DATA PRODUCT SPECIFICATION FOR MEAN POINT WATER VELOCITY

Version 1-01
Document Control Number 1341-00790
2013-08-29

Consortium for Ocean Leadership
1201 New York Ave NW, 4th Floor, Washington DC 20005
www.OceanLeadership.org

in Cooperation with

University of California, San Diego
University of Washington
Woods Hole Oceanographic Institution
Oregon State University
Scripps Institution of Oceanography
Rutgers University

Document Control Sheet

Version	Date	Description	Author
0-01	2012-05-31	Initial Draft	A.Plueddemann, J. Fredericks
0-02	2012-06-01	Added accuracy requirements in App B.	S. Webster
0-03	2012-06-20	Incorporated comments from focused review.	J. Fredericks
0-04	2012-07-02	Clarified where the L0 VELPTMN product is captured.	S. Webster
0-05	2012-07-06	Added note to clarify the current lack of code to transform transmit pulse length and blanking distance from counts to meters.	S. Webster
0-06	2012-07-18	Minor correction to highlighting. Added parameter names and descriptions.	S. Webster
1-00	2012-07-20	Initial Release	E. Chapman
1-01	2013-08-29	Updates to description of source of magnetic declination estimate in Section 4.3 and test data set in Section 4.6.	S. Pearce

Signature Page

This document has been reviewed and approved for release to Configuration Management.

OOI Senior Systems Engineer: ______________________________

Date: 2012-07-20

This document has been reviewed and meets the needs of the OOI Cyberinfrastructure for the purpose of coding and implementation.

OOI CI Signing Authority: ______________________________

Date: 2012-07-20

Table of Contents

1 Abstract

This document describes the computation used to calculate the OOI Level 1 Mean Point Water Velocity (VELPTMN) core data product, which is calculated using data from Nortek Aquadopp current meter instruments. This document is intended to be used by OOI programmers to construct appropriate processes to create the L1a velocity point mean product.

2 Introduction

2.1 Author Contact Information

Please contact Al Plueddeman (aplueddeman@whoi.edu), Janet Fredericks (jfredericks@whoi.edu) or the Data Product Specification lead (DPS@lists.oceanobservatories.org) for more information concerning the computation and other items in this document.

2.2 Metadata Information

2.2.1 Data Product Name

The OOI Core Data Product Name for this product is

- VELPTMN

The OOI Core Data Product Descriptive Name for this product is

- Mean Point Water Velocity

The VELPTMN data product is comprised of 3 parameters:

Identifier	Name	Description	L0 Units	L1 Units
VELPTMN-VLE	eastward sea water velocity	East component*	mm/s	m/s
VELPTMN-VLN	northward sea water velocity	North component*	mm/s	m/s
VELPTMN-VLU	upward sea water velocity	Up component*	mm/s	m/s

* in earth coordinates relative to North True (magnetic variation accounted for)

2.2.2 Data Product Abstract (for Metadata)

The OOI Level 1a Mean Point Water Velocity core data product characterizes the mean water velocity at a "point" (a volume of several square meters in this case). The mean is created by averaging over time-scales associated with surface gravity waves. The instrument estimates water velocity by measuring the Doppler shift of acoustic signals reflected by particles in the water. The assumption is that the particle is moving with the water and therefore represents the water velocity.

2.2.3 Computation Name

N/A

2.2.4 Computation Abstract (for Metadata)

The OOI Level 1a Mean Point Water Velocity core data product for water velocity is computed by decoding *data ensembles* in binary format from the Aquadopp current meter into velocity in units of m/s, in earth coordinates. The velocity components are adjusted to compensate for the earth's magnetic field effect on the compass heading at the location of the observation.

2.2.5 Instrument-Specific Metadata and Ancillary Data

For each deployment, the following metadata from the Hardware Configuration and the Head Configuration structures (described in Sec 4.2) are required:

Hardware (Instrument) Serial Number, PIC code version number, HW revision, FirmWare version
Recorder Size (in bytes), Velocity Range (Normal or High)
Head frequency (kHz), Head type, Head Serial Number,
Number of beams, Head configuration, Cell Size (always 1.5 meters)

Each velocity record has an associated user configuration record, which includes instrument-specific metadata. Since the instrument may be reconfigured in the field, the metadata must be associated with each observation. These metadata from the User Configuration include:

Average interval (seconds), MeasurementInterval (seconds)
Transmit pulse length** (m), Blanking distance** (m)
Compass update rate (s), Orientation (up or down), Cell Size (fixed at 1.5m)
Sound Speed (measured or assigned in m/s)
Salinity (PPT)—note that this is not measured, it is set by the operator prior to deployment (used for sound speed, if it is computed, as recommended)
DiagnosticInterval (seconds), NsampDiag, NPingsDiag
Measurement load (%)
Software version
Deployment name

** *These parameters are provided in the User Configuration in the units of counts, highlighted in cyan. When it becomes available the code to transform these values from counts to meters will be added to this DPS.*

Along with the user configuration data, ancillary data from the Velocity Data record can be used to assess data quality or provide supporting information. The following data should be available:

Instrument Clock Time
Error code (1 Byte-hex) (See description below.)
Status code (1 Byte-hex)
Battery Voltage (V)
Amplitude (counts)
SoundSpeed (m/s)
Heading, Pitch, Roll (degrees)
Pressure (dbar)
Temperature (degrees C)

Finally, the Aquadopp Diagnostic Data records, if specified in the configuration, should be decoded and available as ancillary data for use in quality assessment. Diagnostic Data have a unique header (see Sec. 4.2) with the rest of the record unpacked according to the Aquadopp Velocity data structure. Diagnostic data include a series of high frequency observations. The first diagnostic data record, if available, can be used to generate a background BeamNoise for each beam, which should be available as time-stamped metadata. The other data in the first diagnostic data record will be meaningless because the transducer is passively recording background noise. The remainder of the diagnostic records, when available, should be stored as an auxiliary data stream, which will look similar to the VELPTMN, but be observed as a high frequency intermittent (perhaps daily) data stream.

2.2.6 Data Product Synonyms

Synonyms for this data product are

- Ocean Current
- Water Velocity

2.2.7 Similar Data Products

Velocity Profile (VELPROF)

Echo Intensity (ECHOINT)
Turbulent Point Water Velocity (VELPTTU)

2.3 Instruments

The Data Processing Flow document (DCN 1342-00790) for Aquadopp current meter instruments describes the flow of velocity data from Aquadopps through all of the relevant QC, data product computations and procedures.

Please see the Instrument Application in the SAF for specifics of instrument locations and platforms.

2.4 Literature and Reference Documents

The electronic file of the reference is stored on Alfresco under REFERENCE>Data Product Specification Artifacts (1341-00790_VELPT).

Nortek AS, “System Integrator Manual”, September 2011
Nortek AS, “Aquadopp Current Meter User Manual”, Document No: N3009-100 Revision C

2.5 Terminology

2.5.1 Definitions

Definitions of general OOI terminology are contained in the Level 2 Reference Module in the OOI requirements database (DOORS). The following terms are defined here for use throughout this document.

Earth (geographic or geodetic) coordinates: Using onboard software, velocity is converted into north, east and up components by using the system compass heading and tilt. The earth coordinates are then referenced to North True by applying a correction for the effect of the magnetic variance on the compass at the specific deployment location at the time of the observation.

Data ensemble: A data ensemble consists of the data collected and averaged during the ensemble interval. A data ensemble can contains a header, date, error code, analog input, input voltage, sound speed, compass heading, tilt, pressure, status code, temperature, velocity, signal strength, and a checksum.

Signal Strength (amplitude): Backscatter Signal Strength is encoded in counts for each beam. This reflects the concentration of backscattering particles in the sample volume and also can be used to assess the quality of the observation.

BCD format: Binary coded decimal is an encoding for decimal numbers in which each digit is represented by its own binary sequence. Four bits are used per digit. The binary sequence is shown in hex.

Measurement Load: The Aquadopp current meter can sample at up to 23Hz. A Measurement Load (ping density) of 13% indicates that sampling is 13% of 23 Hz or 3 pings/second throughout the Average Interval.

2.5.2 Acronyms, Abbreviations and Notations

General OOI acronyms, abbreviations and notations are contained in the Level 2 Reference Module in the OOI requirements database (DOORS).

2.5.3 Variables and Symbols

None

3 Theory

3.1 Description

Aquadopp instruments characterize velocity by measuring the Doppler shift of acoustic signals reflected by particles in the water. The assumption is that the particle is moving with the water and therefore represents the water velocity. An Aquadopp has three acoustic beams, as well as sensors measuring pressure, temperature, compass heading and tilt. Each beam measures a single velocity component parallel to the beam. Velocity components from the three beams are combined using knowledge of beam geometry, along with measured heading, pitch and roll to create east, north and vertical velocity components.

The instrument can be deployed in either upward (e.g. on a multi-function node (MFN)) or downward (e.g. on a near-surface instrument frame (NSIF)) orientation. The internal sensors will detect and compensate appropriately for the instrument orientation.

The output data format of an Aquadopp can be either ASCII or binary. For the OOI DPS 1341-00790, we are describing the binary format product. The L0 mean point water velocity data product output from the Aquadopp driver requires decoding of the binary data and correction for the magnetic field effect on the compass heading to provide the L1a mean point water velocity data product in m/s.

3.2 Mathematical Theory

Not applicable.

3.3 Known Theoretical Limitations

Too few or too many scattering particles can limit the performance of the Aquadopp. The Aquadopp cannot measure currents if the particles are moving perpendicular to the beam.

3.4 Revision History

No revisions to date.

4 Implementation

4.1 Overview

Appendix A includes example code for unpacking Aquadopp hexadecimal data ensembles from an L0 velocity data product to the L1a velocity data product in units of mm/s. The data stream will include transducer head and hardware configuration files, which should remain constant throughout the deployment. Each velocity data record will have an associated user configuration record. The user configuration record may change, since OOI instruments may be reconfigured during deployment to meet the needs of varying objectives and field conditions. Periodically, a diagnostic record will be generated to enable the assessment of instrument performance (e.g. mooring vibration, which can degrade the quality of the velocity observation). The diagnostic data will be carried forward as ancillary data for potential use in assessment of data quality. When present, a diagnostic data record will have an associated header record with metadata relating to how to read the diagnostic data records that will follow the header. Diagnostic records are typically less frequent than velocity records, although the sampling is at a higher frequency (e.g., 1 Hz for 2 minutes once per day).

The instruments shall be configured to always generate data in earth coordinates, which are converted from beam coordinates using the manufacturer's onboard software. The flow direction is reported in "oceanographic convention" for direction – i.e. "direction to", as in "North" is flow towards the north. The compass heading is affected by the earth's magnetic field. As part of the generation of the L1a core product, a rotation on the vector will be applied.

4.2 Inputs

Inputs are:

- L0 VELPTMN velocity data product output from the Aquadopp driver
- Magnetic Variance (See section 4.3 Step 3b below.)

Input Data Formats:

Data records are decoded according to formats according to each Sync ID. The Sync ID is described for each record type in the tables below. The Aquadopp point data stream may contain any of the following types of records:

SYNC ID	STRUCTURE DESCRIPTION
A5 00	User Configuration
A5 01	Aquadopp Velocity Data
A5 04	Head Configuration
A5 05	Hardware Configuration
A5 06	Aquadopp Diagnostics Data Header
A5 80	Aquadopp Diagnostics Data

Aquadopp instruments binary record formats for each record type are described below (Nortek AS *System Integrator Manual* – 2011). Variables that are the L0 VELPTMN data product are highlighted in green. Variables that are directly stored as metadata are highlighted in yellow. Data used for derived metadata are highlighted in cyan.

Head Configuration (224 bytes):

Size	Name	Offset	Description
1	Sync	0	A5 (hex)
1	Id	1	04 (hex)
2	Size	2	Size of structure in words (2 bytes/word)
2	Config	4	Head Configuration: bit 0: Pressure sensor* bit 1: Magnetometer* bit 2: Tilt sensor* bit 3: Tilt sensor mounting (0=up, 1=down)a

2	Frequency	6	Head frequency (kHz)
2	Type	8	Head type
12	SeralNo	10	Head serial number
176	System	22	System Data
22	Spare	198	Spare
2	NBeams	220	Number of Beams
2	Checksum	222	= B58C (hex) + sum of all bytes in structure

* 0=no, 1=yes

Hardware Configuration (48 bytes):

Size	Name	Offset	Description
1	Sync	0	A5 (hex)
1	Format_ID	1	05 (hex)
2	Size	3	Size of structure in words (2 bytes/word)
14	SerialNo	4	Instrument type/serial number
2	Config	18	Board configuration: bit 0: recorder installed (0=no, 1= yes) bit 1: compass installed (0=no, 1=yes)
2	Frequency	20	Board frequency [kHz] acoustic signal
2	PICversion	22	PIC code version number
2	HWrevision	24	Hardware revision
2	RecSize	26	Recorder size (*65536 bytes)
2	Status	28	Bit 0: Velocity Range (0=normal, 1=high)*
12	Spare	30	Spare
4	FWversion	42	Firmware version
2	Checksum	46	= B58C(hex) + sum of all bytes in structure

User Configuration (512 bytes):

Size	Name	Offse t	Description
1	Sync	0	A5 (hex)
1	Id	1	00 (hex)
2	Size	2	Size of structure in words (2 bytes/word)
2	T1	4	Transmit pulse length (counts)
2	T2	6	Blanking distance (counts)
2	T3	8	Receive length (counts)
2	T4	10	Time between pings (counts)
2	T5	12	Time between burst sequences (counts)
2	NPings	14	Number of beam sequences/burst
2	AvgInterval	16	Average interval (seconds)
2	Nbeams	18	Number of beams (3)

2	Tim CtrlReg	20	Timing cotroller mode
2	PwrCtrlReg	22	Power control register
2	A1	24	Not used
2	B0	26	Not used
2	B1	28	Not used
2	CompassUpdRate	30	Compass update rate (seconds)
2	RefSys	32	Coordinate system (0=ENU)
2	NBins	34	Number of cells (1)
2	BinLength	36	Cell size (1.5 m for Aquadopp)
2	MeasInt	38	Measurement Interval (seconds)
6	DepName	40	DeploymentName
2	WrapMode	46	(0=NoWrap,1=wrap when full)
6	clockDeploy	48	Deployment start time
4	DiagInterval	54	Number of seconds between diagnostic measurements
2	Mode	58	*Mode (see below)
2	AdjSounSpeed	60	User input sound speed adjustment factor
2	NsampDiag	62	# samples in diagnotics mode
2	NbeamsCellDiag	64	#beams/cell no. to measure in diagnostics mode
2	NpingsDiag	66	# pings in diagnostics/wave mode
2	ModeTest	68	N/A
2	AnalnAddr	70	Analog input address
2	SWVersion	72	Software Version
2	Spare	74	Spare
180	VelAdjTable	76	Velocity adjustment table
180	Comments	256	File comments
2	Mode	436	Wave measurement mode
2	DynPercPos	438	% for wave cell positioning (N/A)
2	WT1	440	Wave transmit pulse
2	WT2	442	Fixed wave blanking distance (counts) N/A)
2	WT3	444	Wave measurement cell size (N/A)
2	Nsamp	446	Number of diagnostics/wave samples
2	A1	448	Not used
2	B0	450	Not used
2	B1	452	Not used
2	Spare	454	Spare
2	AnaOutScale	456	Analog output scale factor (16384=1.0, max=4.0)
2	CorrThresh	458	Correlation threshold for resolving ambiguities
2	Spare	460	Spare

2	TiLag2	462	Transmit pulse length (counts) second lag
30	Spare	464	Spare
16	QualConst	494	Stage match filter constants (EZQ)
2	Checksum	510	= B58C (hex) + sum of all bytes in structure

***Mode**	58	mode: **bit 0:** use user specified sound speed (0=no, 1=yes) **bit 1:** diagnostics/wave mode 0=disable, 1=enable) **bit 2:** analog output mode (0=disable, 1=enable) **bit 3:** output format (0=Vector, 1=ADV) **bit 4:** scaling (0=1 mm, 1=0.1 mm) **bit 5:** serial output (0=disable, 1=enable) **bit 6:** reserved EasyQ **bit 7:** stage (0=disable, 1=enable) **bit 8:** output power for analog input (0=disable, 1=enable)

AQUADOPP DIAGNOSTICS DATA HEADER (36 bytes):

Size	Name	Offset	Description
1	Sync	0	A5 (hex)
1	Id	1	06 (hex)
2	Size	2	Size of structure in words (2 bytes/word)
2	Records	4	Number of diagnostics samples to follow
2	Cell	6	Cell number of stored diagnostics data
1	Noise1	8	Noise amplitude beam 1 (counts)
1	Noise2	9	Noise amplitude beam 2 (counts)
1	Noise3	10	Noise amplitude beam 3 (counts)
1	Noise4	11	N/A
2	ProcMagn1	12	Processing magnitude beam 1
2	ProcMagn2	14	Processing magnitude beam 1
2	ProcMagn3	16	Processing magnitude beam 1
2	ProcMagn4	18	N/A
2	Distance1	20	Distance beam 1
2	Distance2	22	Distance beam 2
2	Distance3	24	Distance beam 3
2	Distance4	26	N/A
6	Spare	28	Spare
2	Checksum	34	= B58C (hex) + sum of all bytes in structure

AQUADOPP VELOCITY DATA (42 bytes)
Aquadopp Diagnostics Data
Same as Aquadopp Velocity Data, except Id = 0x80.

Size	Name	Offset	Description
1	Sync	0	A5 (hex)
1	Id	1	01 (hex)
2	Size	2	Size of structure in number of words (2 bytes/word)
1	Minute	4	Minute (BCD)
1	Second	5	Second (BCD)
1	Day	6	Day (BCD)
1	Hour	7	Hour (BCD)
1	Year	8	Year (BCD)
1	Month	9	Month (BCD)
2	Error	10	Error Code (see below)
2	Analn1	12	Analog Input 1
2	Power	14	Battery voltage (0.1 V)
2	SoundSpeed	16	Speed of sound (0.1 m/s)
2	Heading	18	Compass heading (0.1 Degrees)
2	Pitch	20	Pitch (0.1 Degrees)
2	Roll	22	Roll (0.1 Degrees)
1	PressMSB	24	Pressure MSB *
1	Status	25	Status Code (see below)
2	PressLSB	26	Pressure LSW*
2	Temperature	28	Temperature (0.01 DegC)
2	VelEastward	30	Eastward flow (mm/s)
2	VelNorthward	32	Northward flow (mm/s)
2	Upward	34	Upward flow (mm/s)
1	AmpB1	36	Beam1-amp (counts)
1	AmpB2	37	Beam2-amp (counts)
1	AmpB3	38	Beam3-amp (counts)
1	Fill	39	Fill byte
2	Checksum	40	B58C(hex) + sum of all bytes in structure

*Pressure (0.001 dbar) = 65536*PressMSB + PressLSB

AQUADOPP ERROR*:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Coord Transf	CT Sensor	Beam Number	Flash	Tag bit	Sensor Data	Measurement data	Compass

0 = ok, 1= error

AQUADOPP STATUS:

Bit 7-6	**Bit 5-4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Power level	WakeUp state	Roll	Pitch	Scaling	Orientation
00= high 3=low (0->3 =6dB)	00=bad 01=power applied 10=break 11=RTC alarm	0=ok 1=out of range	0=ok 1=out of range	0=mm/s 1=0.1mm/s	0=up 1=down

4.3 Processing Flow

All data record types have a specific and unique ID number, as noted above. Each record has an associated CHECKSUM. To validate a record, insuring that you have a complete record, a CHECKSUM test should be conducted. Calculate the "checksum" by summing the total number of bytes in the ensemble excluding the 2-bytes with the checksum and add to the value 0x5B8C (hex). Then read the 2-byte checksum at the end of the ensemble and compare it to the calculated value. If they match, then the data ensemble is valid. If they do not match, search for the next header ID number occurrence.

The term "decoded" below indicates that all units should be adjusted according to the recommended specifications.

The processing flow for the velocity point mean from the Aquadopp is as follows:

Step 1:
Locate, validate and decode the hardware and head configuration records, when available, to confirm that the deployment has not changed. If it has, then the header and hardware configuration metadata need to be time-stamped and archived.

Step 2:
The user configuration record should be located, validated and decoded. Associated metadata (Section 2.2.5) should be archived and made available.

The RefSys should always be zero (NEU). If the reference system is non-zero, then an error should alert a user that the velocity vectors are not in earth coordinates. Additionally, an operations manager should be notified.

Velocity range (scaling bit from the Status word) should always be zero. If the scaling bit is non-zero, the velocity conversion to engineering units will be in error. An operations manager should be notified if the Status Velocity Range bit is set.

It is recommended that the sound speed should be calculated, rather than fixed. If the User Configuration Mode indicates that the sound speed is fixed (bit 0 = 1), an operations manager should be alerted.

User Configuration metadata will not change frequently, so can be stored with a timestamp to be available for corresponding data records. E.g., a user request data from time 1 – time 2, the user configuration(s) for that time need to be available.

Step 3:
The velocity data record should be located, validated and decoded. All parameters should be converted to appropriate units, as specified in Section 2.2.5. The parameters VelEastward, VelNorthward, Upward (highlighted in green) should be saved as the L0 VELPTMN data product (units are mm/s).

3a) Health Checks: Instrument Range checks should be conducted from the manufacturer's specification. The temperature should not be below -4 degC or greater than 40 degC. There should be no velocity component with an absolute velocity greater than 5 meters/second. Also, if the Roll or Pitch bits are set in the STATUS byte, data may be suspect. If the Orientation bit differs from the expected orientation from the Head Configuration, a problem may exist. If the Or if the Scaling bit is not zero, there is a problem. If any of these conditions occur, an instrument health check flag (IH_qcflag) should be set. IH_qcflag should be a structure such as:

bit 0: Orientation (Masked so it's zero if it matches the Head defined orientation)
bit 1: Scaling (should be zero)
bit 2: Pitch (should be zero)
bit 3: Roll (should be zero)
bit 4: TempQC (set to 1 if range check fails)
bit 5: VelocityQC (set to 1 if velocity range check fails)

The Error Byte can be left as is and available as metadata.

3b) The magnetic variation correction must be applied to the velocity components. The magnetic declination estimate for each deployment will be calculated from the World Magnetic Model (WMM) using Latitude, Longitude, and the data time stamp. The WMM is implemented directly in the Ocean Observatories Initiative Network (OOIN) for automated calculation of magnetic declination. For more information on the WMM, go to http://ngdc.noaa.gov/geomag/WMM/DoDWMM.shtml.
Positive magnetic variation means that magnetic north is east of true north and negative one means that magnetic north is west of true north. Inputs are horizontal velocity profiles in earth coordinates, eastward flow and northward flow relative to magnetic north, and the magnetic variation. Example code is provided in Appendix D

Archive and publish the rotated core L1 product, mean velocity in m/s.

3c) Archive and make available all associated auxiliary data (Section 2.2.5).

Step 4:
Locate and validate the diagnostics data header, when available. This will be used to interpret the diagnostic data.

Step 5:
Locate, validate and decode the diagnostics data, when available. Diagnostic data can be processed identically to the data record described in Step 3, above, but it should be treated as ancillary data for quality assessment. For the first A5-80 diagnostic data record the sensors passively listen. This provides a background acoustic noise level that can be used in the computation of a signal to noise ratio (SNR). Therefore, a time-stamped value for BeamNoise (counts) for each of the three beams should be part of the auxiliary diagnostic data. The rest of the first record can be discarded. The following A5-80 records should be treated identically to the Velocity Data Record (Step 3, above).

4.4 Outputs

The outputs of the VELPTMN computation are

- L1 VELPTMN Northward, Eastward and Upward (relative to North True) velocity point mean in m/s

The metadata and ancillary data requirements are listed in Section 2.2.5 and highlighted in the "Inputs" tables of record formatting (Section 4.2).

See Appendix B for a discussion of the accuracy of the output.

4.5 Computational and Numerical Considerations

4.5.1 Numerical Programming Considerations

There are no numerical programming considerations for this computation. No special numerical methods are used.

4.5.2 Computational Requirements

Computation estimates are not required for algorithms that are not computationally intensive.

4.6 Code Verification and Test Data Set

Below is a table of test data to use for testing the algorithm.

			L0 VELPTMN Inputs		
Lat	Lon	Date Time	East Velocity	North Velocity	Upward Velocity
14.685	-51.044	2005-03-11 21:00:00	-3.2	18.2	-1.1
14.685	-51.044	2005-03-11 22:00:00	0.1	9.9	-0.6
14.685	-51.044	2005-03-11 23:00:00	0	12	-1.4
14.685	-51.044	2005-03-12 00:00:00	2.3	6.6	-2.0
14.685	-51.044	2005-03-12 01:00:00	-0.1	7.4	-1.7
14.685	-51.044	2005-03-12 02:00:00	5.6	3.4	-2.0
14.685	-51.044	2005-03-12 03:00:00	5.1	-2.6	1.3
14.685	-51.044	2005-03-12 04:00:00	5.8	0.2	-1.6
14.685	-51.044	2005-03-12 05:00:00	8.8	-1.5	-1.1
14.685	-51.044	2005-03-12 06:00:00	10.3	4.1	-4.5

L1 VELPTMN Outputs		
East Velocity	North Velocity	Upward Velocity
-0.085136	0.164012	-0.011
-0.028752	0.094738	-0.006
-0.036007	0.114471	-0.014
0.002136	0.06986	-0.02
-0.023158	0.07029	-0.017
0.043218	0.049237	-0.02
0.056451	-0.009499	0.013
0.054727	0.019311	-0.016
0.088446	0.012096	-0.011
0.085952	0.070017	-0.045

Appendix A Example Code

This Appendix contains C code for reading raw binary Aquadopp data files. It was taken from the Nortek AS, "System Integrator Manual", September 2011

A-1. Structure Definitions

```
#define PD_MAX_BEAMS 3
#define PD_MAX_BINS 128
#define PD_MAX_STAGECELLS 1024
#pragma pack(push)
#pragma pack(1) // 1 byte struct member alignment used in firmware
////////////////////////////////////////////////////////////////////////
///////
// Clock data (6 bytes) NOTE! BCD format
typedef struct {
unsigned char cMinute; // minute
unsigned char cSecond; // second
unsigned char cDay; // day
unsigned char cHour; // hour
unsigned char cYear; // year
unsigned char cMonth; // month
} PdClock;
////////////////////////////////////////////////////////////////////////
///////
// Aquadopp diagnostics header data
typedef struct {
unsigned char cSync; // sync = 0xa5
unsigned char cId; // identification = 0x06
unsigned short hSize; // total size of structure (words)
unsigned short nRecords; // number of diagnostics samples to follow
unsigned short nCell; // cell number of stored diagnostics data
unsigned char cNoise[4]; // noise amplitude (counts)
PdClock clock; // date and time
unsigned short hSpare1;
unsigned short hDistance[4]; // distance
unsigned short hSpare[3];
short hChecksum; // checksum
} PdDiagHead;
////////////////////////////////////////////////////////////////////////
///////
// Aquadopp velocity data 3 beams
typedef struct {
unsigned char cSync; // sync = 0xa5
unsigned char cId; // identification (0x01=normal, 0x80=diag)
unsigned short hSize; // size of structure (words)
PdClock clock; // date and time
short hError; // error code:
// bit 0: compass (0=ok, 1=error)
// bit 1: measurement data (0=ok, 1=error)
// bit 2: sensor data (0=ok, 1=error)
// bit 3: tag bit (0=ok, 1=error)
// bit 4: flash (0=ok, 1=error)
// bit 5:
// bit 6: serial CT sensor read (0=ok, 1=error)
unsigned short hAnaIn1; // analog input 1
```

```
unsigned short hBattery; // battery voltage (0.1 V)

union {
unsigned short hSoundSpeed; // speed of sound (0.1 m/s)
unsigned short hAnaIn2; // analog input 2
} u;
short hHeading; // compass heading (0.1 deg)
short hPitch; // compass pitch (0.1 deg)
short hRoll; // compass roll (0.1 deg)
unsigned char cPressureMSB; // pressure MSB
char cStatus; // status:
// bit 0: orientation (0=up, 1=down)
// bit 1: scaling (0=mm/s, 1=0.1mm/s)
// bit 2: pitch (0=ok, 1=out of range)
// bit 3: roll (0=ok, 1=out of range)
// bit 4: wakeup state:
//bit 5:(00=bad power,01=break,10=power applied,11=RTC alarm)
// bit 6: power level:
// bit 7: (00=0(high), 01=1, 10=2, 11=3(low))
unsigned short hPressureLSW; // pressure LSW
short hTemperature; // temperature (0.01 deg C)
short hVel[3]; // velocity
unsigned char cAmp[3]; // amplitude
char cFill;
short hChecksum; // checksum
} PdMeas;
```

A-2. Sample code for decoding the Aquadopp data structure

```
typedef struct {
unsigned char cSync; // sync = 0xa5
unsigned char cId; // identification (0x01=normal, 0x80=diag)
unsigned short hSize; // size of structure (words)
PdClock clock; // date and time
short hError; // error code
short hSpare;
unsigned short hBattery; // battery voltage (0.1 V)
unsigned short hSoundSpeed; // speed of sound (0.1 m/s)
short hHeading; // compass heading (0.1 deg)
short hPitch; // compass pitch (0.1 deg)
short hRoll; // compass roll (0.1 deg)
unsigned char cMSB; // pressure MSB
char cStatus; // status code
unsigned short hLSW; // pressure LSW
short hTemperature; // temperature (0.01 deg C)
short hVel[3]; // velocity (mm/s)
unsigned char cAmp[3]; // amplitude (counts)
char cFill;
short hChecksum; // checksum
} PdMeas;
{
.
.
.
```

```
PdMeas meas;
SYSTEMTIME st;
double dVel[3];
double dAmp[3];
short hChecksum;
double dPressure;
double dBattery;
double dHeading;
double dPitch;
double dRoll;
double dTemperature;
// Assuming three beams
// Checksum control
if (meas.hChecksum != Checksum((short *)&meas,meas.hSize - 1)) {

// Handle the error.
}
st = ClockToSystemTime(meas.clock);
dVel[0] = (double)meas.hVel[0] * 0.001;
dVel[1] = (double)meas.hVel[1] * 0.001;
dVel[2] = (double)meas.hVel[2] * 0.001;
dAmp[0] = (double)meas.cAmp[0];
dAmp[1] = (double)meas.cAmp[1];
dAmp[2] = (double)meas.cAmp[2];
dPressure = (65536.0*(double)meas.cMSB + (double)meas.hLSW)*0.001;
dBattery = (double)meas.hBattery * 0.1;
dHeading = (double)meas.hHeading * 0.1;
dPitch = (double)meas.hPitch * 0.1;
dRoll = (double)meas.hRoll * 0.1;
dTemperature = (double)meas.hTemperature * 0.01;
.
.
}//////////////////////////////////////////////////////////////////////
/
// Convert from BCD time to system time

SYSTEMTIME ClockToSystemTime(PdClock clock)
{
SYSTEMTIME systime;
WORD wYear;
wYear = (WORD)BCDToChar(clock.cYear);
if (wYear >= 90) {
wYear += 1900;
}
else {
wYear += 2000;
}
systime.wYear = wYear;
systime.wMonth = (WORD)BCDToChar(clock.cMonth);
systime.wDay = (WORD)BCDToChar(clock.cDay);
systime.wHour = (WORD)BCDToChar(clock.cHour);
systime.wMinute = (WORD)BCDToChar(clock.cMinute);
systime.wSecond = (WORD)BCDToChar(clock.cSecond);
systime.wMilliseconds = 0;
return systime;
}
//////////////////////////////////////////////////////////////////////
```

```
// Convert from BCD to char
unsigned char BCDToChar(unsigned char cBCD)
{
unsigned char c;
cBCD = min(cBCD,0x99);
c = (cBCD & 0x0f);
c += 10 * (cBCD >> 4);
return c;
}
///////////////////////////////////////////////////////////////////////
// Compute checksum
short Checksum(short *phBuff,int n)
{
int i;
short hChecksum = 0xb58c;
for (i=0; i<n; i++)
hChecksum += phBuff[i];
return hChecksum;
}
```

Appendix B Aquadopp System Specifications

The algorithm output accuracy for the OOI L1a VELPTMN core data product calculated by this algorithm is equivalent to the accuracy of the instrument. The manufacturer, Nortek-AS, provides accuracy as stated below (from Aquadopp User Manual):

Velocity Measurement:
- Range: +- 5 m/s
- Accuracy: 1% of measured value +- 0.5 cm/s
- Max sampling rate (output): 1 second
- Internal sampling rate: 23Hz

Temperature (thermistor embedded in head):
- Range: -4 DegC – 40 DegC
- Accuracy/Resolution: 0.1 DegC/0.01DegC
- Time response: 10 minutes

Compass (flux-gate with liquid tilt)
- Maximum tilt: 30 Degrees
- Accuracy/Resolution: 2 Deg/0.1 Deg
- Time response: user defined

Tilt (liquid level)
- Accuracy/Resolution: 0.2 Deg/0.1 Deg
- Up or down: Automatic detect

Pressure (piezoresistive)
- Range: 0-200 m (standard)
- Accuracy: 0.25%
- Resolution: Better than 0.005% of full scale/sample

Relevant requirements from The OOI requirements database (DOORS, L2_Science_Requirements_ReferenceOnly_Baseline_Version_2.24 exported from DOORS SL CCB 2012-05-02) are listed below.

L2-SR-RQ-3274	Single Point Water Velocity shall be measured with a speed accuracy of +/-1% of measured value +/-1 cm/s.
L2-SR-RQ-3736	Single Point Water Velocity shall be measured with a speed resolution of 0.1 cm/s.
L2-SR-RQ-3737	Single Point Water Velocity shall be measured with a single sample speed precision of 3 cm/s.
L2-SR-RQ-3739	Single Point Water Velocity shall be measured with a direction accuracy of +/-2 degrees.
L2-SR-RQ-3740	Single Point Water Velocity shall be measured with a direction resolution of 0.1 degrees.
L2-SR-RQ-3158	Mean Point water velocity at the seafloor shall be measured to a value no less than 200 cm/s.

L4-CG-IP-RQ-248	Single Point Water Velocity instruments shall have a speed accuracy of +/-1% of measured value +/-1 cm s-1.
L4-CG-IP-RQ-250	Single Point Water Velocity instruments shall have a speed resolution of 0.1 cm s-1.
L4-CG-IP-RQ-484	Single Point Water Velocity instruments shall have a single sample speed precision of no greater than 3 cm s-1.
L4-CG-IP-RQ-256	Single Point Water Velocity instruments shall have an absolute direction accuracy of +/- 2 degrees.
L4-CG-IP-RQ-257	Single Point Water Velocity instruments shall have a direction resolution of 0.1 degrees.

Appendix C Sensor Calibration Effects

None.

Appendix D Matlab Example Code

Matlab example code for transformation to correct for magnetic variance:

```
% Apply magnetic variation 'Degrees' (see Section 4.3 Step 3b)
% Note: use of atan2 changes heading range from 0-360 to +/-180.
% Note: example data provided used magvar=-17.6
% east/north are velocities relative to magnetic north
magvar = Degrees * pi/180;    % convert to radians
i = sqrt(-1);
unit_vec = sin(hd*pi/180) + i* cos(hd*pi/180);
u = real(unit_vec)*cos(magvar) + imag(unit_vec)*sin(magvar);
v = imag(unit_vec)*cos(magvar) - real(unit_vec)*sin(magvar);
hd = atan2( u, v ) * 180/pi;
u =    east*cos(magvar) + north*sin(magvar);
v = north*cos(magvar) -    east*sin(magvar);
east_rotated = u;    north_rotated = v;
magvar = magvar * 180/pi;    % save value in degrees
```