CLIVAR P02 (VANC32) Leg 1 LADCP Cruise Report

A.M. Thurnherr
ant@ldeo.columbia.edu

July 25, 2004

1 Overview

NOTE: This section contains the same information as the LADCP section in the main cruise report.

1.1 LADCP System

Two separate ADCP heads were used as LADCPs during the cruise: a 150kHz RDI broadband instrument (BB150) and a 300 kHz RDI Workhorse (WH300). The BB150 was installed looking downward on all stations. The WH300 was used as an uplooker on stations 1–30 (across the Kuroshio, Sikoku Basin, and the Izu Ogasawara Trench) and as a secondary downlooker during bottom-track tests during LADCP casts 82–87 (section 1.5). On the remaining stations the WH300 was not installed on the rosette but kept as a spare.

Each instrument had its own battery — the BB150 a new oil-filled rechargeable lead-acid battery, while the WH300 used non-rechargeable alkaline battery packs in a pressure housing. Only one set of alkaline batteries was used during the cruise. There was plenty of time for charging the oil-filled battery between stations. The battery was found do be easy to vent and performed flawlessly during the cruise. The fact that it does not require a pressure housing is a big advantage.

While on deck, both ADCPs were connected to a Dell Latitude laptop (model PP01L) running Linux. It handled both communications and data processing. The WH300 was connected to the computer via a long (≈20 m) RS-232 cable, while the BB150 communicated via RS-422 and a 422/232 converter. Both instruments were hooked up to the same Keyspan USA-49W 4-port RS-232-to-USB converter. Communications with the instruments was carried out with software written by A. Thurnherr (bbabble and expect scripts), avoiding the need for a Windows PC. Data downloading was carried out in parallel from both instruments at full nominal speed (115 kbps) without any problems. (The effective download speed of the BB150 is approximately 3 times lower than that of the WH300.)

1.2 Instrument Setup

A command file for the BB150 was provided by Eric Firing and Jules Hummon. The WH300 parameters were set for consistency with the BB150. The only change that was made to the setup provided by the UH group was to increase the bin size from 8 to 10 m (and adjust the number of bins accordingly) from station 8 onward, because of memory limitation on the WH300. The setup was not changed back on later casts when only the BB150 was used. In order to distribute previous-ping interference (PPI) across two depth layers, the BB150 was set up for staggered pings (1.0/1.6 s

intervals). The WH300 was set up to ping once every 0.9 s, except during the bottom-tracking tests (section 1.5). Pinging was not synchronized between the instruments. Single-ping ensembles were used throughout and most of the data were collected in beam coordinates. (Earth coordinates were used for the uplooking WH300 on casts 7–10, while the processing software was being adapted to handle uplooker beam coordinates.) No bottom-tracking pings were used, except during the bottom-tracking test casts 82–87 (section 1.5). The ambiguity velocity of the WH300 was increased from 2.5 to 3 and 4 $m \cdot s^{-1}$ during casts 7–20 in order to diagnose a large-velocity warning, which was later found to be caused by surface contamination. LADCP casts before 7 and after 20 used an ambiguity velocity of 2.5 $m \cdot s^{-1}$.

Based on earlier experiences of A. Thurnherr during the KAOS and Anslope cruises, the blanking distance of the WH300 was set to zero and the first bin was always discarded. On the other hand, the blanking distance of the BB150 was set to 16 m, the same size as the transmit length. Note that this implies that the bin length is shorter than the transmit length — it was found that during a cast the instrument resets the transmit length to the bin length. It is therefore not clear what effect, if any, the long transmit length has.

1.3 Data Processing

The data were processed on the Dell laptop using Martin Visbeck's inversion software version 8b. During the cruise, many modifications to the software were implemented (section 2). In particular, it was found that previous-ping interference from the BB150 had to be filtered and this is not possible with the original software. All LADCP profiles were processed in 20 m bins.

During data processing the LADCP data were merged with shipboard ADCP data (see main cruise report) and with CTD data, the latter provided in a suitable format by the shipboard CTD group. Because of data-acquisition problems of the OS-75 SADCP system, no SADCP data were used for shipboard processing of LADCP casts 10 and 11. Because of large water depths, no bottom-track data are available for LADCP casts 29, 35 and 37.

1.4 Station/Cast Numbering

CTD station numbering was done geographically, rather than sequentially (in time). The advantage of this is that the station numbers increase monotonically from west to east along the section, while the disadvantage is that the stations were not occupied sequentially and that station number 41 is missing, because of a weather-related change in plans. LADCP cast numbers were done sequentially, on the other hand. The full CTD station/cast number was used as the LADCP station name — note that it is the station name, rather than the LADCP cast number, that is used as the plot title in the plots produced by the Lamont software (found in the same directory as the processed data files). The following table gives the association between CTD stations and LADCP cast numbers:

CTD Station	LADCP Cast Number	Notes
999	000	test station
001-013	001-013	identical
014-021	021-014	reversed order
022-040	022-040	identical
042-108	041-107	offset by 1

In addition to the station number, every CTD file has a cast number of 01 or 02 associated with it, depending on whether a prior trace-metal cast was carried out at the same location. With the exception of the following CTD station numbers, all CTD data files have cast number 01: 999, 6, 10,

12, 14, 16, 24, 26, 32, 38, 40, 46, 56, 62, 64, 68, 70, 74, 80, 84, 86, 90, 96, 102, 106, 108. Thus, the CTD files for the test station are named 99902 and while those of the first real station are 00101.

1.5 Bottom-Tracking Tests

If true bottom tracking (using dedicated BT pings) were to be used, less water-track data could be collected. Therefore, the BB150 was not configured for BT pings and bottom-tracked velocities were calculated by post-processing the water-track data. Geometric considerations imply that this method is associated with potential bias when there is significant horizontal package motion over ground (see Anslope II cruise report). In order to test whether such bias occurs in practice, both ADCPs were mounted as downlookers during LADCP casts 81–86 (corresponding to CTD stations 82–87); the standard BB150 configuration was used, while the WH300 was configured to use true BT pings. Additionally, during CTD station 86, the Melville was steaming slowly (at 1–2 knots) during the entire downcast and during the first 200 m of the upcast in order to force horizontal package motion over ground. While this may appear extreme, it should be noted that we drifted less during this test station than earlier on in the cruise when we were occupying stations in the Kuroshio in strong winds (e.g. on station 13).

The BT tests were very useful, as they not only confirmed the presence of post-processed BT biases of order 5cm/s both on stations with and without significant drift, but because they also indicate processing-software problems resulting in strong erroneous shears in the bottom-tracked velocity profiles on stations with strong drift. See section 4.1 for details.

1.6 Data Quality

The following comments apply to the processing that was carried out during the cruise. The data will be reprocessed with all the other CLIVAR data after all the CLIVAR cruises have been carried out.

The quality of LADCP data is difficult to assess. The simplest test consists in checking the consistency between the shear- and inverse solutions and, perhaps, also between the downcast- and upcast-only inverse solutions. When the data-processing software detects large inconsistencies, it empirically increases the magnitude of the formal error estimates. In the case of the CLIVAR P02 data set, removal of previous-ping interference (PPI) led to a decrease in consistency in some of the profiles, in particular near the sea bed. Therefore, the results of two separate processing runs (both with and without data editing) are provided as part of the cruise data. It is recommended that any user of the LADCP data carry out a careful station-by-station visual quality check before interpretation of the data, especially where small-scale features are concerned. On stations 1–30 independent data from two ADCP heads are available and those can be processed separately if desired.

(During the steaming to Hawaii at the end of the cruise a novel data quality assessment was attempted. The data from each cast were processed twice, each time using half of the available ensembles. The resulting profiles were then checked for mutual consistency and for consistency with the formal error bars of the full solution. See section 3 for details.)

As indicated above, there is some uncertainty as to the quality of the bottom-tracking data, in particular for casts where the CTD package drifted more than a few 100 meters while in range of the sea bed. Any bottom-tracking errors are likely to contaminate the resulting LADCP profiles, in particular near the sea bed. The drift velocity of each cast is printed on top of the top panels of processing-figures 13.

In general, the data in the western part of the section are of higher quality than those farther

east. This is most likely primarily due to the fact that the instrument range dropped toward the east, because of reduced number of scatterers in the water column. (The dual-head configuration used early on during the cast further improves the quality of the western data.)

1.7 Files and Directories

The LADCP data should contain the following directories, which contain everything that is needed in order to re-process the LADCP data:

raw raw data, instrument-setup command files, communication logfiles;

CTD CTD time series and profiles used for LADCP processing

SADCP shipboard ADCP data used for LADCP processing

processed processed data files and processing figures

processed_noedit 2nd processing run carried out without data editing

2 Modifications to the Processing Software

The processing software is based on version 8b of the Lamont software, written primarily by Martin Visbeck. In addition to numerous small changes, the following major additions were implemented:

1. New processing-control script with finer grain control regarding the processing steps that are to be re-run, better debugging options, and checkpointing to increase re-processing speed.
2. Implementation of data editing for removing specific bins, masking subsets of the available ensembles, filtering spikes (e.g. due to instrument interference because of asynchronous pinging), removing side-lobe contaminated data, and filtering previous-ping interference.
3. Re-implementation of CTD- and GPS-data reading, resulting in large speedups ($\approx$ factor 6 for CTD- and factor 100 for navigation data).

2.1 New Processing Control Script

In the original software, processing is controlled by the three scripts `laproc` (read all data except SADCP, merge LADCP with CTD data, find surface and sea bed), `presolve` (form super-ensembles, calculate preliminary solution, remove velocity outliers, re-form super-ensembles) and `resolve` (calculate inverse and shear solutions, and diffusivity profile; plot and save results). If only part of the processing has to be repeated the user can do that by calling the appropriate script, as long as the results from the previous steps are still in memory. On the CLIVAR P02 cruise, single-ping ensembles were used, resulting in large data files and, consequently, in long processing times. Because significant experimentation was to be carried out it was important that individual steps could be re-run. Therefore, the processing control was made finer. The processing steps are as follows (step 9 — data editing — is new and described in section 2.2):

1. load LADCP data
2. fix LADCP-data problems
3. load navigation (GPS) data

4. get (load or calculate) bottom-track data
5. load CTD profile
6. load CTD time series
7. merge LADCP with CTD time series and find surface and sea bed
8. apply pitch/roll corrections
9. perform data editing
10. form super ensembles
11. remove velocity outliers
12. re-form super ensembles
13. load SADCP data
14. calculate inverse solution
15. calculate shear solution
16. calculate diffusivity profile
17. plot results
18. save output

Additionally, it is now possible to return to a given profile at a later stage and re-do some of the processing steps with new parameters (or after implementing changes to the software). To this purpose, checkpointing was implemented: after a number of (user-definable) processing steps the entire Matlab workspace is written to the disk. When the profile is re-visited, the appropriate checkpoint file is automatically loaded before the desired processing steps are repeated. Checkpointing requires a large amount of disk space. However, disks are cheap and the checkpoint files do not have to be backed up. When disk space is running out, checkpoint files can be deleted at will, resulting only in increased processing times because processing will re-start from scratch later on. (Note that disk space should be monitored when checkpointing is enabled as it can be problematic if the disk fills up during a lengthy download of a cast.) By default, checkpointing is disabled, but it can be enabled by setting `p.checkpoints` (e.g. to `[1 8]`) in `set_cast_params.m`.

In the original software, debugging can be enabled by setting `p.debug > 0`. When debugging is enabled, processing frequently stops (`keyboard` is called) but there is no clear control over where this happens (some `p.debug` values have specific undocumented meanings). It was therefore decided to add a unified debugging mechanism to the processing control: a MATLAB `keyboard` prompt can be requested before and/or after each processing step.

The new function that implements the processing control is called `process_cast(stn)`, where the mandatory argument is the profile number. `process_cast()` uses two auxiliary Matlab scripts, `begin_processing_step` and `end_processing_step`.) In addition to `stn`, the following optional arguments are recognized:

begin_step selects the entry point at which processing is to resume. This parameter defaults to 1, i.e. by default all processing steps are executed. When a value > 1 is given, `process_cast` backtracks to the last valid checkpoint file prior to `begin_step` and processing resumes there. Therefore, setting `begin_step` has no effect unless valid checkpoint files exist.

stop sets debugging stops. Recognized values are -1, 0, 1, and 2; 0 is the default, which disables debugging stops. Using a value of -1 makes `process_cast` stop (Matlab `keyboard` command) immediately before `begin_step` is carried out. Using a value of 1 requests a stop *after* `begin_step` has been executed, and using a value of 2 requests a stop after every step. `stop` can be re-set during the `keyboard` loop, i.e. it is possible to plot some data before a given step is carried out, executing `stop=1; return`, and plotting data during the resulting stop after the step has been executed.

eval_expr contains a Matlab expression (string) that is evaluated immediately after the checkpoint is loaded. This allows executing the same statement when a batch of profiles are to be re-processed.

In the original software, the user would typically write another wrapper script that sets the per-cruise parameters, calls `laproc`, and then does some post-processing. These steps are now integrated as hooks in `process_cast`, which calls the following Matlab scripts:

`set_cast_params` is called after the parameter defaults are set (`defaults.m`) at the beginning of processing. It is a script, rather than a function, in order to have access to all the `process_cast` variables and parameters (e.g. profile number). Typically, the user would fill the structure arrays `p`, `ps`, and `f` with cruise- and profile-specific values. The same script is called immediately after loading a check point (and before `eval_expr` is evaluated), because check-point loading overwrites all variables in the workspace. It is important to note that `set_cast_params` is usually called twice when processing resumes at a checkpoint.

`post_process_cast` is called after processing has finished. The script is optional — i.e. if it does not exist there will not be an error.

2.2 Data Editing

Eric Firing's software implements several editing routines that remove data that are known (or suspected) to be bad, such as the side-lobe contaminated data from the bottom 15% or so of every profile that is in range of the sea bed. Martin Visbeck's inversion software, on the other hand, does not implement any such predetermined edits but instead relies on statistical outlier detection. I like Eric's approach better, and I've therefore added a new processing step to handle bin masking, side-lobe contamination editing, time-domain spike filtering, and previous-ping interference filtering (described in detail in the following subsections). A potentially important edit step that is still missing is wake editing. Data editing is illustrated with a new processing figure (number 14), which shows the target strength before and after data editing (e.g. Figure 1).

All editing parameters are stored in the `p` structure and have names that begin with `edit_`. The defaults are currently set in `edit_data.m`; this initialization should eventually be folded into `defaults.m` if data editing is to be incorporated into the official software release.

2.2.1 Bin Masking

During the KAOS and Anslope II cruises it was found that it is advantageous to set the blanking distance of RDI Workhorse instruments to zero and to ignore all data from bin 1, rather than using a finite blanking distance and down-weighing the bin-1 velocities as had been done previously. Bin masking is controlled using `p.edit_mask_up_bins` and `p.edit_mask_dn_bins`, both of which default to the empty list. In the example shown in Figure 1 bin 1 from the uplooker is masked.

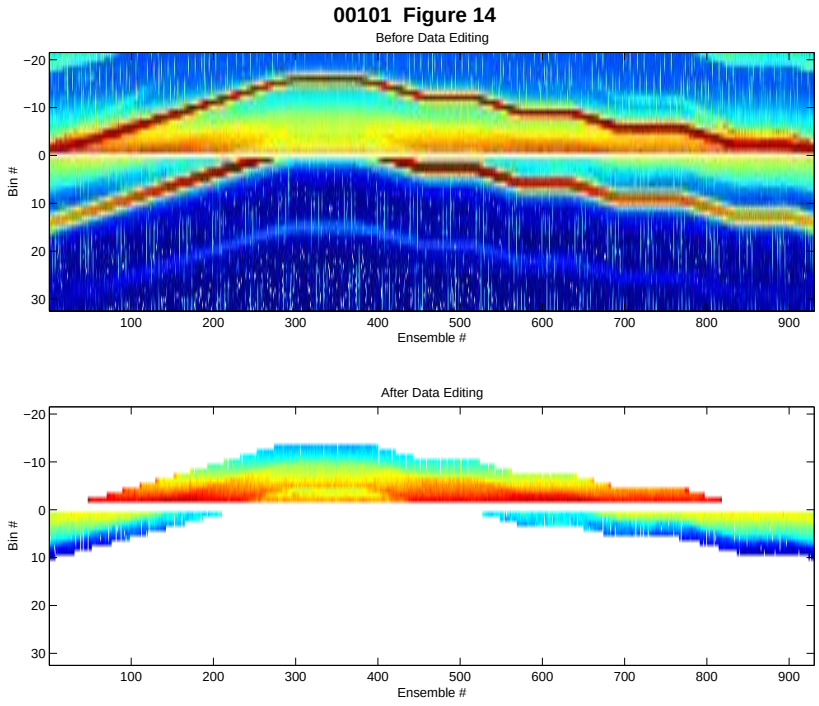

Figure 1: Target-strength before (top) and after (bottom) data editing is carried out; data are removed due to bin masking, side-lobe interference editing, and time-domain spike filtering.

2.3 Side-Lobe Contamination

Side-lobe contaminated velocities near the sea bed (downlooker) and near the sea surface (uplooker) are removed unless `p.edit_sidelobes` is set to false (0). Similar to Eric's software, 1.5 times the bin length is added to the calculated distance from the boundary before the data are removed. In contrast to his programs, where a beam angle of 30° is hard-coded, I use the true beam angle for the calculation. Most of the edited velocities in the example shown in Figure 1 are removed because of side-lobe contamination, which also removes all data below the sea bed and above the sea surface.

2.4 Previous-Ping Interference

The unmodified inversion software does not handle previous-ping interference (PPI) well in all casts as evidenced by the strong shear near 5500 m in the left panel of Figure 2. Note that the downlooker was set up to use staggered pings (1 s/1.6 s), which implies that PPI occurs at different depths in successive pings. A detail of the target strength of a different profile (60) illustrates the PPI signatures in target strength (left panel of Figure 3). The distance of the upper limit of the PPI from the sea bed is given by the product of half the time since the last ping and the sound speed; the center of the PPI is at that distance divided by the cosine of the beam angle; the lower limit is not clear (as the outer sidelobe is unbounded) but for Figure 4 a beam angle of 1.5 times the nominal beam angle was chosen to calculate the lower limit.

PPI editing was implemented by setting the weights of all velocities in the PPI layer to `NaN`, which

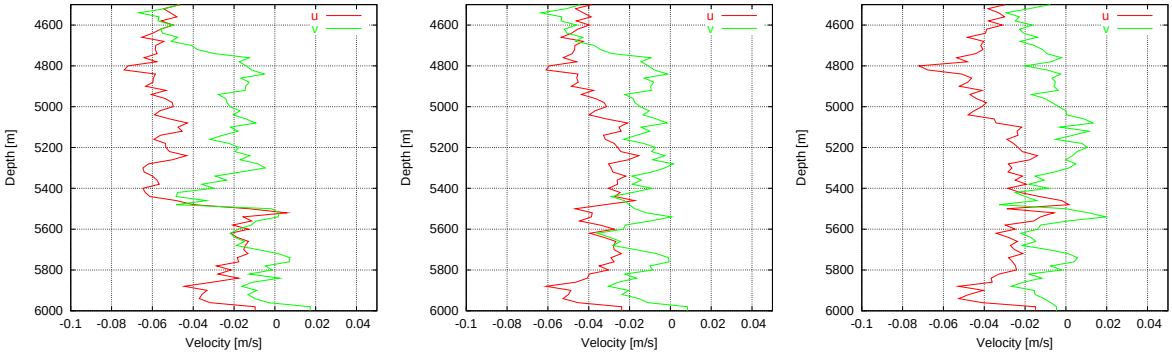

Figure 2: Deep velocities of profile 35 with different methods of PPI editing. Left panel: no editing; center panel: dedicated PPI filtering; right panel: time-domain spike filtering.

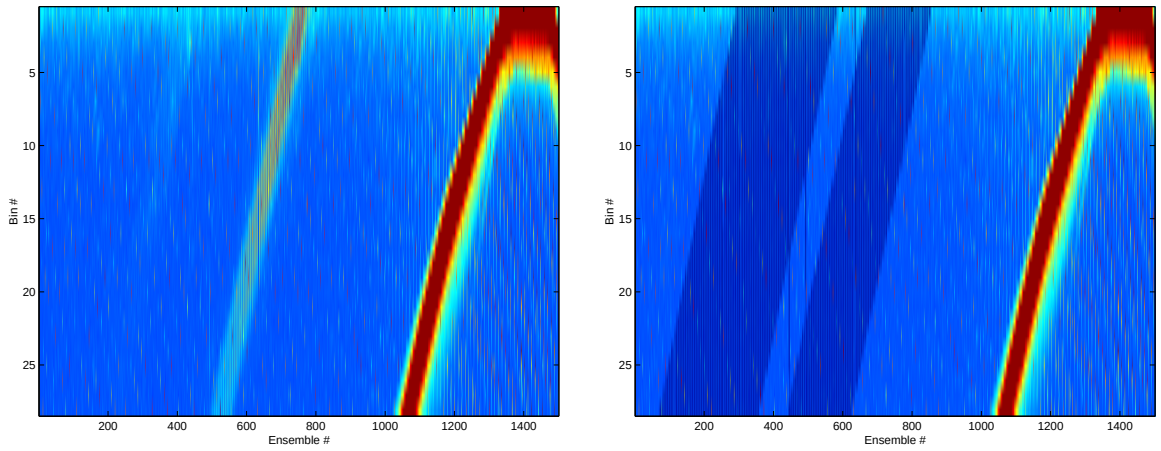

Figure 3: Target strength of profile 60 near the sea bed (left panel) before and (right panel) after PPI editing.

is similar to the approach taken in the Hawaii software. The center panel of Figure 3 shows the target strength after PPI editing with the limits as shown in Figure 4. After some initial experimentation it was found that in many instances too much data are removed by such conservative PPI editing. The results for many casts are similar to not doing any PPI editing at all, i.e. bad shear is introduced at the depth of the PPI peak(s). I suspect that the reason for this behavior is that the super-ensemble depth bins with few velocity values get down-weighed and spurious shear is introduced by inconsistencies between the inversion constraints. An attempt to find a set of parameter values that allow all casts to be processed without either leaving PPI contamination or introducing editing artifacts was not successful.

Therefore, the dedicated PPI filter is disabled by default; in order to enable it, `p.edit_PPI` has to be set to true (1). `p.edit_PPI_layer_thickness` (defaulting to 180 m, the value used in the Hawaii demo) controls the thickness of the layer to be edited and `p.edit_PPI_max_hab` (defaulting to 1000 m) controls the maximum distance from the sea bed where PPI editing is applied.

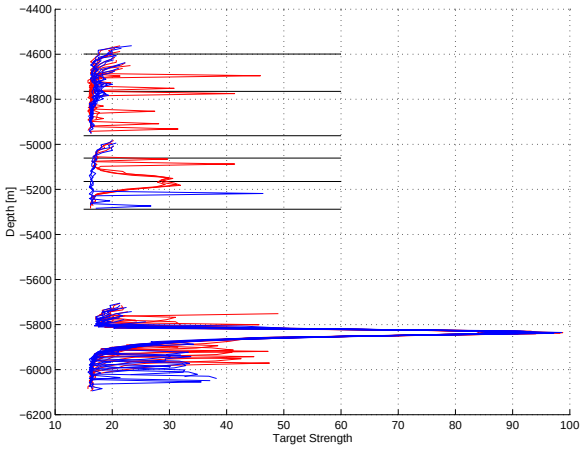

Figure 4: Selected target-strength profiles of LADCP profile 60; even/odd ensembles are shown in red/blue, respectively; the horizontal black lines indicate the PPI limits that were used for the PPI editing shown in Figure 3; the water depth at this station is 5836 m.

2.5 Time-Domain Spike Filtering

During the cruise, multiple acoustic instruments were pinging asynchronously: in addition to the one or two LADCP heads, an acoustic altimeter was mounted on the CTD rosette. At the same time, the ship had two SADCP systems running simultaneously, a narrow band 150 kHz and a new broadband 75 kHz, both from RDI. Additionally, a bathymetric pinger (Knudsen?) was used and, occasionally, the SeaBeam system was left running during casts. It is presumably the cross-talk between the different acoustic instruments that causes most of the speckle apparent in the upper panel of Figure 1. While instrument cross-talk did not appear to have any significant detrimental influence on the quality of the LADCP profiles overall, I nevertheless implemented a simple time-domain spike filter in order to remove the contaminated velocities. The filter is enabled by default but can be disabled explicitly by setting `p.edit_spike_filter` to false (0). The filter removes from each bin all velocities with (sign-changed) second target-strength differences in time that are greater than a limiting value (`edit_spike_filter_max_curv`, defaulting to 2).

To my surprise, the spike filter appears to be handling PPI more robustly than the dedicated PPI filter described in the previous section (right panel of Figure 2). This, of course, crucially depends on staggered pings being used, because in this case the sea bed will appear as spikes in alternate ensembles. In order for the spike filter to effectively remove PPI its limit has to be set quite low, implying that many other velocities associated with (small) spikes in target strength are also removed. This does not appear to degrade the final solutions but this is a subjective assessment.

One, quite annoying, side-effect of PPI editing — regardless of whether dedicated PPI removal or spike filtering are used — is that the different solutions (full, down-/upcast only, shear) can become mutually inconsistent near the sea bed (e.g. Figure 5). In some cases, one or more of the partial solutions can go totally off scale. It is not entirely clear to me why this is, but I suspect it may be because the PPI filtering results in super-ensembles with very few data points and, therefore, low weights.

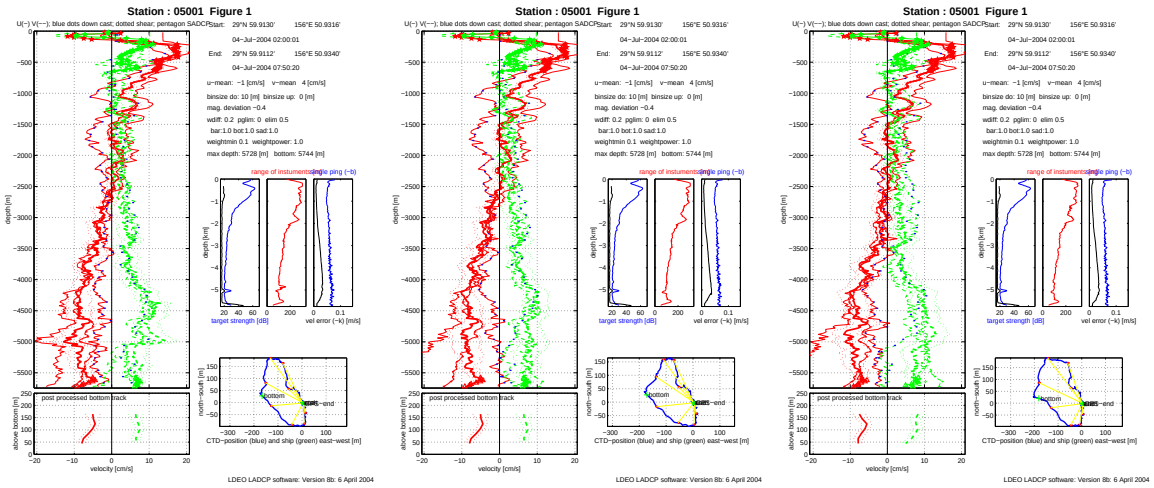

Figure 5: Velocities of LADCP profile 50. Left panel: without PPI editing; center panel: after dedicated PPI editing; right panel: after spike filtering.

2.6 ASCII File Reading

The ASCII data reading portions of `loadctd.m` and `loadnav.m` were largely re-written. They should now be easier to adapt to different file layouts and time bases. Furthermore, they are a lot faster. In case of profile 29, navigation loading time decreased from 527 s to 7 s (not a typo!) and CTD loading time (including lag fitting) decreased from 732 s to 106 s.

Initially, I had not based the cruise-specific file-loading routines on the newest available ones (from Anslope). This led to significant problems because the old ones are not consistent with the newest inversion routines. Therefore, I suggest that the old data loading routines be removed from the software distribution. Furthermore, I think it would be best if, as a policy, all auxiliary data will have to be in ASCII format, in which case the routines can easily be standardized, as I have attempted with `loadctd.m` and `loadnav.m`.

Initially, I got the CTD files with Gregorian times, as preferred by Martin Visbeck. Later, I realized that the times were not calculated with enough accuracy, resulting in apparently uneven time sampling of the CTD time series. Because it would have been more difficult for the CTD group to adapt their time calculation I decided to implement handling of elapsed time, which should be more portable.

2.7 Miscellaneous Changes

A number of additional improvements and bug fixes were implemented to the following routines. All changes are documented in the headers of the corresponding files:

`checkbrk.m` changes were required by the introduction of data editing (section 2.2); edited data points are plotted in green.

`geterr.m` bin numbering in figure 3 was wrong if up-/downlookers had different number of bins; introduced global variable to work around Linux Matlab bug.

`getinv.m` bottom-track weight had been set to zero if no magnetic declination was available; this

restriction seems unnecessary and was removed. A new warning is generated in case all SADCP profiles are removed because of large standard deviation.

`getkzprof.m` bad typo in plot title was corrected.

`lanarrow.m` adapted to changes in `getinv.m`.

`loadctd.m` file-reading was re-written; time offset is now estimated at 90% max depth instead of at max. depth, which may be ill defined during a bottle stop; time-base handling of elapsed time was introduced; time interpolation was introduced for casts with gaps in the CTD time series.

`loadctdprof.m` adapted to CLIVAR P02 data files.

`loadnav.m` re-wrote file reading.

`loadrdi.m` changed large-velocity warning to check only central hour of each cast.

`loadsadcp.m` added plot title (station name) to figure 9; added warning if not SADCP data in time window found.

3 Partial-Data Consistency Check

Assessing the quality of LADCP profiles is difficult. Without any independent data, the only tests that are possible are consistency checks. Consistency is easier to assess in case of the shear solutions calculated with the UH software than in case of the inverse solutions calculated with the LDEO software. This is primarily due to the extra constraints (SADCP, BT, smoothness, low-mode, drag model) that are used for the full inverse solutions, which can sometimes yield reasonable-looking profiles from bad data, e.g. when the instrument range becomes too small because of a lack of scatterers in the water column. One could turn off all those extra constraints and calculate separate inverse solutions from the down- and upcasts, constrained by the GPS data. However, the extra constraints provide additional information that is expected to significantly improve the solutions in many cases.

Toward the end of the P02 cruise, a novel consistency check of the inverse solutions was carried out. All profiles were re-processed twice, each time with half of the available ensembles removed. Because staggered pings were used to avoid PPI (section 1.2) separation into even and odd ensembles would not be a good idea. Instead, processing run "A" was carried out with ensemble numbers 3, 4, 7, 8, 11, 12, ... and processing run "B" with the remainder. The resulting velocity profiles were plotted, together with error bars derived from the full inverse solution (Figure 6). A visual assessment indicates that all but a handful (casts 34 — shown in the right panel of Figure 6 —, 35, 53, 55, 71, 107) of the profiles are consistent within the corresponding error estimates over most of the depth ranges.

The partial-data solutions can also be assessed quantitatively (Figure 7). The median rms velocity discrepancy is $3.6\,\mathrm{cm\cdot s^{-1}}$, consistent with expectations. More interestingly, the data-quality distribution is not random. The high quality of the earliest profiles is most likely at least in part due to the fact that the profiles were collected on comparatively shallow continental-shelf stations. The cause for the low quality of casts 11–20 is probably weather related: both high sea states and large drifts occur in this group. (There are indications that both effects play a role. High sea states lead to large and spiky instrument tilts but high drift velocities alone also appear to be able to degrade the profile quality.) It seems likely that the grouping of later casts into good and bad runs can also be accounted for by weather effects — a more detailed analysis could be carried out.

In summary, it appears that the partial-data consistency check is useful for determining in which part of the water column and to what extent the final velocity profiles should be distrusted. Based on the same idea, it would be possible to determine error estimates using Montecarlo methods. Additionally, it appears that this consistency check can be used to determine which environmental parameters affect the quality of the LADCP profiles. It should be noted, however, that this and similar consistency checks are not expected to be sensitive to errors caused by bad external inversion constraints, such as the bottom-track biases discussed in section 4.1.

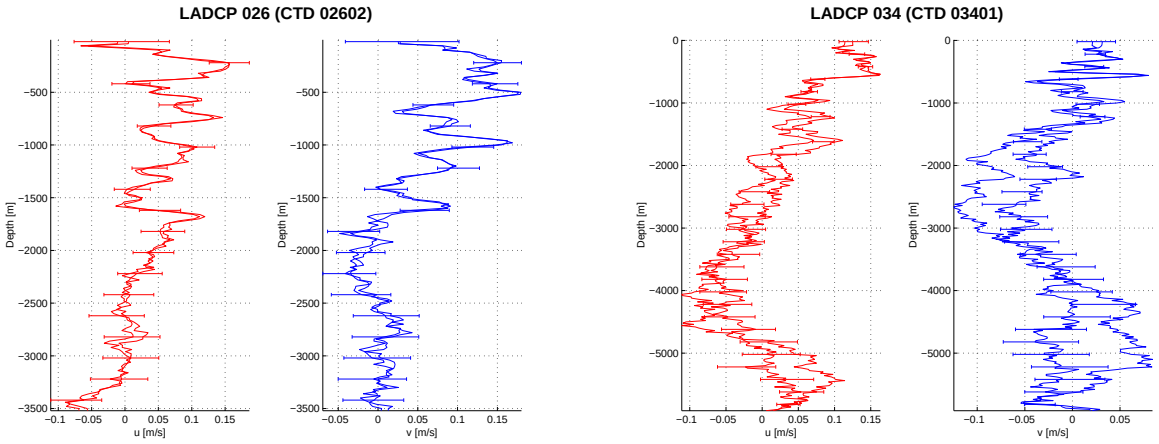

Figure 6: LADCP velocity profiles 26 and 34. Solid lines show solutions calculated with half the available ensembles; error bars show full solutions. The left panel illustrates a high-quality profile; the right panel illustrates the least self-consistent profile of the data set.

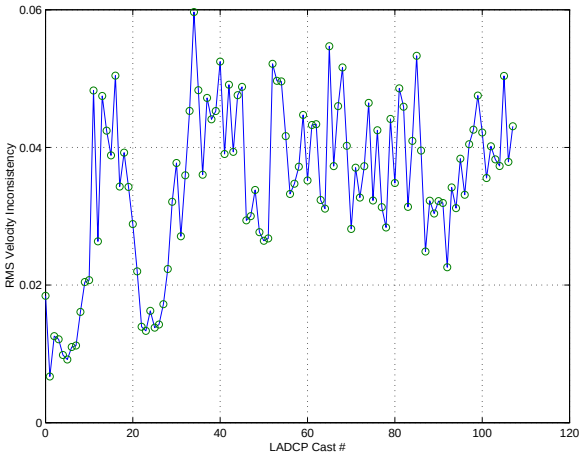

Figure 7: RMS velocity discrepancies between the two solutions, each calculated from half the available ensembles.

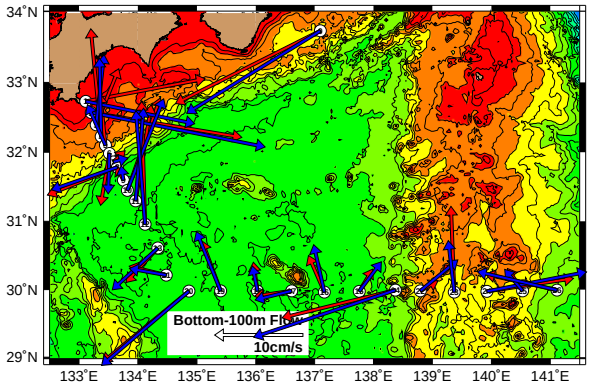

Figure 8: Deep LADCP velocities in the Sikoku Basin with (red) and without (blue) post-processed bottom-track data.

4 Open Problems

4.1 Bad Post-Processed Bottom-Track Data

During the KAOS cruise in 2003 I started having doubts about post-processed bottom-track (BT) data, and these doubts were strengthened during the Anslope cruise (see the corresponding LADCP cruise reports). During Anslope, Martin Visbeck and I discussed bottom-tracking in some detail and we found the potential for side-lobe interference introducing bias into the post-processed BT data. Any potential side-lobe related bias should be particularly strong in casts where the CTD package drifts over ground; furthermore, the bias should be stronger the closer the package is to the sea bed.

Because of these considerations the early casts were processed both with and without BT information. (The range of the instruments was very good during the early casts, which did not extend significantly below 2000 m — therefore, the quality of the solutions without BT data is expected to be good. On later stations the range dropped significantly and most profiles extend to depths below 5000 m, so that it is not clear just how reliable the profiles are without BT constraints.) The general results are encouraging (Figure 8), although there are a few profiles with rather large discrepancies.

One example illustrating bad post-processed BT data is shown in Figure 9, where the CTD drifted with a high velocity near the bottom of the cast ($\approx 0.3\,\mathrm{m\cdot s^{-1}}$ southward, according to the inversion solution). Apart from some apparent inconsistencies near the sea bed the profile is self consistent. However, the shear of the BT velocity profile (lower left panel) is inconsistent with the shear of the inverse- and of the shear solutions and the inversion cannot fully satisfy the BT constraint. When the BT constraint is removed, the inverse solution becomes fully consistent with the shear solution, on the other hand (right panel). All things considered, it appears likely that the meridional BT velocities are bad; the error is of order $10\,\mathrm{cm\cdot s^{-1}}$. Note that the bad BT data do not significantly deteriorate the solution except near the sea bed. Whether the same would be true for deeper casts or for casts with less instrument range is doubtful, however.

The Workhorse 300 kHz instrument used on the cruise had a bottom-tracking option installed. (The BT option was provided without cost by RDI for testing purposes.) In order to assess the quality of the post-processed BT data directly, this instrument was mounted as a secondary downlooker. The resulting BT velocity profiles and drift velocities are shown in Figure 10 and Table 1, respectively. The results are variable but somewhat discouraging overall. While the agreement is good in some cases, there are discrepancies of $5\,\mathrm{cm\cdot s^{-1}}$ or more in others. The biggest consistent biases were

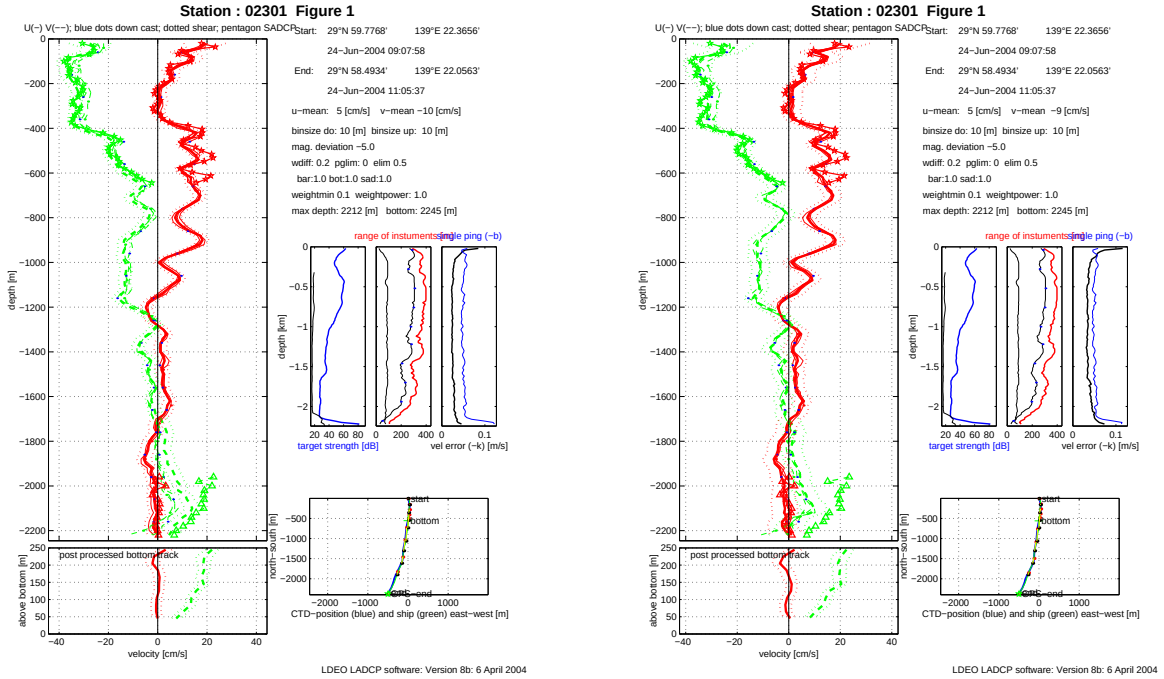

Figure 9: Profile 23 processed with (left) and without (right) bottom-track constraint.

observed on station 86 (bottom center panel), where the package was towed at 1–2 knots during the entire downcast in order to ensure a high package motion over ground. The figure suggests that there may be a depth discrepancy between the two profiles. This is unlikely, however, as the bottom depth was consistently determined (within one meter of each other) by both instruments.

In addition to the biases, there is another problem apparent in the BT velocities from profiles with large drift velocities. In the examples shown in Figures 9 and 10 there is strong BT shear in the components where the drift is strong (meridional in profile 23; meridional and zonal in profile 86). The same observation is true for profile 13, where the largest drift was observed (Figure 11), and also for profile 55. This strongly suggests that the BT velocity profiles are contaminated by side-lobe interference. It is not clear to what extent this problem influences the inverse solutions, as the BT data should directly constrain the CTD velocities over ground, and not the water velocities, i.e. the BT shear should not enter the calculation. Whether this is really the way it is implemented is unclear to me, however.

In summary, the BT velocity profiles that are shown in the bottom panel of processing figure 1 of each station should be treated with suspicion for those stations where the CTD drifted appreciably over ground. Even when the drift was small, it appears that biases in the BT data of up to $5\,\mathrm{cm\cdot s^{-1}}$ are possible. Whether it is generally better to process the data with or without BT constraints is not clear to me at present.

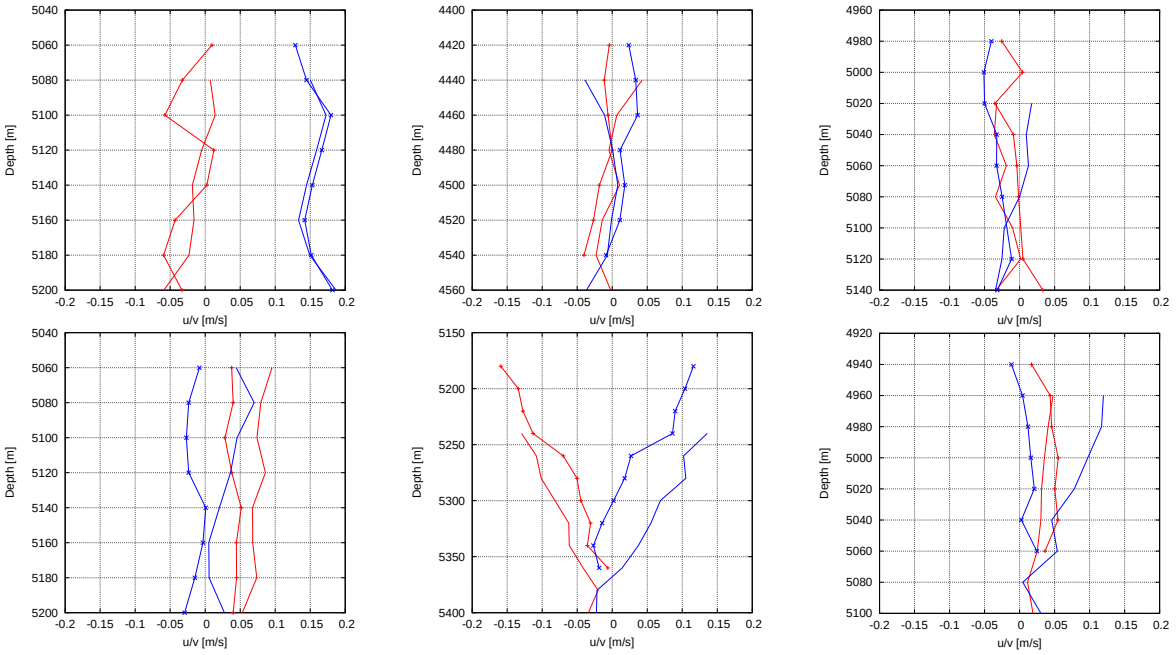

Figure 10: Post-processed (lines without symbols) and RDI BT-mode (lines with symbols) BT data for stations 82–87 (top left to bottom right).

Station	Drift BB150 u	v	Drift WH300 u	v
82	0.00	0.01	0.01	0.01
83	0.05	-0.06	0.05	-0.06
84	0.07	0.00	0.07	0.01
85	0.02	0.01	0.02	0.01
86	0.24	-0.42	0.22	-0.40
87	0.04	-0.02	0.04	-0.02

Table 1: Inversion-derived drift velocities (in $\mathrm{m \cdot s^{-1}}$) near the sea bed during the test casts with two downlookers installed.

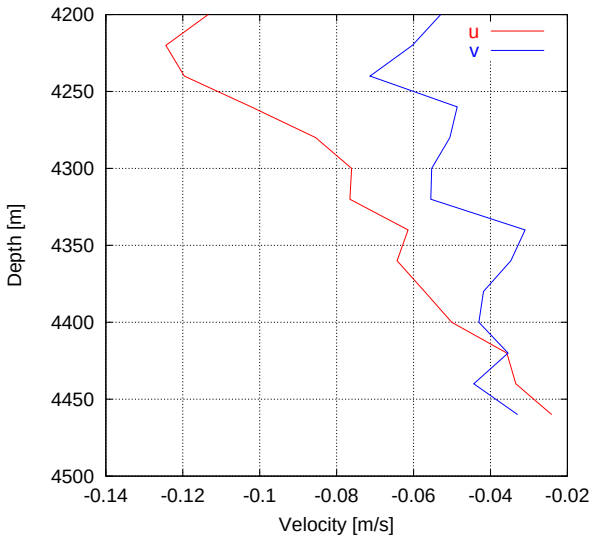

Figure 11: Bottom-track velocity profile 13; zonal and meridional drift velocities were $0.57\,\mathrm{m\cdot s^{-1}}$ and $0.27\,\mathrm{m\cdot s^{-1}}$, respectively.

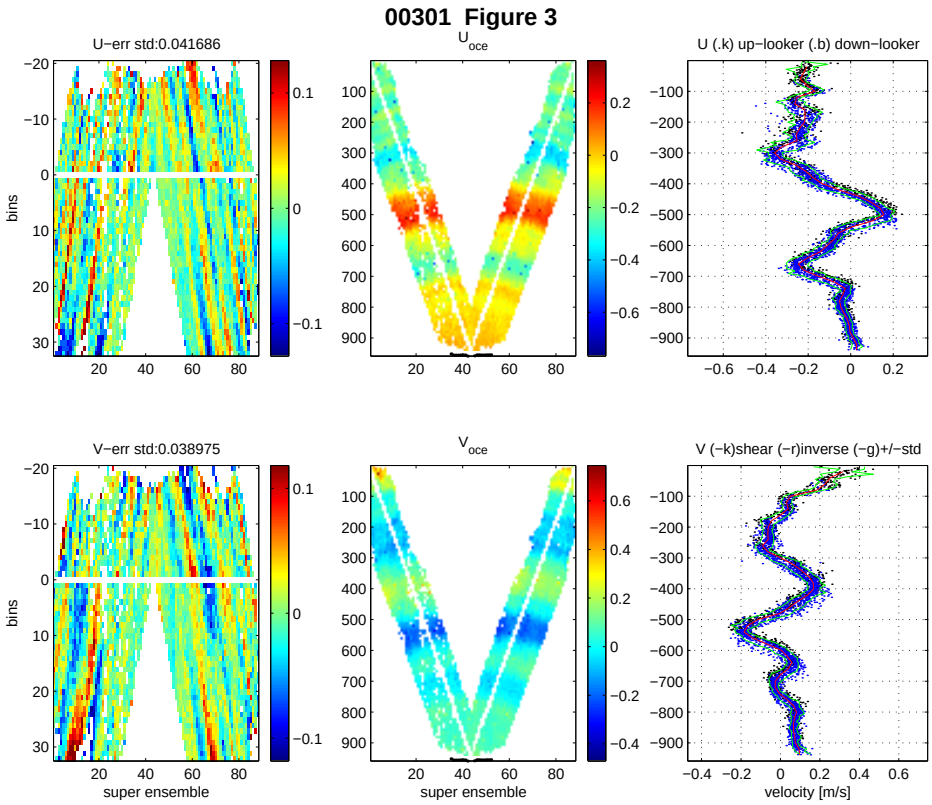

Figure 12: Processing-figure 3 of LADCP cast 3.

4.2 Downlooker Velocity Bin Errors

Early on during the cruise it was noticed that the residual velocities of several casts are associated with strongly banded structure (left panels of Figure 12) and a slanting of the ocean-velocity bands (center panels). Since these two problems always occur together they are almost certainly symptoms of the same problem. In order to exclude possible effects of inconsistent tilt sensors or compasses the data from the two instruments were processed separately, leading to similar residual banding and velocity slanting in the downlooker. (The uplooker data are less clear but if the problem is there, it is much less severe.)

The most likely hypotheses at this stage were i) that something was wrong about the length of the downlooker bins (overestimated bin lengths could account for the velocity slants) or ii) that there was a subtle bug in the separation between ocean- and CTD velocities (the core of the inversion). Figure 13 shows the raw data of cast 3. Note that the velocity slanting is somewhat harder to see in the raw data but that it is nevertheless apparent on close inspection. Note, however, that there are no indications whatsoever for any slanting in the bottom-return in the target-strength sub-panel. Therefore, a processing-related problem can be excluded.

I cannot offer any explanation whatsoever for this problem. Somehow, it appears that the velocities, but not the target strengths, in the far bins of the instrument are reported to be farther away than they really are. The problem is significant enough to be clearly apparent in those casts where

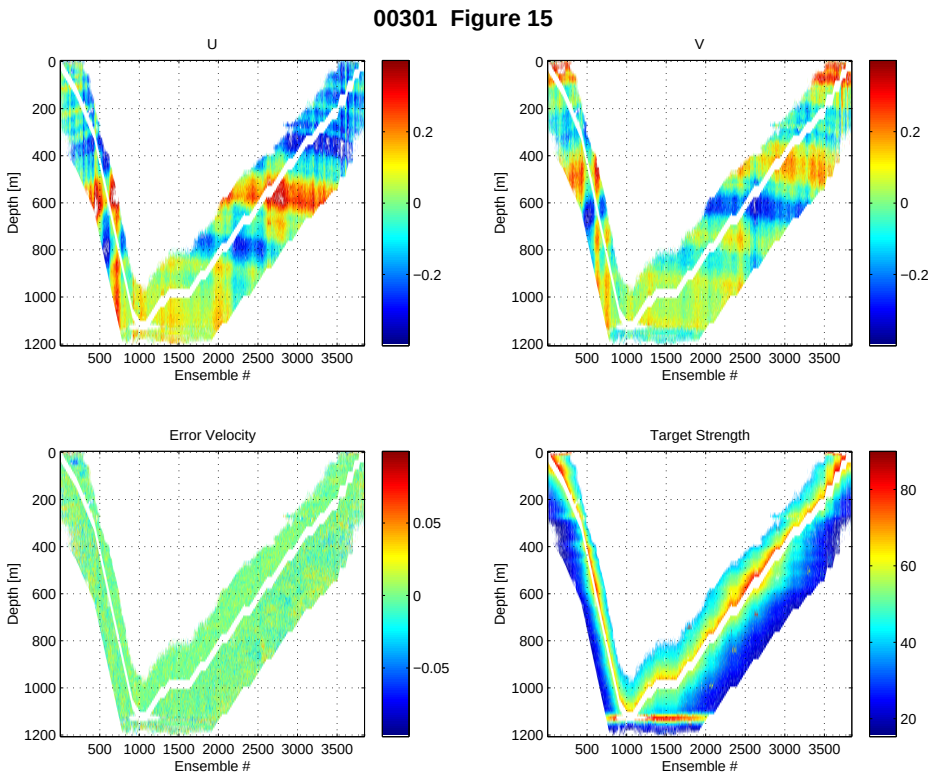

Figure 13: Raw velocities and target strength of LADCP profile 3.

there is strong vertical shear in the water column. On the other hand, the problem does not appear to significantly degrade the overall solutions.