IOP office hours

Integrative Observational Platforms group (IOP)

iopsg@uw.edu

April 2, 2026

UNIVERSITY of WASHINGTON
Applied Physics Laboratory

What we are going cover/who asked us questions

1. "How often should I adjust flight model coefficients to achieve best quality real-time T&S data?"
2. "How do I re-process data locally? Can I re-process old missions using the new basestation3 flight model?"
3. “Basically, I'm interested in an FMS-for-dummies explainer with SG679 as an example case.”
4. What do I configure in sg_calib_constants.m for an RBR Tridente and RBR CODATodo?
5. “What are the avantages/differences between **$CALL_WAIT** and **$T_RSLEEP** ?”

When to adjust flight model coefficients **$HD_A**, **$HD_B** and **$HD_C** on the glider (as suggested by FMS)?

- Brief review: FMS is attempting to estimate an optimal **$HD_A**, **$HD_B** (**$HD_C** is held as a constant) that are fed into the basestation's hydro model - potentially with different values for each dive - for calculating vehicle forward velocity (which affects Seabird CTD corrections, DAC calcs and other science processing).
- **$HD_A**,**B**,**C** on the glider impact selected pitch and VBD positions for dive/climb*
- Broadly speaking, unless these values on the glider impact flight to the point where the hydro model "thinks" the glider isn't flying (i.e., stalled) or the qc engine "thinks" it's flying too slow, you can leave the glider values alone.
  - The **$HD_A**,**B**,**C** are typically the least likely reason for the above problems - generally its something like getting the value of the mass incorrect that throws the hydro model off.
- You can also update them whenever a new recommendation comes in, and switch back to the old values if the update didn't improve the flight.

* **$HD_A**,**B**,**C** impacts **$NAV_MODE,2** and **3** (affects heading selected at start of dive), and **$IMPLIED_C_VBD** (informative only, unless using **$C_VBD_AUTO_DELTA/$C_VBD_AUTO_MAX**)

When to adjust **$HD_A**, **$HD_B** and **$HD_C** on the glider (as suggested by FMS), continued

- Typically, FMS **$HD_A**,**B** are slow in changing:
  - https://seaglider.pub/plot/webp/eng/605/460/FM_ab_dives?mission=Ellett_Array_16&wrap=page
- Typically, we (IOP) will update glider parameters once or twice after the first 10-20 dives, then just monitor the FM_ab_dives plot to see if there is some larger trend we want to be captured in the glider.

How do I re-process data locally? Can I re-process old missions using the new basestation3 flight model?

- You can re-process any previous mission using the new basestation3 software (I do it *all* the time)
- @Seaglider.pub users: we prefer that you reprocess data on your own machine
- You need to have an installation of basestation3 (linux and mac supported)
- There are two main entry points:
  - **Base.py** - starts with the files transmitted from the seaglider (or can be used reprocess from files off a uSD/SD card, and processes to netCDF files and plots. This runs lots of other things (sending out email, contact running vis instances, etc.) that aren't always needed.
  - **Reprocess.py** - starts with .log and .eng files from a previous conversion process (**Base.py**) and processes to netCDF and plots. Generally, this is the one to use if you have a previous mission.
- Most missions will reprocess without issues.
  - Wetlabs naming convention being the major exception
- You can see the plots using a local instance of vis with **vis.py**

sg679, salinity, volmax and FMS

- Something affected the conductivity data starting at dive 147
  - https://seaglider.pub/plot/div/dv/679/146/ctd?mission=20260205_WHICEAS&wrap=page
  - https://seaglider.pub/plot/div/section/679/296/conductivity_raw_section_000?mission=20260205_WHICEAS&wrap=page
  - https://seaglider.pub/plot/div/section/679/296/salinity_raw_section_000?mission=20260205_WHICEAS&wrap=page
- Salinity and temperature are inputs to estimate of glider buoyancy used as by glider hydrodynamic models implementations. One way to see this is in estimates for volmax:
  - https://seaglider.pub/plot/div/eng/679/163/mission_volmax?mission=20260205_WHICEAS&wrap=page
- And FMS volume estimates
  - https://seaglider.pub/plot/webp/eng/679/293/FM_vbdbias?mission=20260205_WHICEAS&wrap=pag
- Note that the HD_A,HD_B regressions remain largely the same
  - https://seaglider.pub/plot/webp/eng/679/147/FM_ab_dives?mission=20260205_WHICEAS&wrap=page
  - https://seaglider.pub/679?x=dive&z=regressed_HD_A,regressed_HD_B&op=plot&wholemission

note: Volmax = ($MASS/$RHO) - (($C_VBD + bias/$VBD_CNV) - $VBD_MIN) * $VBD_CNV

What do I need to add to sg_calib_constants.m for an RBR Tridente or coda T.ODO?

- Get a recent selftest capture where the sensor(s) are installed and working and check that the instrument metadata was captured.
  - For scicon attached instruments, this is almost always the case (see section beginning with >log meta)
  - For truck attached instruments, this is rarely the case
  - For the Tridente:

    `/opt/basestation/bin/python /usr/local/basestation3/tools/GetTridenteChannels.py <selftest.cap>`
    - Paste the output into the sg_calib_constants.m file
  - For the T.ODO:

    `/opt/basestation/bin/python /usr/local/basestation3/tools/GetCodaMeta.py <selftest.cap>`
    - Paste the output into the sg_calib_constants.m file
  - For an Aanderaa Optode:

    `/opt/basestation/bin/python /usr/local/basestation3/tools/GetOptodeConstants.py <selftest.cap>`
    - Paste the output into the sg_calib_constants.m file
- If the metadata isn't there or the tools fail, consult

  `/usr/local/basestation3/sg000/sg_calib_contants.m`

  for the calibration values you need to supply

What are the avantages/differences between **$CALL_WAIT** and **$T_RSLEEP** ?

- A **Call Cycle** is a a series of up to **$CALL_TRIES** attempts to dial into the basestation and successfully execute a complete **Call Session**.
- The time to wait between the end of one call session and start of the next call in is measured in **$CALL_WAIT** seconds (generally 60 seconds).
- A Call Session is successful if all the files scheduled to be downloaded or uploaded are transferred successfully*
- A glider executes one Call Cycle at the end of each dive. At the end of the call cycle, the glider may enter recovery or continue diving.
- A glider in recovery on the surface will continuously execute one Call Cycle followed by a wait of **$T_RSLEEP** seconds until recovery is exited.
- Generally, we speak of lengthening or shortening **$T_RSLEEP** to affect the time between call for gliders in recovery because that delay will always happen, while **$CALL_WAIT** only happens if the glider needed to make multiple calls in a session.

* xmodem/ymodem are less reliable in detecting transmission errors then raw.