linear):

fs = 1/24

POA_{direct} reduced by 1/2

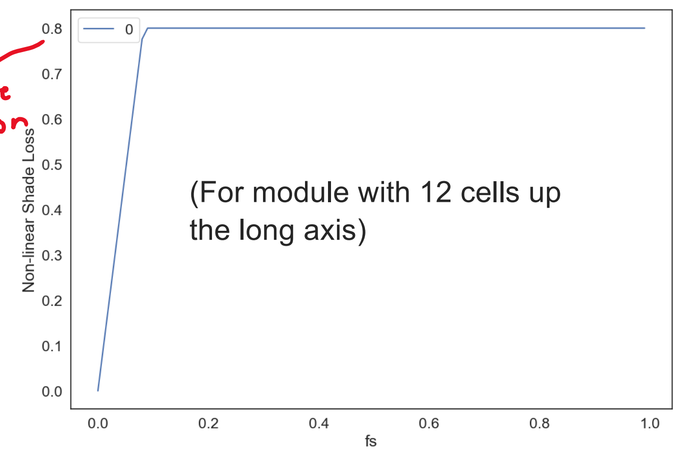
fs ≥ 1/12

POA_{direct} reduced by 100%

1/2 of a cell

module is limited to cells w/ most shade (in portrait...)

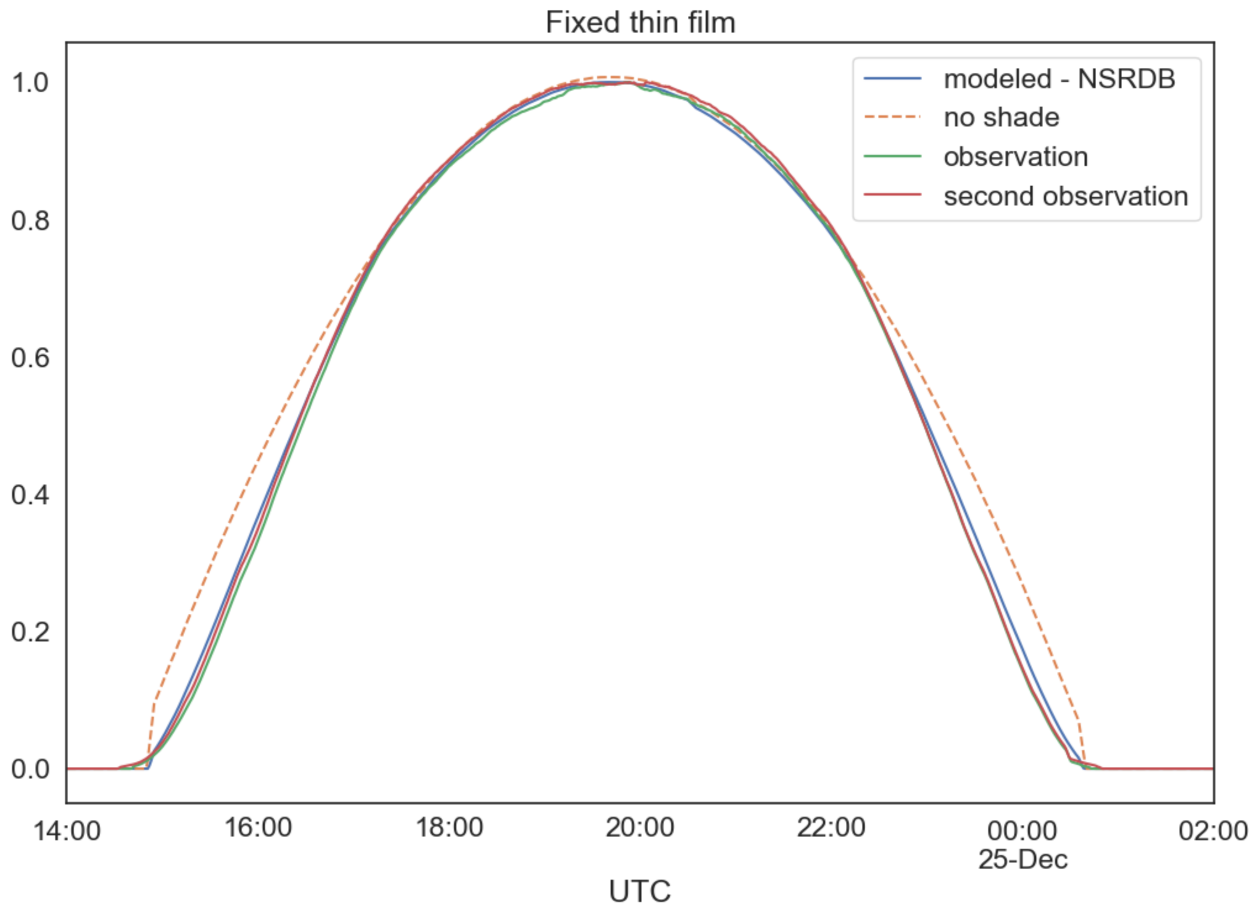
1 - diffuse fraction



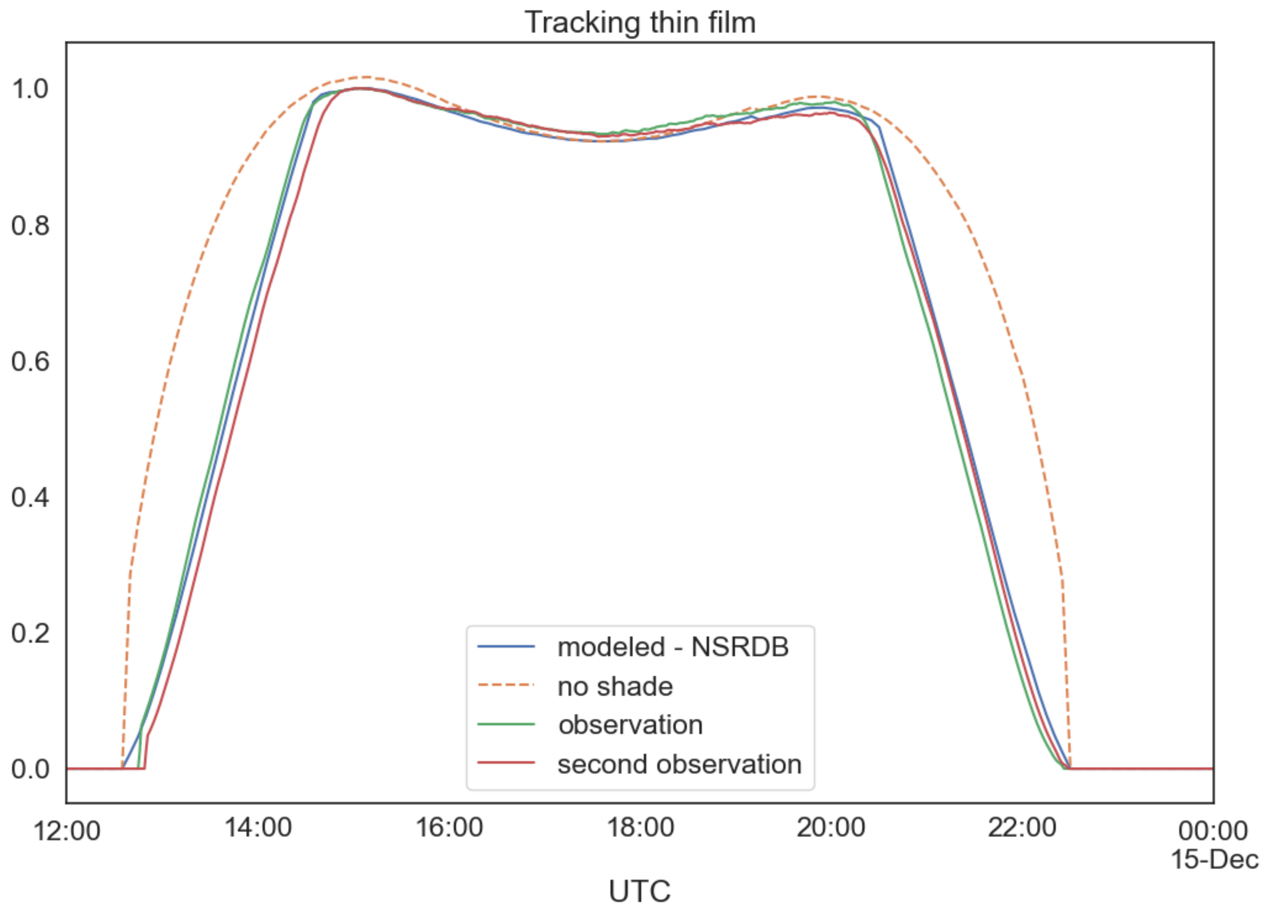
Anderson, Kevin. 2020. Maximizing Yield with Improved Single-Axis Backtracking on Cross-Axis Slopes: Preprint. Golden, CO: National Renewable Energy Laboratory. NREL/CP-5K00-76023. https://www.nrel.gov/docs/fy20osti/76023.pdf.

Some results

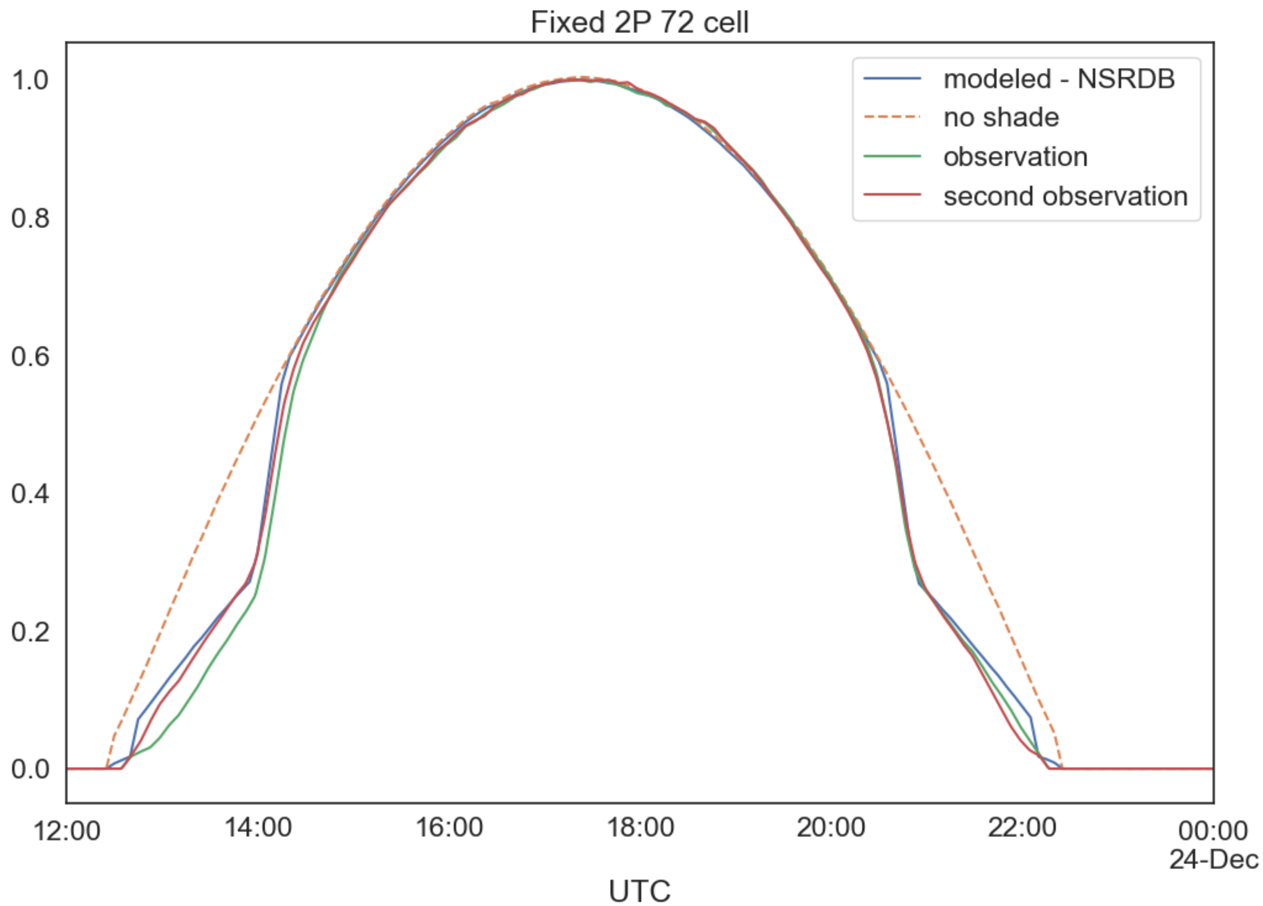
with observations from 2 inverters per plant



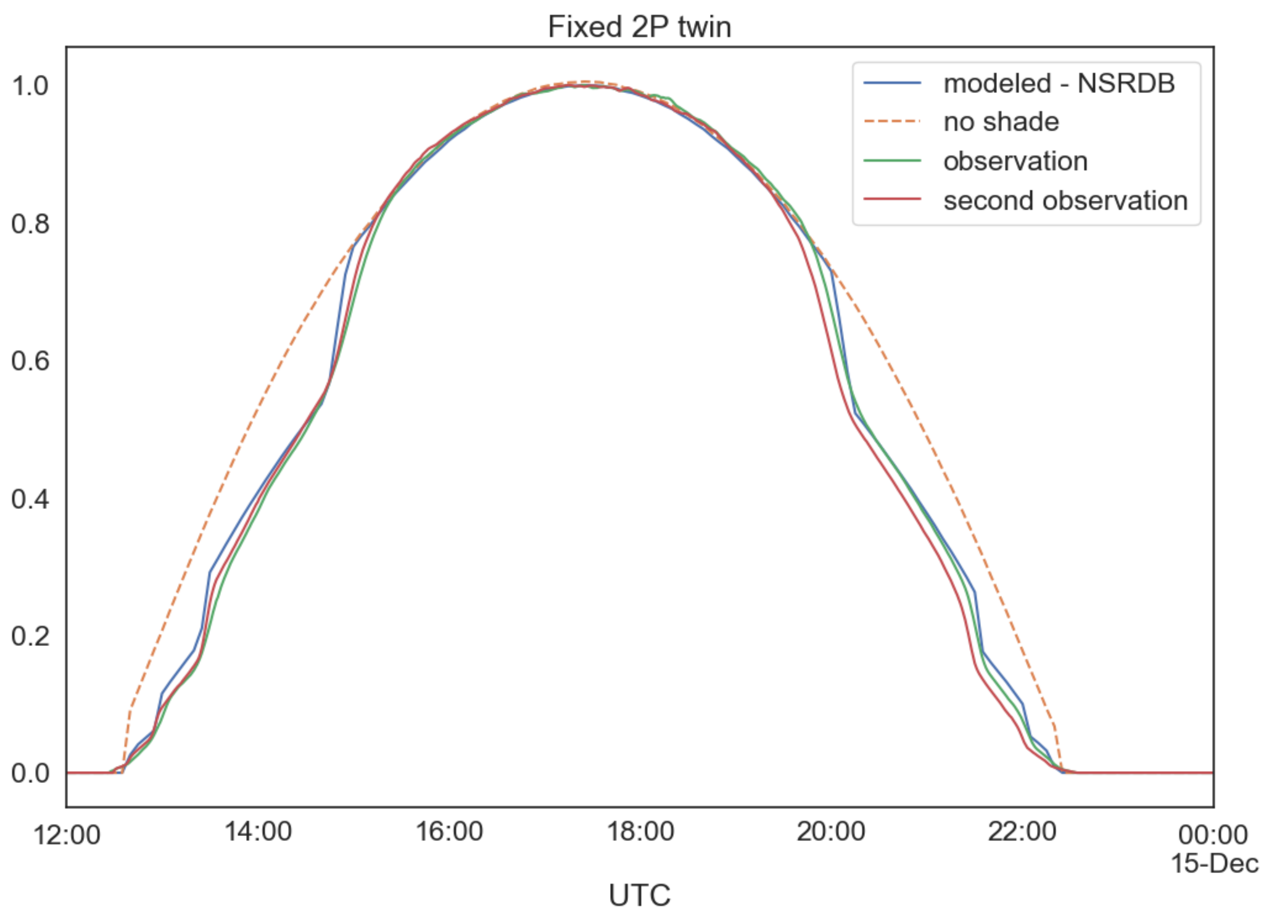
1.9 % annual difference (NSRDB)



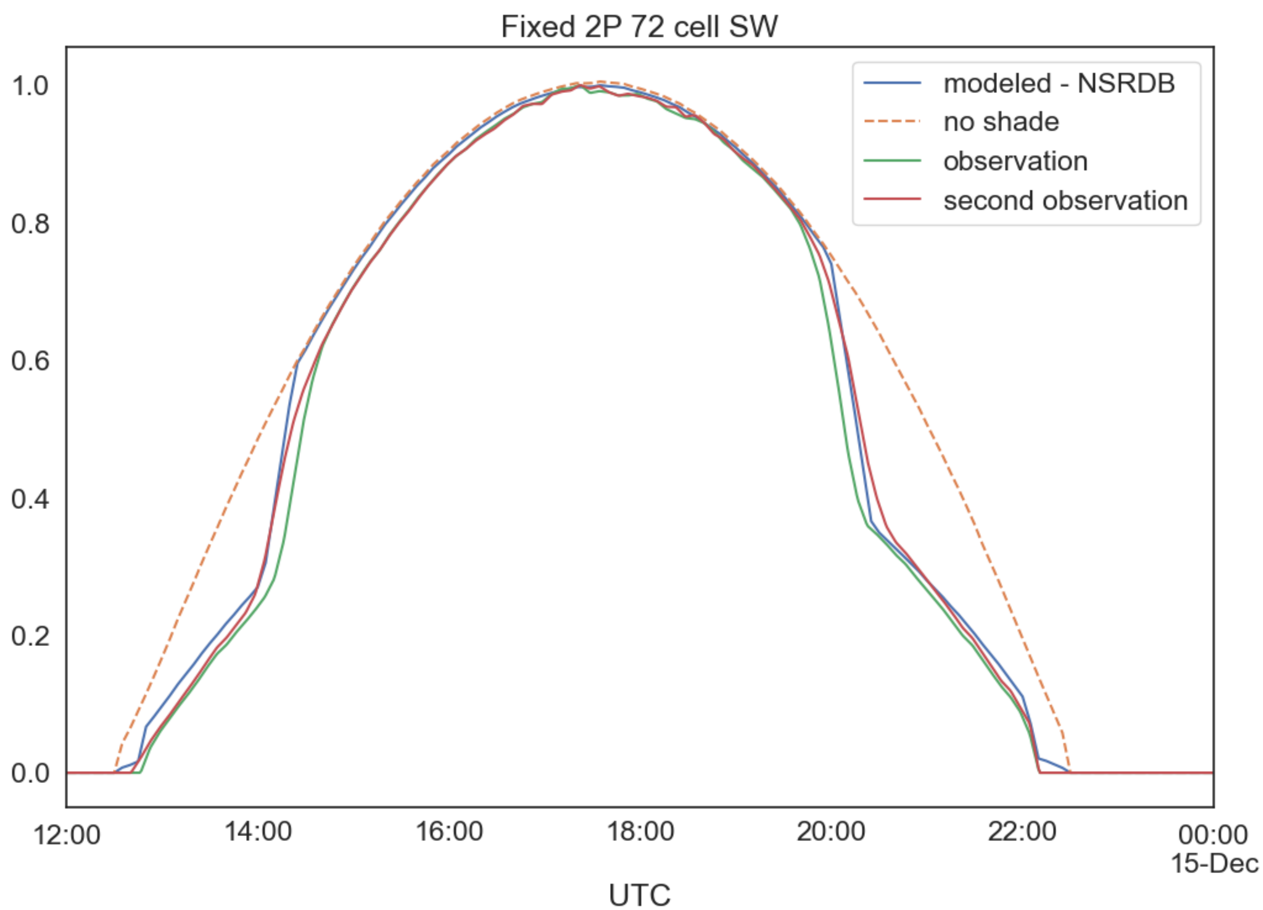
8.5% annual difference (NSRDB)



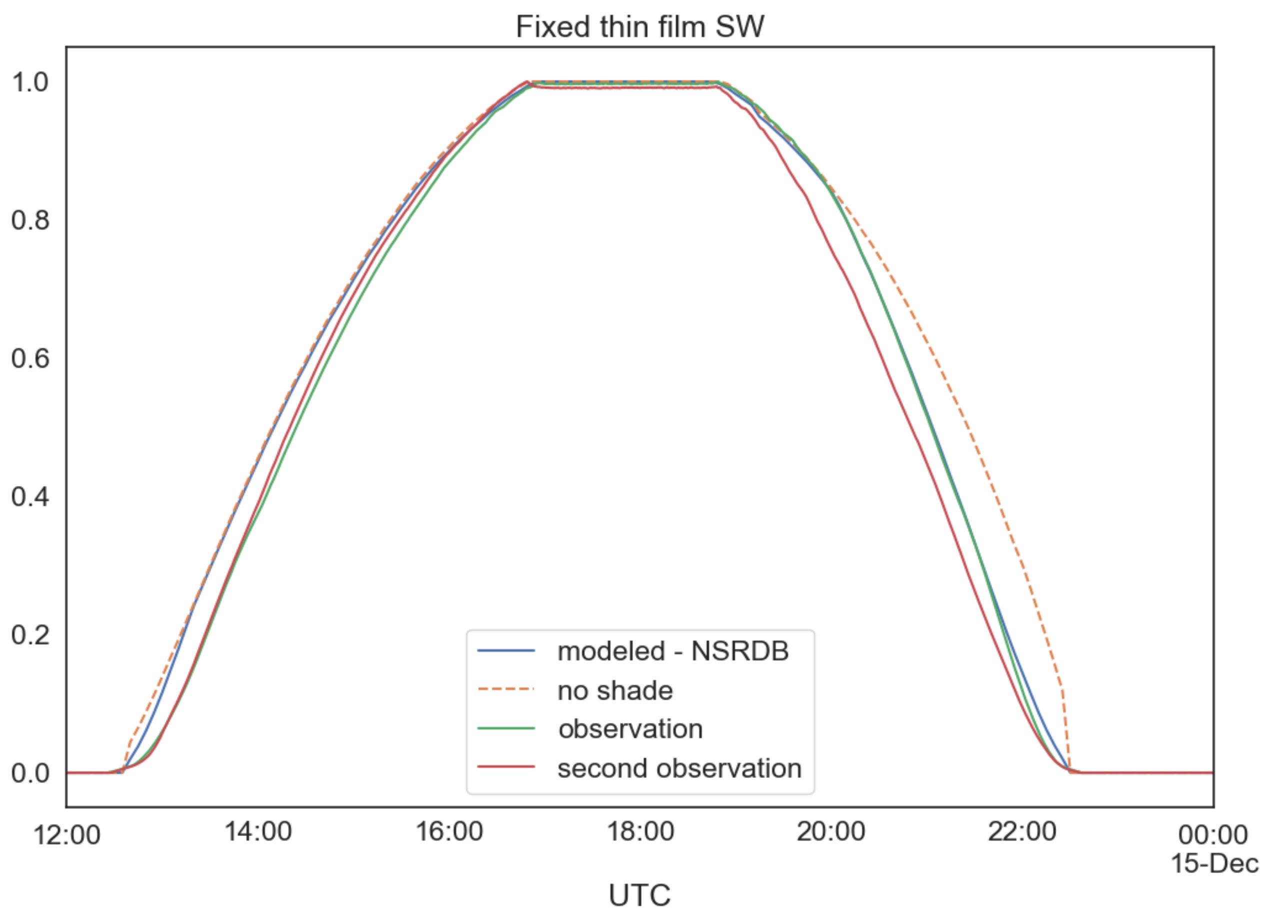
1.9% annual difference (NSRDB)



2.5% annual difference (NSRDB)



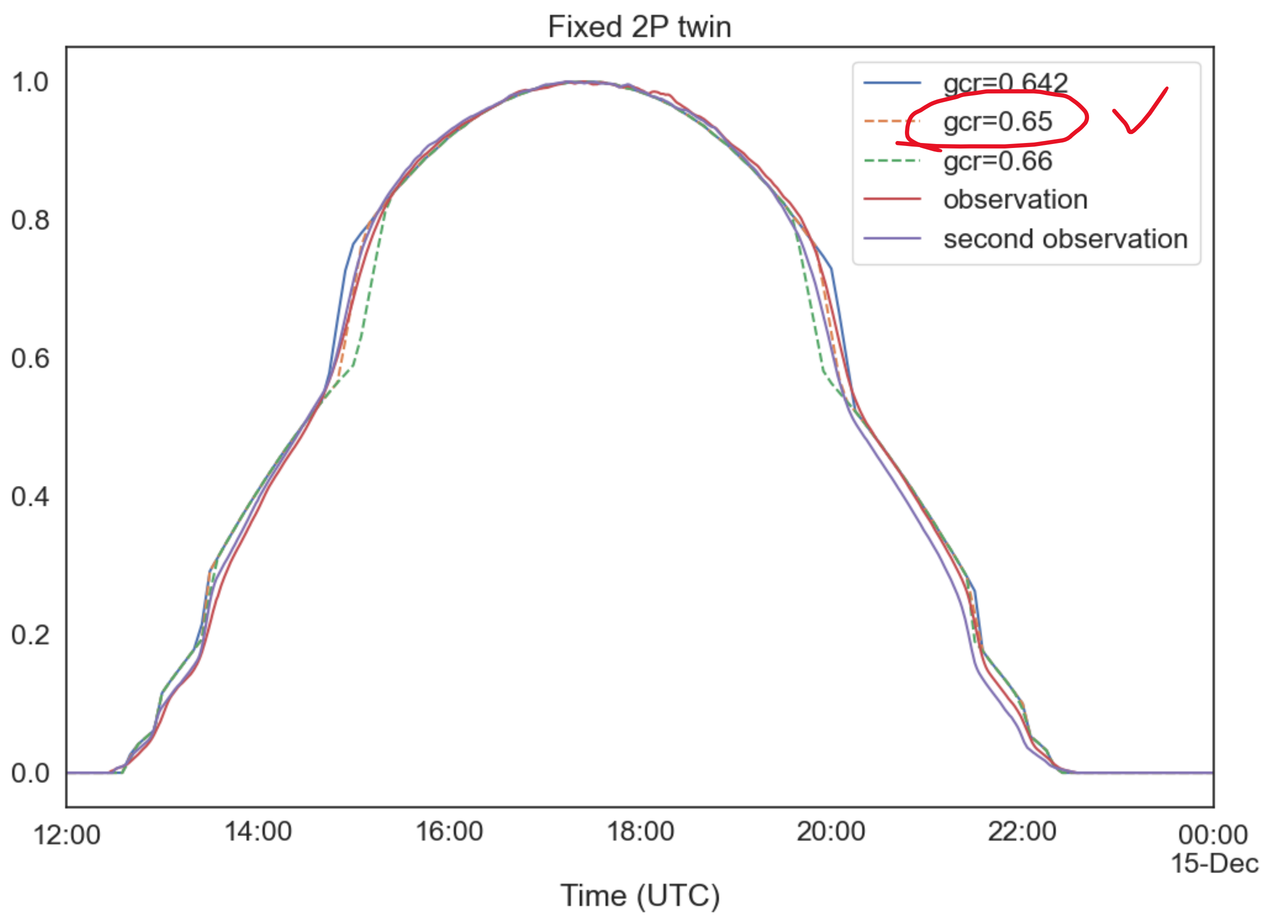
2.7% annual difference (NSRDB)



2.3% annual difference (NSRDB)

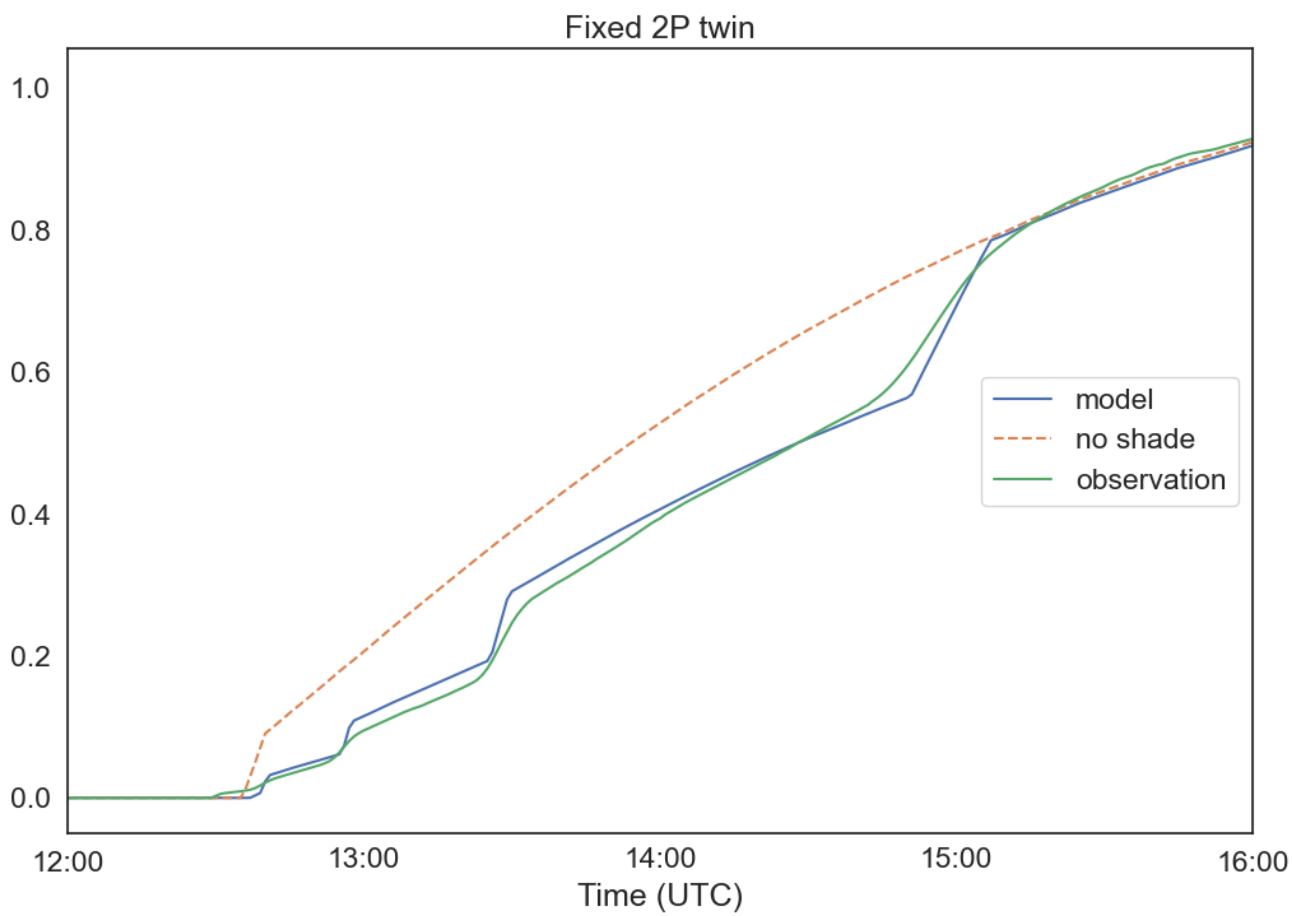
Back to the 2P twin plant

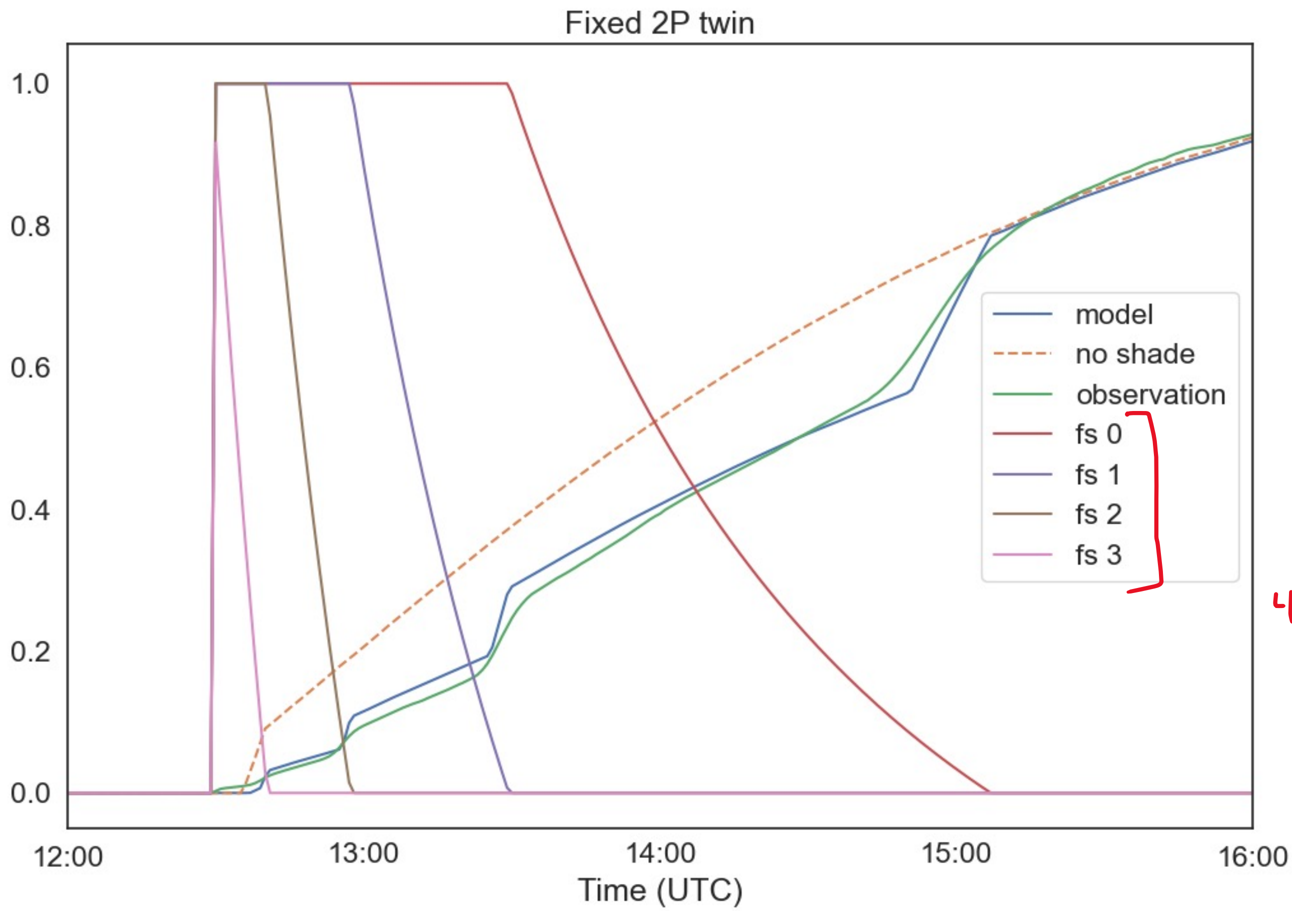
let's look at GCR

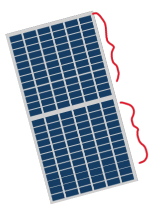

Fixed 2P twin
gcr=0.642
gcr=0.65
gcr=0.66
observation
second observation
1.0
0.8
0.6
0.4
0.2
0.0
12:00
14:00
16:00
18:00
20:00
22:00
00:00
15-Dec
Time (UTC)

Let’s look at the steps

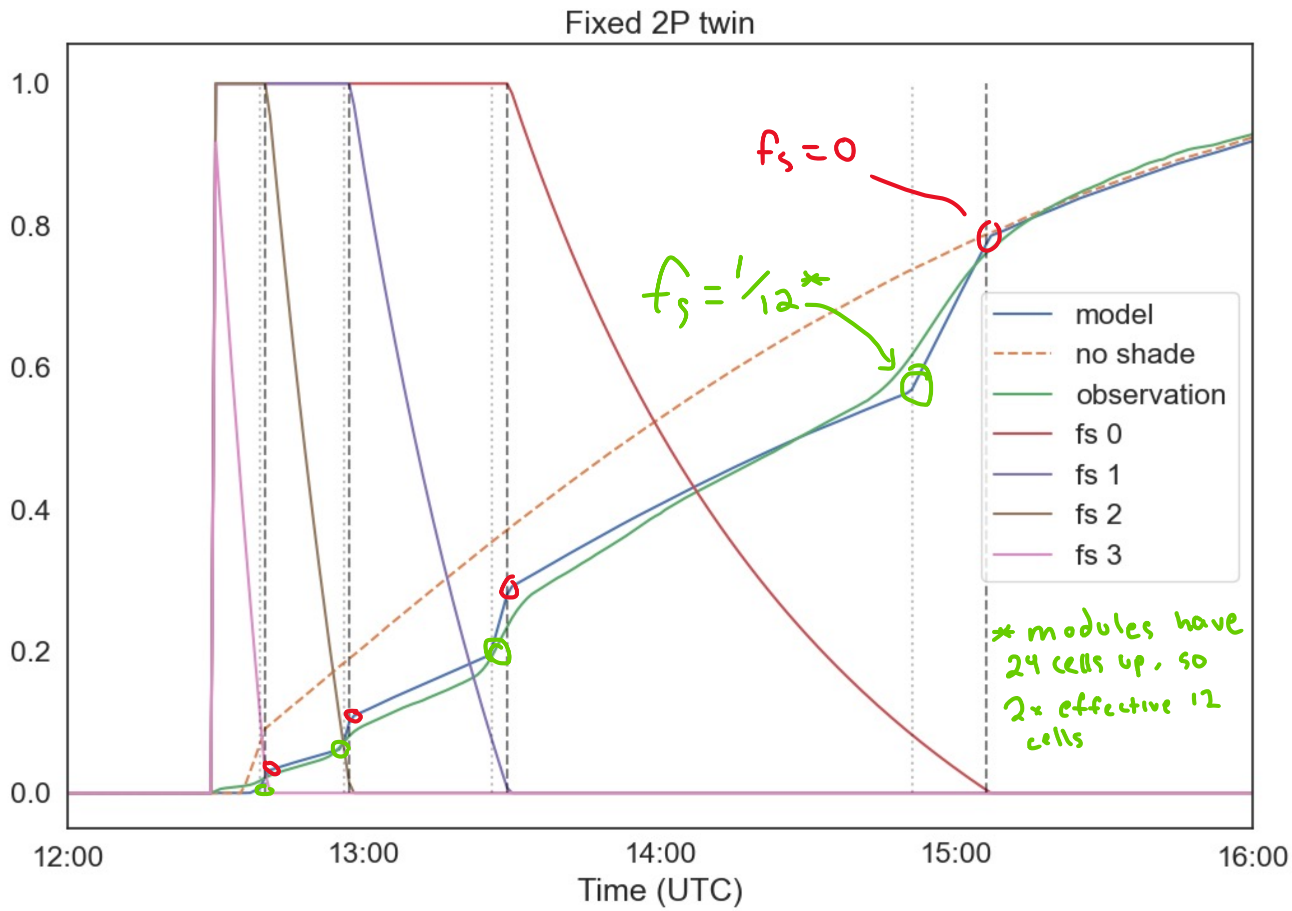
zoom in on the morning


Fixed 2P twin
1.0
0.8
0.6
0.4
0.2
0.0
12:00
13:00
14:00
15:00
16:00
Time (UTC)
model
no shade
observation





4 effective courses b/c twin



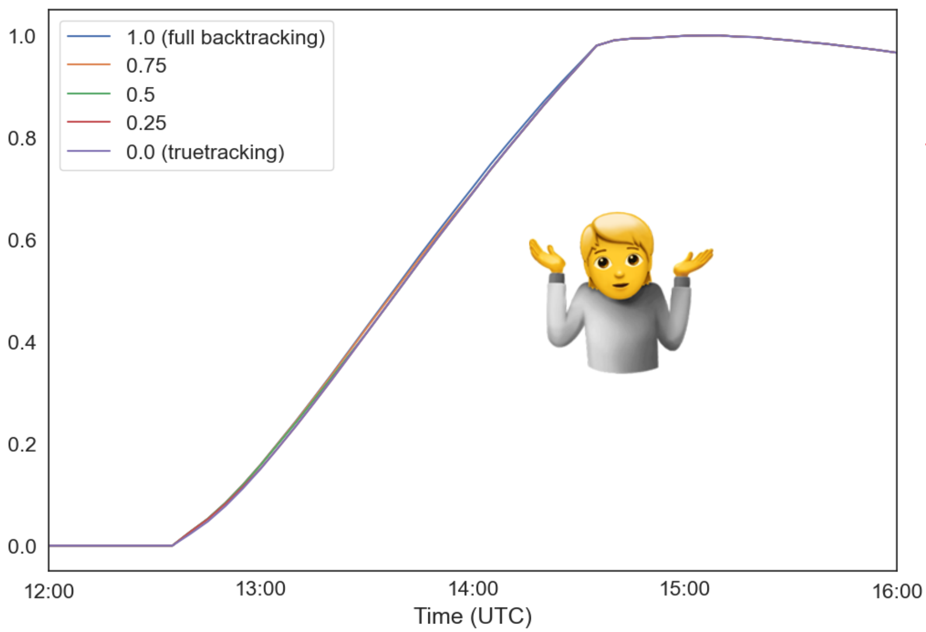
Fixed 2P twin
1.0
0.8
0.6
0.4
0.2
0.0
12:00
13:00
14:00
15:00
16:00
Time (UTC)
model
no shade
observation
fs 0
fs 1
fs 2
fs 3
fs = 0
fs = 1/12*
* modules have 24 cells up, so 2x effective 12 cells

Other applications

fractional backtracking,
backtracking faults

Fractional Backtracking

- For twin modules, especially 2P (effectively 4P conventional) backtracking to allow shade for a fraction of the array might make sense (maybe even truetracking)
- Let's model a 2P twin example:

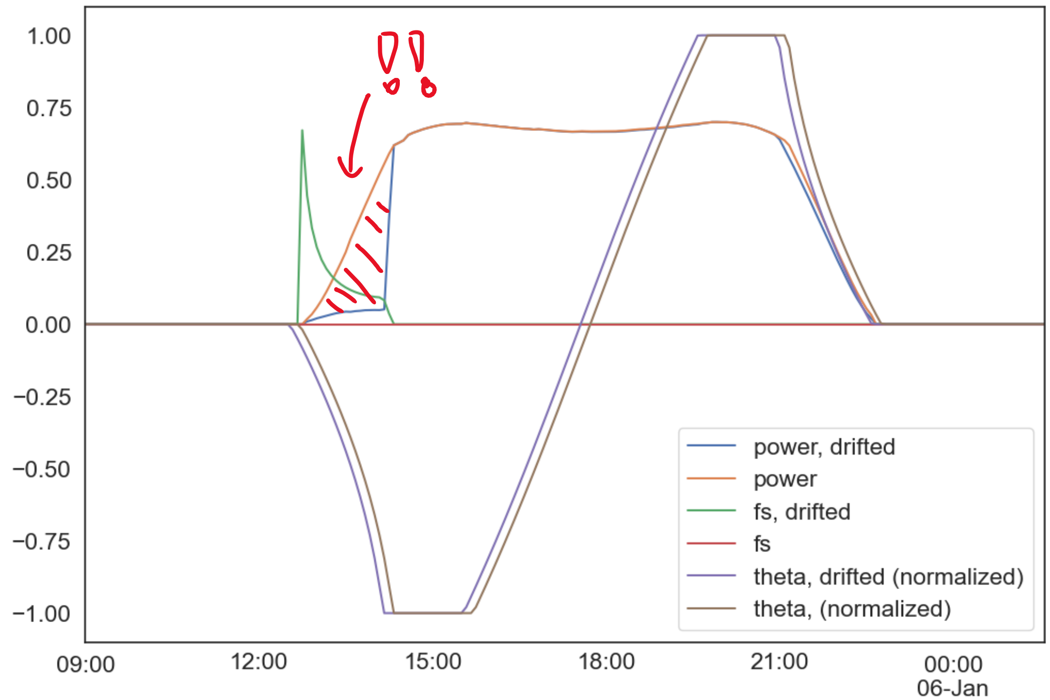


Mode	Annual Energy kWh (NSRDB)
Full backtracking (1.0)	2058
0.75 backtracking	2050
0.5 backtracking	2046
0.25 backtracking	2044
0.0 (truetracking)	2043

Resource-dependent??

Misaligned trackers (clock drift)

- Tracker clock is 10 minutes fast
- 1P 72-cell crystalline



~5% annual losses!
(NSRDB)

Code availability

it’s open-source!

https://github.com/williamhobbs/2024_pvpmc_self_shade

Repo includes everything shown here:

- Functions
- example notebooks with plots
- anonymized observation data and specs

Functions

- shade_fractions(fs_array, eff_row_side_num_mods)

- non_linear_shade(n_cells_up, fs, fd)

- plant_power_with_shade_losses(
  resource_data,
  **plant_data,
  surface_tilt_timeseries,
  surface_azimuth_timeseries,
  use_measured_poa,
  use_measured_temp_module)

Dictionary of specs

(https://github.com/williamhobbs/pv-plant-specifications)

Optional

INCLUDES BIFACIAL!!

Questions?

whobbs@southernco.com

https://github.com/williamhobbs/2024_pvpmc_self_shade

Southern Company

Room for improvement

- Account for “inactive bands” at module edges (or, at least, see how much it matters)
- Crystalline modules in landscape
- Unconventional crystalline module designs (shingled cells, etc.)
- Comparison with single diode model
- Slopping terrain (https://github.com/pvlib/pvlib-python/pull/1962 by echedey-ls will help)
- isotropic looks better than haydavies, even in periods with no direct shade – why?