2017.04.11 sh170411 nav reconstruction using pandas and numpy -

▼ 2017.04.11 sh170411 nav reconstruction using pandas and numpy - recv_iwg1.ipynb

- ▼ Goal
  - IWG1 was not distributed to payload for much of the sh170411 flight.
  - We need to find the correspondence between the part of the flight that was was available, and a reliable time measure on S-HIS (systemTimer).
  - Then we reconstruct IWG1 navigation as FBF files for the earlier part of the flight
  - Finally, we re-run the tail end of the scatter-gather processing, since it can be nav-sensitive (due to segments.int4).

- ▼ evouiyk,evoulpr,evouogy,evouqyf,evoutpm,evouwgt,evouyya,evovbph,evovego,evovgxv,evovlfo - see go_sh170411.sh

  - ```
    #!/bin/bash
    export AIRCRAFT_TYPE=871
    unset DRRTV_TYPE
    export VESSEL=/home/rayg/Repos/svn/SHIS/vessel/MASTER
    # test -n "$DRRTV_TYPE" || export DRRTV_TYPE=HS3
    # test -n "$AIRCRAFT_TYPE" || export AIRCRAFT_TYPE=872
    # export DRRTV_HOME=/home/rayg/svn/SHIS/vessel/fortran_SHIS_DR_RTV_package
    # export SHIS_CAL_HOME=/home/rayg/Repos/svn/SHIS/vessel/MASTER

    time nice bash /home/rayg/svn/SHIS/bin/sg_sdr_shis.sh /data/rdr/shis/dock/sh170411/
    {evouiyk,evoulpr,evouogy,evouqyf,evoutpm,evouwgt,evouyya,evovbph,evovego,evovgxv,evovlfo}.rsh
    ```

  - rdr20170411T161346end20170411T220220sdr20170411T224336

  - the nav situation is not looking so fun... looking like a lot of flight did not have nav

    [6:02]
    let me confirm that, but @steve we may need full nav records (CSV?) and I get to figure out how to patch the nav FBF files and regenerate footprints

    steve [6:07 PM]
    is this maybe related to the inflight dashboard not being available?

    rayg [6:07 PM]
    yeah

    [6:07]
    hypothesis is that NASDAT was not talking to payload for much of the flight, resulting in both nav not coming in and status not going out

    [6:08]
    result: somebody owes us a beer

    [6:08]
    I'm doublechecking however

    rayg [6:13 PM]
    uploaded this image: Pasted image at 2017-04-11, 6:12 PM
    Add Comment

    rayg [6:15 PM]
    min(instrumentLongitude) is -93.11

    [6:15]
    palmdale is longitude -118

    [6:18]
    nav missing from ~1615Z to ~1920Z

    [6:19]
    have to go help with dinner. Let me know what you find in the way of external nav for me to fold in. footprint generator can be run separately, I might have a CSV converter somewhere

    [6:20]
    theory is that we want to use instrumentTime or systemTimer to spread external nav onto aircraftTime and the rest of the nav parameters after matching up the tail end of the flight

- ▶ @rayg: the nav data for today is online in the asp-archive: https://asp-archive.arc.nasa.gov/N809NA/FY2017/2017-04-11/

- ▼ reading IWG1 using pandas

  - ```
    import pandas as pd

    kts2mps = lambda x: 0.514444 * float(x) if x!='' else np.NAN
    ft2m = lambda x: 0.3048 * float(x) if x!='' else np.NAN
    def iso2aircraftTime(s):
    ```

```
        if s=='':
            return np.NAN
        dt = datetime.strptime(s, '%Y-%m-%dT%H:%M:%S.%f')
        midnight = datetime(dt.year, dt.month, dt.day)
        hours = (dt - midnight).total_seconds() / 3600.0
        return hours

    fn_iwg1 = '/data/rdr/shis/dock/sh170411/IWG1.csv'
    columns = [ "iwg", "aircraftTime", "instrumentLatitude", "instrumentLongitude", "_",
               "Altitude", "barometricAltitude", "radarAltitude", "groundSpeed", "_",
               "aircraftIndicatedAirSpeed", "aircraftMachNumber", "aircraftVerticalSpeed",
               "Heading", "aircraftTrackAngle", "driftAngle",
               "Pitch", "Roll", "aircraftSideSlip", "aircraftAttackAngle",
               "staticAirTemp", "dewpointTemp", "totalAirTemp", "aircraftExtPressure", "_", "_",
               "windSpeed", "windDirection", "_", "solarZenithAngle",
               "sunAircraftElevation", "sunGroundAzimuth", "sunAircraftNoseAzimuth" ]
    converters = {'barometricAltitude': ft2m, 'radarAltitude': ft2m, 'aircraftIndicatedAirSpeed': kts2mps,
    'aircraftTime': iso2aircraftTime}

    df = pd.read_csv(fn_iwg1, names=columns, header=None, converters=converters)
```

▼ Pseudocode / theory

  ▼ givens

    • systemTimer is steady, accurate, and monotonic (no instrument restarts)

    • aircraftTime is accurate within 1s (due to SHIS having 0.51s cycle and AC nav 1.0s cycle)

  • for good aircraft records, find linear relation between aircraftTime and systemTimer

  • assign expectedSystemTimer values to IWG1 nav contents based on IWG1 time

  • interpolate aircraft data to actual systemTimer times

  • replace nav arrays in stitched output

  • re-run footprint generator

  • re-run post-stitching pipeline

▼ http://dreadnaught.ssec.wisc.edu/~rayg/shis/sh170411/read_iwg1.html

  • ▢ read_iwg1_1

• for fn in new-nav/*; do bn=$(basename $fn); rm $bn; ln -s $fn $bn; test -f $bn || echo problem with $bn; done

• # correct aircraftType
  fn_type = os.path.join(W.path, 'aircraftType.real4')
  aircraftType = np.memmap(fn_type, mode='r+', dtype=np.float32)
  aircraftType[:] = 871
  aircraftType.flush()

▼ /home/rayg/Repos/svn/SHIS/dev/Sol/shis_footprint

  • [dreadnaught:rdr20170411T161346end20170411T220220sdr20170411T224336]: /home/rayg/Repos/svn/SHIS/dev/Sol/shis_footprint
    info: sceneMirrorAngle adjustment is 0
    info: loading instrumentLatitude.real4
    info: loading instrumentLongitude.real4
    info: loading aircraftType.real4
    info: loading Altitude.real4
    info: loading Heading.real4
    info: loading Pitch.real4
    info: loading Roll.real4
    info: loading sceneMirrorAngle.real4
    info: processing 34747 records
    info: writing Latitude, Longitude, FOVangle files..

▼ finish generation

  • find . -type f -name '*.info' -newer _rad2bt.info |xargs rm
    rm _rad2bt.info
    rm *.mlt
    find dist -type l |xargs rm
    rm -fr pcfilter pcfilter_earth pcfilter_zenith
    make sdr
```