Our approach

Calibration software

mwa_hyperdrive is our bread-and-butter calibration software for MWA.

A scientific paper has recently been submitted such that hyperdrive can be cited. However, our results can be reproduced by using the "SDC3" of the hyperdrive GitHub repo. The steps for using hyperdrive are well documented on the GitHub project page or on the dedicated website. hyperdrive is open source and licensed under the Mozilla Public License version 2.0.

We made a number of improvements and accommodations to hyperdrive for SDC3:

- The ability to read, calibrate and write out single-polarisation data
  - We normally only deal with full-polarisation data, so Jones matrices are assumed as the "unit" of a visibility throughout the code. Using only a single polarisation meant that calibration would always fail, because all our Jones matrices were singular.
- The ability to not write out precessed UVWs
  - uvfits and measurement set formats expect their UVWs to be in the J2000 frame, but OSKAR appears to only write out UVWs in the observation frame.
- Including an "Airy disk" beam model, which was used for the SDC3 data

Bespoke software

Despite all these modifications, we still needed to massage the data. A new repo was created to handle two bespoke tasks:

- hyperdrive expects the visibilities to live in a single file. Attempting to read data from multiple files makes things much more brittle, as metadata across all files must be validated before they can be safely used. sdc3_vis_convert unpacks all of the visibilities from the 901 files and packs them into arbitrary time- and channel-dimensioned files. For our purposes, we made 12 timestep, full bandwidth files (120 of them for 1440 supplied files).
- A small offset between the supplied UVWs and our generated UVWs was noticed. We're not exactly sure how this manifested, but we were able to robustly match the supplied UVWs when using an additional time offset. sdc3_time_offset finds this time offset, and found that the offset itself changes linearly with SDC3 timestep. This rate of change is small, so we use a single number offset.

This code can be found at: https://github.com/cjordan/sdc3_vis_convert

Accessing the EoR

After creating our 12 timestep, full bandwidth files and finding the time offset according to the UVWs, we investigated calibration. We found no gain errors, so we abandoned calibration. We also found that any ionospheric offsets that are present are too small to be noticeable. Instead we focused most of our time on improving our sky model to improve the quality of the sky-model-subtracted visibilities.

We found that the SDC3-supplied LoBES sky model did not work as well as another version of LoBES we managed to locate. After stripping LoBES sources that did not appear to be in the

SDC3 data, we ran the aegean source finder on a cleaned image to obtain a sky model for the TRECS sources. We then iterated on improving the sky model until our subtracted visibilities looked good in images as well as power spectra.

Power spectrum estimation and gridding

The power spectrum estimation methodology followed that used by the CHIPS software for MWA data (Trott et al 2016, MNRAS), with updated parameters relevant for SKA. The data were split into even/odd timesteps for power spectrum estimation, and to remove noise power bias. The calibrated and subtracted visibilities were gridded on the uv-plane using a Blackman-Harris gridding kernel matched to the instrument FOV, along with a separate weights grid. The uv-grid resolution was 4 lambda at the central frequency, and data were gridded out to 600 lambda. This produced two 300 x 300 x 901 channel cubes for both the visibilities and the weights (odd/even). For each uv-cell, a 4th-order polynomial was fitted to the real and imaginary components of the spectrum and subtracted, in order to reduce residual diffuse emission. A 4th-order polynomial has sufficiently long coherence lengths to not affect the cosmological signal. The cubes were then split into 6 sets of 300 x 300 x 150 channel cubes for each redshift range. The data were Fourier Transformed along the spectral direction, after weighting with a Blackman-Harris window.

Visibility cubes were normalised by their weights, and then cylindrically-averaged and squared to produce the final power spectra. Noise was calculated using the difference set of visibilities.

Pipeline

A nextflow pipeline to reproduce our method also lives at:
https://github.com/d3v-null/sdc3-pipeline

Docker build command

```
git clone https://github.com/MWATelescope/mwa_hyperdrive -b SDC3
cd mwa_hyperdrive/docker
docker build . -t chjordan/hyperdrive:SDC3 –build-arg
NVIDIA_VERSION=<the version of your NVIDIA driver>
```

Docker run command

```
docker run --gous all -it –rm -v <path to SDC3 data made with
sdc3_vis_convert>:/SDC3 chjordan/hyperdrive:SDC3
```

sdc3_vis_convert Commands

```
sdc3_vis_convert <path to input vis files> --output <path to single
output vis file> --timesteps <the timestep indices to be read and
written out>
# The frequencies of the output are dictated by the input files.
```

hyperdrive Commands

```
hyperdrive vis-subtract /sdc3.toml --data <path_to_vis_data>
--outputs <path_to_subtracted_vis_data>
```