Development of Radar Altimetry Data Processing in the Oceanic Coastal Zone

ESA/ESRIN Contract No. 21201/08/I-LG

EWP3 and EWP5 – Deliverables D3.4 and D5.3

Implementation of the Bayes Linear Retracker (D3.4) and validation on real waveforms (D5.3)

VERSION 1.0, 28 May 2012

National Oceanography Centre
European Way, Southampton
SO14 3ZH, United Kingdom
Tel: +44 (0)23 80596404 Fax: +44 (0)23 80596400

Code COASTALT2-D34D53 **Edition** 1.0 **Date** 28 May 2012

Client European Space Agency **Final User** -

	Name	Signature	Date
Written by	Luke West Paolo Cipollini Peter Challenor		28 May 2012
Revised by	Paolo Cipollini		28 May 2012

DISTRIBUTION	Affiliation
Jérôme Benveniste, Salvatore Dinardo, Bruno Lucas	ESA
Paolo Cipollini, Peter Challenor, Helen Snaith, Luke West, Phil Woodworth,	NOC
Stefano Vignudelli	CNR
Jesus Gomez-Enri	U Cadiz
Cristina Martin	STARLAB
Joana Fernandes, Alexandra Nunes	U Porto
Susana Barbosa	U Lisbon

Revision History

Issue	Date	Change
1	28 May 2012	Initial Release

Table of Contents

Reference Documents

[D35] **Report on Innovative Techniques**, COASTALT Deliverable 3.5, v. 3.0, 16 Nov 2009.
[C2D12b] **COASTALT Processor: Plug&Play user guide**, COASTALT2 Deliverable 1.2b, COASTALT2-D12b-10 v 1.1, 28/05/2012
[C2D32] **Bayes Linear Retracking of Radar Altimeter Data**, COASTALT2 Deliverable 3.2, COASTALT2-D32-11 v 1.1, 28/05/2012.

1 Introduction

The present document describes the C++ implementation of the Bayes Linear Retracker developed in WP3 of COASTALT, and the attempts to plug it in the COASTALT processor to perform validation over real waveform examples.

Work carried out in WP3 (Retracking) during Phase 1 of COASTALT (see [D35]) had already identified the Bayes Linear Technique as one of the most promising to carry out sequential retracking of waveforms. Therefore it was decided to dedicate considerable efforts to try and develop this technique in Phase 2 of COASTALT, to the point where it would be possible to plug it in the COASTALT processor. The theory was fully developed and is described in [C2D32] alongside the retracker implementation in R language. Unfortunately, and despite significant efforts, we have not been successful in overcoming the numerical problems in the implementation, both in R and in C++, as it is discussed in the conclusion section of this document.

2 Converting the Bayes Linear Retracker to C++

The Bayes Linear retracker described in [C2D32] was implemented in R for a single iteration. The relevant code is available on the COASTALT FTP site under

```
/utilities/bayes_linear_retracker/alttracker_R
```

The heart of the Bayes-Linear R code, i.e. file `bayes.linear.tracker.R` is copied in Appendix A for convenience.

The R implementation was recoded in C++, partly for performance reasons, but also with the intention that this would be an intermediate step towards pluggin it into the COASTALT Processor using the plug-and-play capability [C2D12b], and to allow more efficient debugging. The C++ full implementation code is available on the COASTALT FTP site under

```
/utilities/bayes_linear_retracker/cpp
```

The heart of the C++ code, i.e. an excerpt from file `app.h` is copied in Appendix B for convenience.

Once the C++ code was debugged to the point it could reproduce the results of the R code, it was 'chained' – that is, put into a loop such that the posterior,

$$\{E(\theta \mid w), V(\theta \mid w)\}$$

i.e. the expectation and variance of θ, the vector of parameters (epoch t_0, significant wave height h_s, backscatter σ^0 and thermal noise t_n) affecting the form of the waveform w, would become a new prior

$$\{E(\theta), V(\theta)\}$$

which could be used to generate a subsequent posterior in accordance with the Bayes-Linear method. This was tried over a test dataset of 100 sequential Envisat RA-2 Ku-band 20-Hz waveforms. The test dataset is shown in figure 1 and is available on the COASTALT FTP site in handy ASCII format:

```
/utilities/bayes_linear_retracker/cpp/app/ascii_data_ku.dat
```

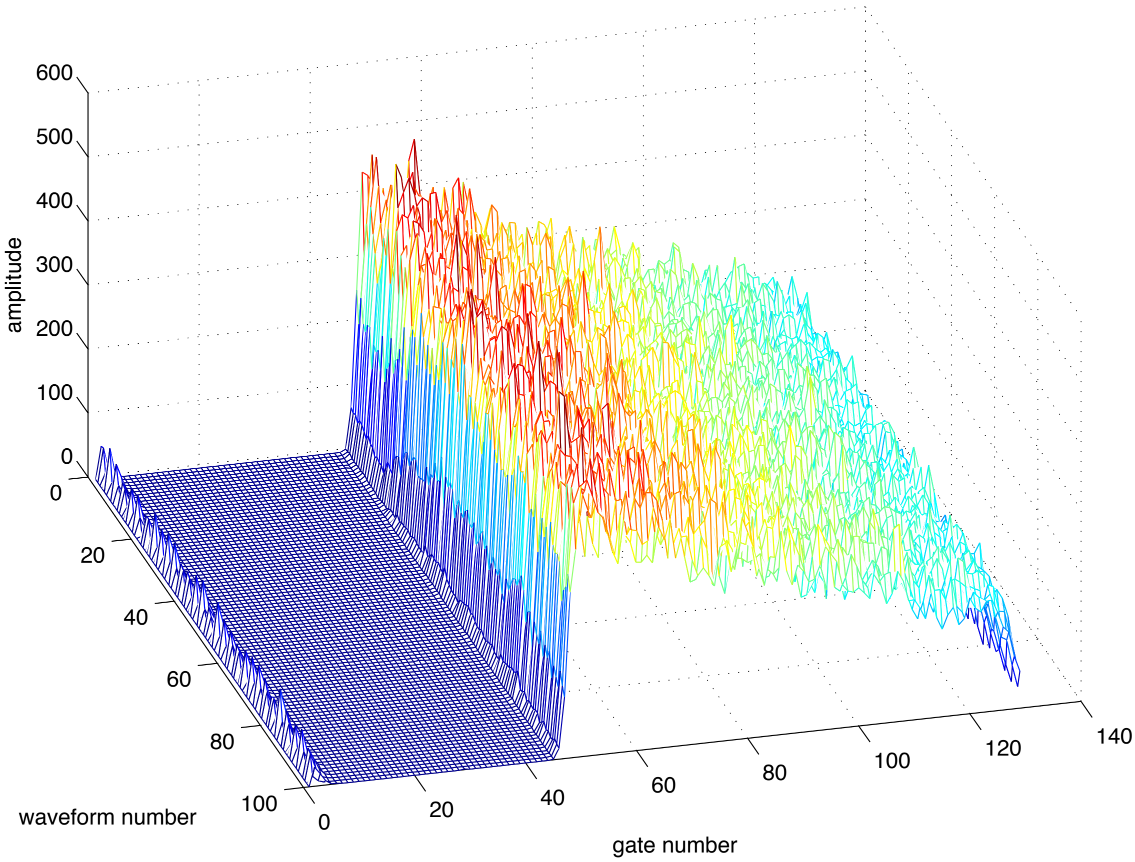

Figure 1. Test dataset of 100 open-ocean RA-2 Ku-band waveforms

In testing over this dataset, we found that the implementation of the retracker suffers from significant numerical problems, which we were not able to overcome despite trying some workarounds. Firstly, for very small values of w, $V(w)$ is often not invertible, or gives rise to parameters which imply a negative value for the thermal noise t_n. This was rectified by making sure the elements of `CovBD` (i.e. the covariance matrix between the prior estimates of our parameters and the measured waveform) were never negative by clamping them to zero, but although this rendered $V(w)$ invertible, the solution did not converge. We thought it likely that a covariance of zero was unrealistic, in that it did not allow for change, so the absolute value was tried for the elements of `CovBD`, and subsequent runs showed the solution did change.

However, although the first iterations seem to converge, the solution then blew-up as shown in Table 1. The initial value of parameter t_0 is approximately unity, but by the third iteration it has suddenly jumped to ~–60. Similarly, σ^0 begin at O(10) but then jumps suddenly to ~174 and never recovers.

It may be possible to introduce a third equation to the basic Bayes-Linear method for prognosing $V(w)$, because $V(w)$ will change as the solution, $E(\theta)$ drifts…

$$E(\theta \mid w) = E(\theta) + Cov(\theta, w)V(w)^{-1}(w - E(w(\theta))) \quad (1)$$

$$V(\theta \mid w) = V(\theta) + Cov(\theta, w)V(w)^{-1}Cov(w, \theta) \quad (2)$$

$$V(w \mid \theta) = ? \quad (3)$$

Θ (parameters)	t_0	h_s	σ^0	t_n
Iteration 1				
$E(\theta)$	1.09984	4.10033	10.5005	0.0115089
$V(\theta)$	0.0189718	0.0107833	0.0131588	-2.74099e-06
	0.0107833	0.00736421	0.00543578	-2.84535e-06
	0.0131588	0.00543578	0.0407997	-1.09274e-06
	-2.74099e-06	-2.84535e-06	-1.09274e-06	5.93302e-08
Iteration 2				
$E(\theta)$	0.0932742	4.4499	12.1191	0.00562172
$V(\theta)$	-0.904715	0.452287	0.356571	8.19343e-05
	0.452287	-0.27229	-0.0808159	-7.95593e-05
	0.356571	-0.0808159	-1.26706	-1.74342e-05
	8.19343e-05	-7.95593e-05	-1.74342e-05	-1.27637e-06
Iteration 3				
$E(\theta)$	-60.3986	7.4009	173.933	0.0224029
$V(\theta)$	-6.66292	1.29369	3.81143	0.000636976
	1.29369	-0.94886	0.365545	-0.000537464
	3.81143	0.365545	-35.5834	-0.000101854
	0.000636976	-0.000537464	-0.000101854	-2.51385e-05

Table 1 – the first three iterations when the C++ retracker is applied to the test dataset of figure 1. The retracker never recovers after the divergent values highlighted on iteration 3

Despite all efforts to debug the sequential retracking, it could not be made to converge, so the code was not plugged in the COASTALT processor as this was deemed pointless (also, the 'plug-and-play' feature described in [C2D12b] requires the user retracker to be written in FORTRAN so we would have had to further recode the Bayes Linear Retracker – not a big problem, but obviously only worth if the prototype C++ retracker worked)

3 Conclusions

The Bayes Linear Retracker had been identified in COASTALT Phase 1 as a very promising aspect, well worth investigating, with the plan of implementing and testing it on real waveforms.

NOC were fully committed to it, as also testified by the filing of a UK Patent Application in September 2009, whose details are given in Appendix C.

The software implementation of the Bayes Linear Retracker has, however, turned out to be much harder than anticipated. When we realized that we were not making progress towards a solution of the numerical issues encountered with the R code, we translated the retracker into C++, in the hope that this would allow extensive testing (as Luke West, the programmer in charge of the retracker testing and indeed of the coding of the COASTALT processor in Phase 2, is a C++ specialist). This translation and testing has indeed been carried out, including testing on real Envisat waveforms, and using time well in excess of the hours officially allocated to COASTALT for this task, but without success, as we have illustrated in section 2.

We have to conclude that this particular aspect of the Project, i.e. the development of a working Bayes Linear Retracker, has not achieved the results we had hoped to be able to achieve with the resources available in COASTALT. There are, however, still some achievements and recommendations to be highlighted.

Achievements:

- We have devised (in Phase 1 of COASTALT) and further developed (in Phase 2) a theoretical form of the Bayes Linear Retracker, presented in [C2D32], which still remains valid.
- The form presented has not been disproven and is certainly amenable to further investigations.

Recommendations:

- In future projects, more resources would need to be allocated to the software implementation of this retracker.
- The Taylor series expansion used in our implementation will not be applicable to numerical waveform models; in that case emulators should be tested, as explained in [C2D32]. This is a likely possibility for advanced altimeter designs such as the delay-Doppler altimeter.

Appendix A: Core of R implementation of Bayes Linear Retracker

```
bayes.linear.retracker <- function(params,w,vtheta,tauarray,nsamp) {
# the Bayes Linear altimeter retracker
 params[2:4]<- sqrt(params[2:4]^2)
# set the prior mean waveform
 prior.mean.w <- wavetheor(params, tauarray)
 presD <- vwinv(params,tauarray,nsamp)
 vw <- matrix(data=0,nrow=length(tauarray),ncol=length(tauarray))
 D <- gradient(tauarray,params)
 diag(vw) <- diag(t(D) %*% vtheta %*%  D)

 for (i in 1:length(tauarray)) {
  J <- matrix(data=0,nrow=length(params),ncol=length(params))
  J <- hessian(tauarray[i],params)
  diag(vw)[i] <- diag(vw)[i]+sum((diag(vtheta %*% J))^2)
 }

 covBD <- covtw(params,vtheta,tauarray)
 covBD <- covBD/sqrt(nsamp)


#  correct for the Taylor series approximation
# According to altimeter theory the variance at a point
# is w(theta,t) we have it set to vw. Calculate the ratio
 correction <- prior.mean.w/(diag(vw)*nsamp)
 correction <- sqrt(correction)

 for(i in (1:length(params))) {
  covBD[i,] <- covBD[i,] * correction
 }

Etheta.post <- params + covBD %*% presD %*% (w- prior.mean.w)
 delvtheta <- covBD %*% presD %*% t(covBD)
 Vtheta.post <- vtheta - delvtheta
 posterior <- list(mean=Etheta.post,var=Vtheta.post)
 return(posterior)
}
```

Appendix B: Core of C++ implementation of the R code.

```
for(size_t i=0;i<nb;i++) tauarray[i]=tlo+i*(thi-tlo)/(nb-1);
for(size_t i=0;i<nb;i++)
for(size_t j=0;j<nb;j++) presD.get(i,j)=0;

p_a<1,T > t; {
 p_cat<T > p(t,etheta);

 for(size_t i=0;i<nb;i++) {
  t[0]=tauarray(i); presD.get(i,i)=ns/wavetheor(p);
 }

 for(size_t i=0;i<nb;i++) {
  t[0]=tauarray(i); priormeanw[i] =   wavetheor(p);
 }
}

{
 p_a<1,T >     ind;
 p_cat<T >   q(ind,t);
 p_cat<T > p(q,etheta);

 for(size_t i=0;i<np;i++)
 for(size_t j=0;j<nb;j++) {
  t[0]=tauarray[j]; ind[0]=i+1; D1.get(i,j)=d1f(p);
 }
}

for(size_t i=0;i<nb;i++)
for(size_t j=0;j<nb;j++) vw.get(i,j)=0;

for(size_t i=0;i<nb;i++) {   size_t j=i; // just the diagonal
for(size_t p=0;p<np;p++)
for(size_t q=0;q<np;q++)
 vw.get(i,j)+=D1(p,i)*vtheta(p,q)*D1(q,j);
}
for(size_t i=0;i<nb;i++)
for(size_t p=0;p<np;p++) {
 covBD.get(p,i) =vtheta(p,p)*D1(p,i)/sqrt(ns);
 flo_t t=priormeanw(i)/vw.get(i,i);
 covBD.get(p,i)*=sqrt(priormeanw(i)/(vw.get(i,i)*ns));
}

{
 size_t i=32; p_a<2,T > ind;
 p_n<1,T > t(tauarray(i)); p_cat<T > p(t,etheta);
 f_D<2,T > hess(wavetheor,ind,eps(0));

 for(size_t i=0;i     < nb;i     ++) {t[0]=tauarray(i);
 for(  ind[0]=1;ind(0)<=np;ind[0]++)
 for(  ind[1]=1;ind(1)<=np;ind[1]++)
 for(size_t k=0;k     < np;k     ++)
  vw.get(i,i)+=Sqr(vtheta.get(ind(0)-1,k)*hess(p));
 }

 for(size_t j=0;j<np;j++)
 for(size_t k=0;k<nb;k++)
 for(size_t i=0;i<nb;i++)
  etheta.get(j)+=
```

```
    covBD.get(j,i)*presD.get(i,k)*(data(k)-priormeanw.get(k));

  for(size_t j=0;j<np;j++)
  for(size_t l=0;l<np;l++) {delvtheta.get(j,l)=0;
  for(size_t k=0;k<nb;k++)
  for(size_t i=0;i<nb;i++)
   delvtheta.get(j,l)+=covBD.get(j,i)*presD.get(i,k)*covBD(l,k);
  }

  for(size_t i=0;i<np;i++)
  for(size_t j=0;j<np;j++) vtheta.get(i,j)-=delvtheta(i,j);
}
```

Appendix C: Details on Patent Application (filing now lapsed)

An initial application for a UK patent on the use of Bayes Linear methods in tracking radar signals had been submitted by the University of Southampton for the Natural Environment Research Council (at that time these institutions were co-owners of the National Oceanography Centre). This had been filed on 15th Sept 2009 and had been assigned application number 0916175.3, and ESA had been informed on 16 December 2009 via the appropriate “Statement of Invention and Inventory” form (Appendix III to ESRIN Contract No. 21201/08/I-LG) confirming that ESA’s rights had been maintained.

The international application of this patent should have been pursued within 30 months (15 March 2012), requiring a substantial financial disbursement, but our decision in the end was not to proceed with the patent application. This was based on the view that, in the time since the patent was filed, there have been no clear commercialisation opportunities identified, which also reflects the fact that a clear demonstration of the technique has not been achieved due to the difficulties encountered in its numerical implementation. The Natural Environment Research Council’s policy in such a case is generally to let the filing lapse rather than fund the next stage.