

Alaska
Fisheries Science
Center

National Marine
Fisheries Service

U.S. DEPARTMENT OF COMMERCE

AFSC PROCESSED REPORT 2014-02

A Calibration Function for Correcting Mean
Net Spread Values Obtained from Marport
Spread Sensors Used in Conjunction with the
Marport MK II Receiver

April 2014

This document should be cited as follows:

Lauth, R. R., and S. Kotwicki. 2014. A calibration function for correcting mean net spread values obtained from Marport spread sensors used in conjunction with the Marport MK II receiver. AFSC Processed Rep. 2014-02, 26 p. Alaska Fish. Sci. Cent., NOAA, Natl. Mar. Fish. Serv., 7600 Sand Point Way NE, Seattle WA 98115.

Available at www.afsc.noaa.gov/Publications/ProcRpt/PR2014-02.pdf

Reference in this document to trade names does not imply endorsement by the National Marine Fisheries Service, NOAA.

INTRODUCTION

Acoustic net mensuration is used on chartered Alaska Fisheries Science Center (AFSC) bottom trawl (BT) survey vessels to monitor and record net spread during standard survey operations. Net spread measurements are a critical component for calculating area swept by the survey BT at each station (Alverson and Pereyra 1969). Area swept estimates are used to extrapolate catch sample weights and numbers to estimate survey abundance and biomass of commercially important bottom fishes and crabs to monitor their population trends (BSAI Groundfish Plan Team 2013). By the late 2000s, the existing inventory of the AFSC's acoustic net mensuration sensors and receivers, produced by either Scanmar or Netmind, Northstar Technical, Inc., were aging and becoming increasingly unreliable and costly to maintain and repair; hence, replacements had to be found. In 2009, the AFSC committed to replacing its inventory with sensors and receivers produced by a new and different manufacturer, Marport Deep Sea Technologies, Inc.. By 2013, the AFSC had purchased a sufficient quantity of Marport MK II acoustic receivers and net mensuration sensors to completely outfit its primary bottom trawl surveys. Previous dock tests using Scanmar and Netmind systems showed that either system produced accurate measurements of distance with a resolution of ± 0.1 m (R. Lauth unpublished data). To ensure that measurements from the new Marport system were comparable to the old AFSC net mensuration systems, further dock tests were conducted by the AFSC in 2012 to compare net spread measurements produced by the Scanmar and Marport systems. Results indicated that Marport spread sensors, when used in conjunction with the MK II acoustic receiver, produced imprecise and inaccurate spread measurements. Ensuing discussions with Marport representatives and engineers prompted Marport to conduct a more controlled test tank experiment in Lorient, France. The experiment was done using three fixed distances (11.90 m,

26.20 m, and 48.93 m) and a combination of the Marport spread sensors with three different acoustic receivers: the Marport MK II, the Marport M4 prototype, and the Scanmar Scanbass. Bottom depth and water salinity within the test tank were static and the master Marport spread sensor was preprogrammed with a 2-m offset and a default speed of sound in water of $1,482 \text{ m s}^{-1}$. Results from the tank test showed that the Marport spread sensors, when used in conjunction with the Marport M4 prototype and Scanmar Scanbass acoustic receivers, produced accurate results with good precision; however, results with the same spread sensors used in conjunction with the MK II acoustic receiver (the receiver used in AFSC BT surveys) were biased with poor precision (Table 1). Using the MK II receiver, the bias of the mean spread distance was + 0.81 m at the shortest fixed distance of 11.92 m. The bias of the mean spread distance was reduced slightly at fixed distances of 26.20 m (+ 0.64 m) and 48.90 m (- 0.15 m), but poor precision remained unchanged.

In anticipation of the switch to Marport in 2013, a field experiment was conducted during the 2012 EBS shelf BT survey (Lauth and Nichol 2013) to determine how the Marport net spread measurements using the MK II receiver varied as a function of Netmind net spread. Data collected from the experiment were used to derive a calibration function for adjusting the mean Marport net spread measurements in subsequent survey years whenever Marport net spread sensors were used in combination with the MK II acoustic receiver.

METHODS AND ANALYSIS

Comparison net spread data were collected on the survey charter vessels *Alaska Knight* and *Aldebaran* during the 2012 AFSC EBS shelf BT survey using both the Netmind and Marport acoustic net mensuration systems. The Netmind system, which has been used on EBS shelf BT

surveys since 2002 (Acuna et al. 2003), provided the benchmark for mean spread measurements. Net spread was measured as the distance between two sensors attached immediately forward of the connection of the upper breastline to the dandyline. The Marport and Netmind spread sensors were deployed on the same net at the same time with the master Marport spread sensor located forward of the starboard wing and the master Netmind spread sensor in the same location forward of the port wing. Slave sensors were attached to dandylines forward of the first set of sensors opposite of their respective master. A Marport MK II acoustic receiver was used for processing raw spread data from the Marport spread sensors, and a Netmind receiver was used for processing raw spread data from the Netmind sensors. The targeted standard tow duration was 30 minutes with a speed over ground of 3.0 knots.

A 2-m offset was preprogrammed into the Marport master spread sensor aboard the *Aldebaran* but not into the Marport sensor aboard the *Alaska Knight*; hence, the mean Marport net spread values for tows from the *Alaska Knight* were adjusted downward by 2-m to make them comparable to those from the *Aldebaran*. Master spread sensors from both vessels were preprogrammed with the speed of sound set at $1,500 \text{ m s}^{-1}$. Start and end times for each tow were selected using bottom contact sensor data to determine the time when the footrope first touched the seafloor and the time it departed the seafloor.

Net spread data near the start and end times of tows were often sparse or missing, and this data scarcity varied by vessel, depth of tow, and the type of net mensuration system being used. To minimize bias caused by these potentially non-random begin- and end-of-tow effects, only data from 3 minutes after start time to 3 minutes before end time were used in the analysis. Because asymmetrically distributed outliers and an unequal density of net spread values are typical in raw net spread data and introduce bias when calculating an arithmetic mean (Kotwicki

et al. 2011), the following steps were taken to calculate net spread means and standard deviations. First, the R program was used to iteratively apply sequential outlier rejection (SOR; Kotwicki et al. 2011) and smoothing procedures to the raw Netmind and Marport net spread data (Appendix 1). To avoid the large influence of sparse data points on the smoothing algorithm, sparse data were removed first. “Sparse” was defined as less than 4 data points in any 1-minute interval during a tow. The SOR procedure was initiated by fitting a cubic spline smoother (Hastie and Tibshirani 1990) to the net spread data and removing a single data point that was identified as having largest absolute value of the residual from smoothing procedure. This procedure was iteratively repeated until the stopping rule for the SOR was reached based on simulations performed by Kotwicki et al. (2011). Second, the mean net spread and standard deviation were calculated from the fitted smoothed line of the remaining data points. This procedure resulted in reduced bias in the estimation of mean net spread compared to the usual method of using a simple arithmetic mean of observed net spread values within a fixed range of values from 8 to 22 m. Finally, a simple linear regression of mean Netmind net spread as a function of mean Marport net spread was done for each vessel. If between-vessel differences in the regression slopes and intercepts were not significant, datasets were pooled and the regression analysis repeated with combined data.

RESULTS AND DISCUSSION

Paired Netmind and Marport spread data were obtained from 136 tows aboard the *Aldebaran* and 122 tows aboard the *Alaska Knight*. Mean net spread and standard deviation were calculated for all 258 tows using the SOR procedure (Appendix 2, Fig. 1). Good linear regression fits were

obtained using mean Netmind net spread as the dependent variable and mean Marport net spread as the independent variable for both the *Aldebaran* ($R^2 = 0.97$, $P = 1.8 \times 10^{-103}$) and the *Alaska Knight* ($R^2 = 0.97$, $P = 7.42 \times 10^{-95}$). The relationship for both vessels indicated a decreasing offset in the mean Netmind spread as the Marport spread increased. There were no significant differences between the slopes ($P = 0.58$) and intercepts ($P = 0.99$) of the two linear regressions, so data from both vessels were pooled and the regression analysis was repeated. Good fits were also obtained with combined data ($R^2 = 0.97$, $P = 6.9 \times 10^{-191}$) producing the following equation:

$$\text{Mean Netmind spread} = 0.9357 \times \text{Mean Marport spread} + 0.4005.$$

There was general agreement between the results from this field experiment and the test tank results from Lorient, France, done by Marport using the MK II acoustic receiver (Fig. 2). Both indicated decreasing bias with increasing separation between the master and slave spread sensors. The good agreement between the test tank results using known fixed distances and the experimental results using Netmind spread measurements as a proxy for “true” distances corroborates that the measurements from the Netmind net mensuration system provided a good benchmark for unbiased estimates of net spread (Fig. 2). The difference in slope (~ 0.08) was not significant because of the relative low sample size from the test tank experiment ($n = 3$) compared to the field experiment ($n = 258$); however, two factors could have resulted in such a difference. First, the speed of sound through water varies with changes in water temperature, salinity, and pressure (Kotwicki et al. 2011) and changes in these physical variables can affect how acoustic net spread measurements are processed. The field experiment was conducted under

in situ conditions capturing a wide range of bottom depths, temperatures, and salinities, but the default speed of sound for processing acoustic signals from sensors used the field experiment was fixed at 1,500 m s⁻¹. The speed of sound for processing acoustic signals from the test tank experiment was set at 1,482 m/s, and the depth, temperature, and salinity in the test tank was relatively constant. A second explanation could be that estimates of mean net spread from the field experiment using a SOR procedure and cubic spline smoother resulted in lower bias estimates than the simple arithmetic means from the test tank experiment.

The combination of Marport spread sensors and the Marport MK II acoustic receiver was used to collect net spread data during the 2013 EBS shelf BT survey. In the analysis of all tows from the 2013 survey, the SOR and cubic spline smoother procedures (Kotwicki et al. 2011) were used to estimate mean Marport net spread and the calibration function from this experiment was used to adjust mean net spreads. The reason for doing this was to eliminate bias relative to mean net spreads estimated from previous survey years which were derived using only the observations from the Netmind gear mensuration system. The intention is to continue using the procedure above for adjusting Marport spread data in the future when using in combination with the MK II acoustic receiver. The adjustment may no longer be necessary once the Marport M4 is in production and being used in AFSC BT surveys. Nevertheless, this study underscores the importance of evaluating new or updated spread sensors and acoustic receivers prior to their introduction into AFSC BT surveys to avoid or minimize the introduction of bias into area swept calculations.

CITATIONS

- Acuna, E., P. Goddard, and S. Kotwicki (Compilers). 2003. 2002 bottom trawl survey of the eastern Bering Sea continental shelf. AFSC Processed Rep. 2003-01, 165 p. Alaska Fish. Sci. Cent., Natl. Mar. Fish. Serv., NOAA, 7600 Sand Point Way NE, Seattle WA 98115.
- Alverson, D. L., and W. T. Pereyra. 1969. Demersal fish explorations in the northeast Pacific Ocean--An evaluation of exploratory fishing methods and analytical approaches to stock size and yield forecasts. *J. Fish. Res. Board Can.* 26:1985-2001.
- Bering Sea Aleutian Island Groundfish Plan Team. 2013. Stock assessment and fishery evaluation report for the groundfish resources of the Bering Sea/Aleutian Islands region. North Pacific Fishery Management Council, 605 W. 4th Ave., Anchorage, AK. 1,064 p.
- Hastie, T. J., and R. J. Tibshirani. 1990. *Generalized Additive Models*, Volume 43 of *Monographs in Statistics and Applied Probability*. Chapman and Hall, London.
- Kotwicki, S., M. H. Martin, and E. A. Laman. 2011. Improving area swept estimates from bottom trawl surveys. *Fish. Res.* 110:198-206.
- Lauth, R. R., and D. G. Nichol. 2013. Results of the 2012 eastern Bering Sea continental shelf bottom trawl survey of groundfish and invertebrate resources. U.S. Dep. Commer., NOAA Tech. Memo. NMFS-AFSC-256, 162 p.

Table 1. Summary of results from a test tank experiment done in Lorient, France, using Marport spread sensors with three different acoustic receivers at three fixed distances.

Fixed distance	Measured distance by receiver (m)								
	MK II			M4 prototype			Scanmar Scanbass		
	Average	Precision	Error	Average	Precision	Error	Average	Precision	Error
11.90 m	12.71	Poor	0.81	11.93	Good	0.03	11.92	Good	0.02
26.20 m	26.84	Poor	0.64	26.25	Good	0.05	26.27	Good	0.07
48.93 m	48.78	Poor	-0.15	48.77	Good	-0.16	48.95	Good	0.02

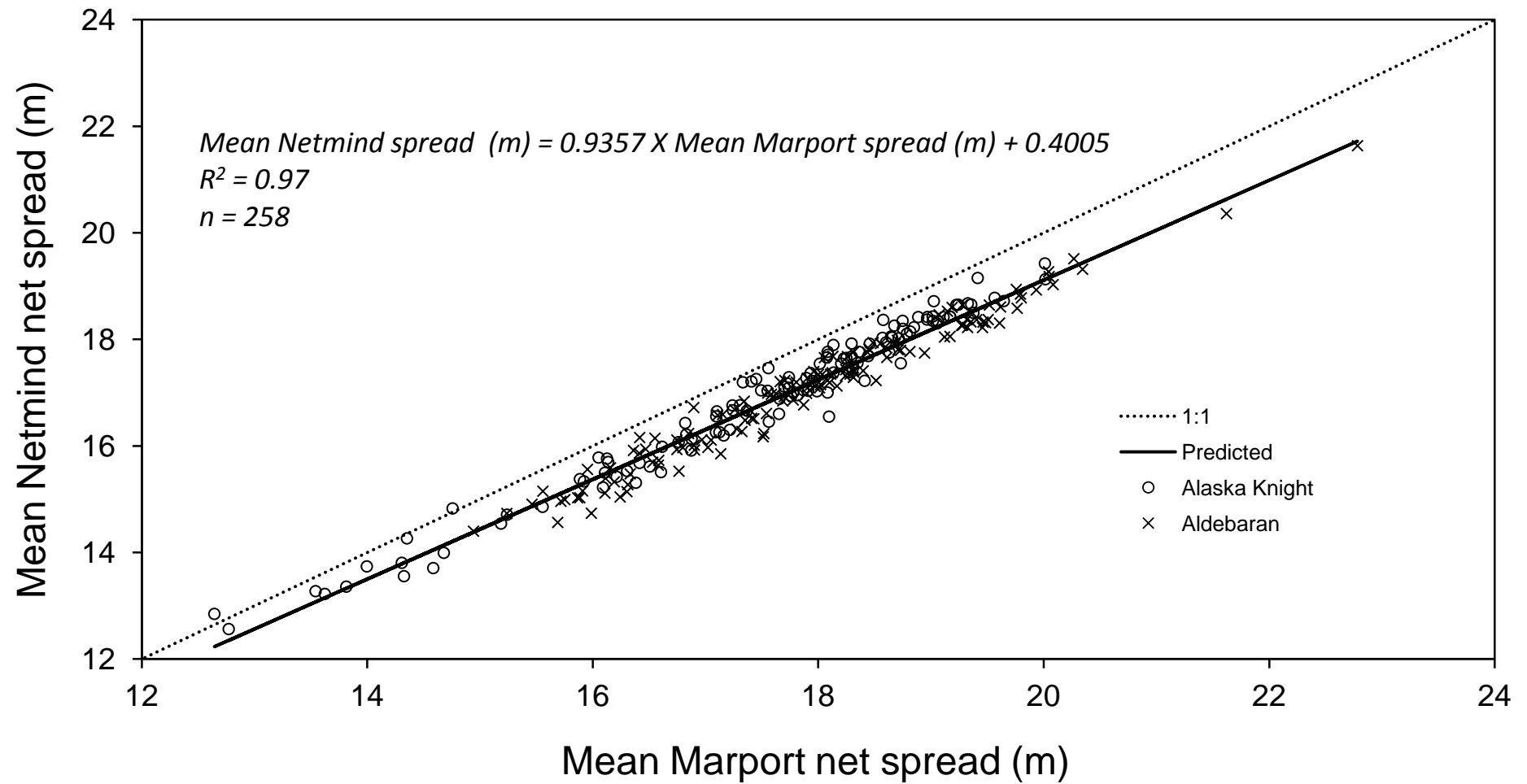


Figure 1. The relationship and simple linear regression of the mean Netmind net spread as a function of mean Marport net spread from field experiment conducted in summer 2012.

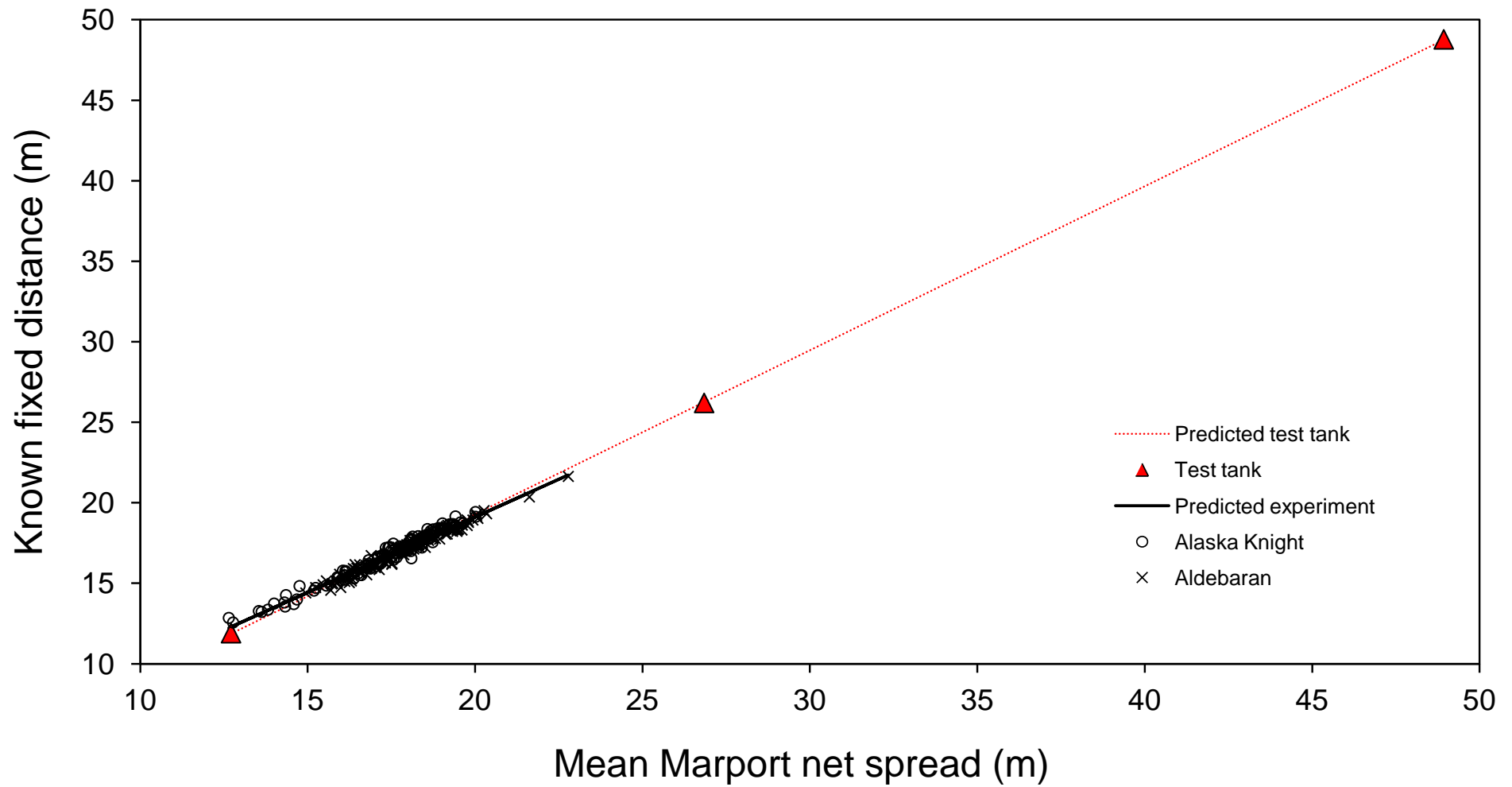


Figure 2. Test tank results from March 2012 using three known fixed distances overlaid with results from the 2012 field experiment. Tank tests were conducted by Marport Deepsea Technologies, Inc. using Marport spread sensors and a Marport Mark II acoustic receiver.

Appendix 1. SOR instructions and R code (version 2.1.5.2) used to obtain mean net spreads for Marport and Netmind.

SOR instructions

1. Copy and paste SOR_table folder into your working directory
2. Put events and spread file in the SOR_table directory
3. Open and run script “do_sor_table.R”.
It is important that you run this script line by line using (Ctr R). I can make it more user friendly later, but for now just run it one line at a time.
4. Scripts will produce csv files for each haul and one large csv file for all hauls that you have in the directory. At one point large file may become to large for R and R may crash. I do not know what this limit is, but if R crashes try to reduce number of hauls in both event and spread files.

R file do_sor_table.R

```
source('sor_table.R')
events=read.events()
haul.data=read.hauls()
sor(haul.data,events,flag=12)
output_sor_table()
dev.off()
sor.means = estimate.means(events)
write.csv(sor.means, 'sor_means.csv')
```

R file sor_table.R

```
require(tcltk)

read.events=function(){
events.file=tclvalue(tkgetOpenFile())
events=read.csv(events.file)
return(events)
}

read.hauls=function(){
haul.data.file=tclvalue(tkgetOpenFile())
haul.data=read.csv(haul.data.file)
return(haul.data)
}

sor = function(haul.data,events,flag=12){
events$DTIME = as.POSIXct(strptime(events$DTIME, "%m/%d/%Y
%H:%M:%OS"))
haul.list=unique(haul.data$HAUL)
for(i in haul.list){
  sgt=haul.data[haul.data$HAUL==i,]
```

```

#sgt$DATE_TIME=sub('/12','/2012',sgt$DATE_TIME)
sgt$DATE_TIME = as.POSIXct(strptime(sgt$DATE_TIME, "%m/%d/%Y
%H:%M:%OS"))
sgt=sgt[sgt$CABINET_SENSOR_FLAG==flag,]
haul=sgt$HAUL[1]
if(length(which(events$HAUL==haul))==2){
on.bottom=events[events$HAUL==haul & events$EVENT==3,]$DTIME
off.bottom=events[events$HAUL==haul & events$EVENT==7,]$DTIME
yy=sgt$MEASUREMENT_VALUE[sgt$MEASUREMENT_VALUE<30 &
sgt$DATUM_CODE==0]
xx=cbind(sgt$RECORD_ID,sgt$DATE_TIME)[sgt$MEASUREMENT_VALUE<30 &
sgt$DATUM_CODE==0,]
point.density = as.data.frame(cbind(
aa1=c(1000,1000,1000,xx[,2][4:(length(xx[,2]))]-
xx[,2][1:(length(xx[,2])-3)]),
aa2=c(1000,1000,xx[,2][3:(length(xx[,2]))]-xx[,2][1:(length(xx[,2])-
2)]),
aa3=c(1000,xx[,2][2:(length(xx[,2]))]-xx[,2][1:(length(xx[,2])-1)]),
aa4=c(xx[,2][1:(length(xx[,2])-1)]-xx[,2][2:length(xx[,2])],-1000),
aa5=c(xx[,2][1:(length(xx[,2])-2)]-xx[,2][3:length(xx[,2])],-1000,-
1000),
aa6=c(xx[,2][1:(length(xx[,2])-3)]-xx[,2][4:length(xx[,2])],-1000,-
1000,-1000)
))
idx.rej=(point.density$aa2<60 | point.density$aa5>-60 |
point.density$aa3-point.density$aa4<60)
xx2=xx[idx.rej,]
yy2=yy[idx.rej]
if(length(yy2)>10){
par(mfrow=c(3,1),mar=c(2, 4, 1, 2))
plot(xx[,2],yy,ylim=c(0,40),main=paste('Haul',haul,'all data'),
xlim=c(on.bottom-360,off.bottom+360))
mdl=smooth.spline(yy~xx[,2],spar=.8)
smooth=predict(mdl)
lines(smooth,col='red')}
xx1=xx[xx[,2]>on.bottom & xx[,2]<off.bottom,]
yy1=yy[xx[,2]>on.bottom & xx[,2]<off.bottom]
if(length(yy1)>10){
plot(xx1[,2],yy1,ylim=c(0,40),
xlim=c(on.bottom-360,off.bottom+360))
abline(v=on.bottom,col=2)
abline(v=off.bottom,col=2)
mtext("click here", side = 3, line=0)
md2=smooth.spline(yy1~xx1[,2],spar=.8)
smooth2=predict(md2)
lines(smooth2,col='red')}
xx3=xx2[xx2[,2]>on.bottom & xx2[,2]<off.bottom,]
yy3=yy2[xx2[,2]>on.bottom & xx2[,2]<off.bottom]
if(length(yy3)>10){
plot(xx3[,2],yy3,ylim=c(0,40),
xlim=c(on.bottom-360,off.bottom+360))
abline(v=on.bottom,col=2)

```



```

abline(v=off.bottom,col=2)
mtext("sparse data removed", side = 3, line=0)
md3=smooth.spline(yy3~xx3[,2],spar=.8)
smooth3=predict(md3)
lines(smooth3,col='red')
a = locator(n=1)

resids=yy3-smooth3$y
max.dist = max(abs(resids))
if(sd(resids)<5) {
  stop.dist=-0.3034*sd(resids)^2 + 2.9428*sd(resids)-0.1112
} else {stop.dist=7}
idx=which(abs(resids)==max.dist)
yy4=yy3
xx4=xx3
for(j in 1:length(yy3)){
  yy4=yy4[-idx]
  xx4=xx4[-idx,]
  #plot(xx4,yy4,ylim=c(0,40))
  md4=smooth.spline(yy4~xx4[,2],spar=.8)
  smooth=predict(md4)
  #lines(smooth,col='red')
  resids=yy4-smooth$y
  max.dist = max(abs(resids))
  if(max.dist>stop.dist){
    idx=which(abs(resids)==max.dist)
    #a = locator(n=1)
  } else {
    print("stopping distance reached")
    #plot(xx4,yy4,ylim=c(0,40))
    md4=smooth.spline(yy4~xx4[,2],spar=.8)
    smooth=predict(md4)
    #lines(smooth,col='red')
    #mtext("data after SOR", side = 3)
    #mtext("click here to acknowledge plot review", side = 1, line=4)
    #a = locator(n=1)
    break
  }
}
plot(xx4[,2],yy4,ylim=c(0,40),xlim=c(on.bottom-360,off.bottom+360),
      ,main=paste('Haul',haul,'data after SOR'))
lines(smooth,col='red')
abline(v=on.bottom,col=2)
abline(v=off.bottom,col=2)

accept = readline("Do you accept SOR solution? (y or n)")
if(accept=="y"){
  DATUM_CODE_N=data.frame(DATUM_CODE_N=rep.int(0,length(xx4[,2])))
  XX=data.frame(xx4)
  colnames(XX)[1]="RECORD_ID"
  newdata=cbind(XX,DATUM_CODE_N)
  newdata1 = merge(sgt,newdata,all=T)
}

```

```

newdata1=newdata1[,-11]
newdata1$DATE_TIME=format(newdata1$DATE_TIME, "%m/%d/%Y %H:%M:%S")
newdata1$DATUM_CODE_N[is.na(newdata1$DATUM_CODE_N)]=10

newdata1$DATUM_CODE_N[newdata1$DATUM_CODE!=0]=newdata1$DATUM_CODE[newd
ata1$DATUM_CODE!=0]
newdata1$DATUM_CODE=newdata1$DATUM_CODE_N
newdata1=newdata1[,-11]
write.csv(newdata1,paste("sor_h",haul,".csv",sep=""), quote =
F,row.names = F)
} else {
DATUM_CODE_N=data.frame(DATUM_CODE_N= rep.int(10,length(sgt[,1])))
newdata1=cbind(sgt,DATUM_CODE_N)
newdata1$DATE_TIME=format(newdata1$DATE_TIME, "%m/%d/%Y %H:%M:%S")

newdata1$DATUM_CODE_N[newdata1$DATUM_CODE!=0]=newdata1$DATUM_CODE[newd
ata1$DATUM_CODE!=0]
newdata1$DATUM_CODE=newdata1$DATUM_CODE_N
newdata1=newdata1[,-11]
write.csv(newdata1,paste("sor_h",haul,".csv",sep=""), quote =
F,row.names = F)
print("Spread data rejected")}
} else {
plot(0,0,pch="",main=paste('Haul',haul))
text(0,0,"data not sufficient")
a=locator(n=1)
}
} else {
plot(0,0,pch="",main=paste('Haul',haul))
text(0,0,"no on bottom times for this haul")
a=locator(n=1)
}
}

}

#sor(haul.data,events)

output_sor_table=function(){
file.type = 'sor_h'
file.list=list.files(pattern=file.type)
sor.data=NA
for(i in 1:length(file.list)){
sor.data1=read.csv(file.list[i])
sor.data=rbind(sor.data,sor.data1)
}
sor.data=sor.data[-1,]
write.csv(sor.data,"sor_output.csv")
}

```

```

estimate.means = function(events){

events$DTIME = as.POSIXct(strptime(events$DTIME, "%m/%d/%Y
%H:%M:%OS"))
haul.data = read.csv('sor_output.csv')
haul.list=unique(haul.data$HAUL)

Haul=NA
mean.spread=NA
N=NA
resid.sd=NA

for(i in 1:length(haul.list)){
  sgr=haul.data[haul.data$HAUL==haul.list[i],]
  sgr$DATE_TIME = as.POSIXct(strptime(sgr$DATE_TIME, "%m/%d/%Y
%H:%M:%OS"))
  haul=sgr$HAUL[1]
  Haul[i]=haul
  if(length(which(events$HAUL==haul))==2){
    on.bottom=events[events$HAUL==haul & events$EVENT==3,]$DTIME
    off.bottom=events[events$HAUL==haul & events$EVENT==7,]$DTIME
    xx=sgr$DATE_TIME[sgr$DATUM_CODE==0]
    yy=sgr$MEASUREMENT_VALUE[sgr$DATUM_CODE==0]
    xx1=xx[xx>on.bottom & xx<off.bottom]
    yy1=yy[xx>on.bottom & xx<off.bottom]
    if(length(yy1)>10){
      plot(xx1,yy1,ylim=c(0,40),main=haul,
           xlim=c(on.bottom-360,off.bottom+360))
      abline(v=on.bottom,col=2)
      abline(v=off.bottom,col=2)
      md2=smooth.spline(yy1~xx1,spar=.8)
      smooth2=predict(md2)
      lines(smooth2,col='red')#
      min.xx1=min(xx1)
      max.xx1=max(xx1)

preds=predict(md2,x=as.numeric(seq(min.xx1,max.xx1,length.out=500)))
      difs.xx=smooth2$x[2:length(smooth2$x)]-
smooth2$x[1:(length(smooth2$x)-1)]
      difs.preds=preds$x[2]-preds$x[1]
      gaps=which(difs.xx>20)
      for(gg in gaps){
        xx1.from=smooth2$x[gg]
        xx1.to=smooth2$x[gg+1]
        idx.preds=which(preds$x>xx1.from & preds$x<xx1.to)
        yyp1=preds$y[min(idx.preds)-1]
        yyp2=preds$y[max(idx.preds)+1]
        xxp1=preds$x[min(idx.preds)-1]
        xxp2=preds$x[max(idx.preds)+1]
        incr.gap=(yyp2-yyp1)/(length(idx.preds)+1)
        predsb=yyp1+c(1:length(idx.preds))*incr.gap
        preds$y[idx.preds]=predsb

```

```

    }
    lines(preds,col=3)
    mean.spread[i]=mean(preds$y)
    N[i]=length(yy1)
    resid.sd[i]=sd(residuals(md2))
    mtext(paste('Mean =',round(mean.spread[i],2),'SD
=' ,round(resid.sd[i],2)),
          side = 3, line=0)
    a=locator(n=1)
  } else {
    plot(0,0,pch="",main=paste('Haul',haul))
    text(0,0,"data not sufficient")
    a=locator(n=1)
  }
  } else {
    plot(0,main="no on or off bottom times")
    mean.spread=mean(preds$y)
    a=locator(n=1)
  }
}
}
summary.spreads =
as.data.frame(cbind(Haul=Haul,N=N,Mean=mean.spread,SD=resid.sd))
return(summary.spreads)
}

```

```
#estimate.means(events)
```

R file means sor

```

estimate.means = function(events){

events$DTIME = as.POSIXct(strptime(events$DTIME, "%m/%d/%Y
%H:%M:%OS"))
haul.data = read.csv('sor_output.csv')
haul.list=unique(haul.data$HAUL)

Haul=NA
mean.spread=NA
N=NA
resid.sd=NA

for(i in 1:length(haul.list)){
  sgr=haul.data[haul.data$HAUL==haul.list[i],]
  sgr$DATE_TIME = as.POSIXct(strptime(sgr$DATE_TIME, "%m/%d/%Y
%H:%M:%OS"))
  haul=sgr$HAUL[1]
  Haul[i]=haul
  if(length(which(events$HAUL==haul))==2){
    on.bottom=events[events$HAUL==haul & events$EVENT==3,]$DTIME
    off.bottom=events[events$HAUL==haul & events$EVENT==7,]$DTIME
    xx=sgr$DATE_TIME[sgr$DATUM_CODE==0]
    yy=sgr$MEASUREMENT_VALUE[sgr$DATUM_CODE==0]

```

```

xx1=xx[xx>on.bottom & xx<off.bottom]
yy1=yy[xx>on.bottom & xx<off.bottom]
if(length(yy1)>10){
  plot(xx1,yy1,ylim=c(0,40),main=haul,
       xlim=c(on.bottom-360,off.bottom+360))
  abline(v=on.bottom,col=2)
  abline(v=off.bottom,col=2)
  md2=smooth.spline(yy1~xx1,spar=.8)
  smooth2=predict(md2)
  lines(smooth2,col='red')

preds=predict(md2,x=as.numeric(seq(on.bottom,off.bottom,length.out=100
)))
  mean.spread[i]=mean(preds$y)
  N[i]=length(yy1)
  resid.sd[i]=sd(residuals(md2))
  mtext(paste('Mean =',round(mean.spread[i],2),'SD
=' ,round(resid.sd[i],2)),
        side = 3, line=0)
  a=locator(n=1)
} else {
  plot(0,0,pch=" ",main=paste('Haul',haul))
  text(0,0,"data not sufficient")
  a=locator(n=1)
}
} else {
  plot(0,main="no on or off bottom times")
  mean.spread=mean(preds$y)
  a=locator(n=1)
}
}
summary.spreads =
as.data.frame(cbind(Haul=Haul,N=N,Mean=mean.spread,SD=resid.sd))
return(summary.spreads)
}

#estimate.means(events)

```


Appendix 2. Mean, standard deviation, and ping count for Marport and Netmind net spreads by vessel and haul resulting from the sequential outlier rejection procedure.

Vessel	Haul	Mean	Number of pings	Standard deviation	Mean	Number of pings	Standard deviation
		Marport net spread		Marport net spread	Netmind net spread		Netmind net spread
<i>Alaska Knight</i>	1	14.585	389	0.730	13.707	227	0.979
<i>Alaska Knight</i>	4	16.874	342	0.937	15.914	198	1.270
<i>Alaska Knight</i>	7	13.814	442	0.490	13.359	146	0.930
<i>Alaska Knight</i>	8	15.186	410	0.793	14.543	209	1.045
<i>Alaska Knight</i>	9	14.679	434	0.731	13.993	162	1.042
<i>Alaska Knight</i>	17	16.212	346	0.457	15.421	141	0.531
<i>Alaska Knight</i>	18	16.382	384	0.534	15.309	137	0.359
<i>Alaska Knight</i>	19	17.124	386	0.609	16.258	140	0.767
<i>Alaska Knight</i>	20	16.830	383	0.588	16.216	131	0.718
<i>Alaska Knight</i>	21	18.096	178	0.837	16.550	133	0.650
<i>Alaska Knight</i>	23	14.326	375	0.454	13.555	75	1.025
<i>Alaska Knight</i>	26	17.868	392	0.464	17.048	120	0.708
<i>Alaska Knight</i>	27	17.562	91	0.553	16.454	144	0.764
<i>Alaska Knight</i>	28	17.652	386	0.806	16.601	79	0.784
<i>Alaska Knight</i>	31	16.605	344	0.821	15.508	157	1.031
<i>Alaska Knight</i>	32	16.507	417	0.732	15.613	191	1.270
<i>Alaska Knight</i>	33	14.307	412	0.623	13.806	88	1.185
<i>Alaska Knight</i>	34	14.352	434	0.593	14.265	150	1.124
<i>Alaska Knight</i>	35	15.555	419	0.695	14.857	248	0.926
<i>Alaska Knight</i>	40	17.239	417	0.527	16.761	227	0.479
<i>Alaska Knight</i>	41	17.161	309	0.611	16.196	244	0.269
<i>Alaska Knight</i>	42	17.216	403	0.520	16.302	263	0.233
<i>Alaska Knight</i>	43	17.700	395	0.387	17.114	251	0.364
<i>Alaska Knight</i>	44	16.415	404	0.370	15.681	266	0.365
<i>Alaska Knight</i>	46	18.732	101	0.652	17.551	124	0.333
<i>Alaska Knight</i>	49	18.229	375	0.401	17.642	197	0.484
<i>Alaska Knight</i>	50	20.015	327	0.480	19.129	152	0.544
<i>Alaska Knight</i>	51	19.170	406	0.337	18.425	158	0.465
<i>Alaska Knight</i>	52	19.646	380	0.376	18.719	156	0.451
<i>Alaska Knight</i>	53	19.565	363	0.372	18.780	85	0.393
<i>Alaska Knight</i>	54	18.647	329	0.456	18.055	82	0.502
<i>Alaska Knight</i>	55	18.362	392	0.364	17.766	191	0.542
<i>Alaska Knight</i>	56	18.457	408	0.399	17.919	201	0.560
<i>Alaska Knight</i>	58	17.989	108	0.541	17.026	217	0.459
<i>Alaska Knight</i>	59	18.412	105	0.816	17.222	170	0.578
<i>Alaska Knight</i>	66	13.996	423	0.503	13.738	166	0.722
<i>Alaska Knight</i>	73	12.644	413	0.619	12.847	223	1.072
<i>Alaska Knight</i>	78	18.083	265	0.637	17.002	161	0.354
<i>Alaska Knight</i>	79	17.551	407	0.474	17.035	213	0.483
<i>Alaska Knight</i>	80	18.849	393	0.433	18.227	115	0.557
<i>Alaska Knight</i>	81	18.087	415	0.347	17.712	206	0.359
<i>Alaska Knight</i>	86	20.010	404	1.053	19.425	256	1.126
<i>Alaska Knight</i>	87	18.137	307	0.678	17.378	240	0.413
<i>Alaska Knight</i>	90	18.084	342	0.738	17.766	152	0.385
<i>Alaska Knight</i>	91	17.780	338	0.523	17.012	97	0.466
<i>Alaska Knight</i>	96	16.137	416	0.753	15.699	55	0.531

Appendix 2. Continued.

Vessel	Haul	Mean Marport net spread	Number of pings	Standard deviation Marport net spread	Mean Netmind net spread	Number of pings	Standard deviation Netmind net spread
<i>Alaska Knight</i>	104	13.624	417	0.529	13.222	117	0.642
<i>Alaska Knight</i>	106	12.771	406	0.583	12.563	80	0.908
<i>Alaska Knight</i>	111	13.542	419	0.545	13.274	76	0.421
<i>Alaska Knight</i>	114	16.052	411	0.622	15.782	81	0.261
<i>Alaska Knight</i>	118	16.305	417	0.759	15.529	155	0.266
<i>Alaska Knight</i>	119	17.092	402	0.702	16.257	275	0.274
<i>Alaska Knight</i>	123	15.922	401	0.632	15.334	254	0.274
<i>Alaska Knight</i>	124	16.108	416	0.714	15.497	228	0.274
<i>Alaska Knight</i>	128	17.657	332	0.602	16.885	136	0.363
<i>Alaska Knight</i>	129	17.952	391	0.399	17.218	122	0.349
<i>Alaska Knight</i>	132	17.910	408	0.586	17.034	207	0.213
<i>Alaska Knight</i>	133	18.306	356	0.418	17.453	36	0.419
<i>Alaska Knight</i>	134	18.631	387	0.572	17.755	198	0.257
<i>Alaska Knight</i>	135	17.367	394	0.323	16.661	59	0.371
<i>Alaska Knight</i>	136	16.877	419	0.507	16.133	181	0.327
<i>Alaska Knight</i>	137	15.240	409	0.623	14.709	223	0.294
<i>Alaska Knight</i>	138	16.820	421	0.757	16.430	85	0.269
<i>Alaska Knight</i>	139	16.127	411	0.789	15.762	39	0.348
<i>Alaska Knight</i>	140	16.615	418	0.791	15.985	220	0.349
<i>Alaska Knight</i>	143	17.243	431	0.527	16.669	235	0.335
<i>Alaska Knight</i>	147	17.306	413	0.640	16.764	148	0.396
<i>Alaska Knight</i>	148	19.025	410	0.845	18.717	50	0.675
<i>Alaska Knight</i>	152	18.601	385	0.739	17.943	256	0.399
<i>Alaska Knight</i>	153	16.091	341	1.115	15.221	109	0.329
<i>Alaska Knight</i>	155	17.095	413	0.607	16.555	178	0.303
<i>Alaska Knight</i>	158	18.266	426	0.373	17.458	268	0.235
<i>Alaska Knight</i>	159	18.348	431	0.247	17.558	281	0.309
<i>Alaska Knight</i>	160	18.272	387	0.493	17.382	258	0.196
<i>Alaska Knight</i>	161	18.297	411	0.339	17.676	264	0.334
<i>Alaska Knight</i>	162	17.737	325	0.894	17.087	216	0.301
<i>Alaska Knight</i>	163	17.331	405	0.256	17.194	63	0.319
<i>Alaska Knight</i>	165	17.450	403	0.404	17.256	43	0.333
<i>Alaska Knight</i>	166	17.816	1220	0.583	16.953	25	0.273
<i>Alaska Knight</i>	167	17.932	418	0.407	17.367	247	0.384
<i>Alaska Knight</i>	168	18.296	415	0.515	17.922	110	0.251
<i>Alaska Knight</i>	169	17.740	391	0.314	17.290	140	0.380
<i>Alaska Knight</i>	171	18.014	413	0.259	17.544	243	0.386
<i>Alaska Knight</i>	172	18.203	424	0.264	17.547	247	0.301
<i>Alaska Knight</i>	173	18.712	321	0.460	18.021	269	0.325
<i>Alaska Knight</i>	174	17.895	395	0.460	17.275	225	0.364
<i>Alaska Knight</i>	175	17.725	392	0.482	17.172	105	0.401
<i>Alaska Knight</i>	176	18.443	442	0.548	17.686	270	0.325
<i>Alaska Knight</i>	177	18.816	389	0.433	18.147	237	0.293
<i>Alaska Knight</i>	179	18.076	418	0.248	17.658	215	0.341
<i>Alaska Knight</i>	180	18.249	392	0.256	17.651	252	0.369
<i>Alaska Knight</i>	184	19.015	245	0.506	18.429	155	0.367
<i>Alaska Knight</i>	185	18.675	385	0.475	18.258	231	0.466

Appendix 2. Continued.

Vessel	Haul	Mean	Standard	Mean	Standard		
		Marport net spread	deviation Number of pings	Netmind net spread	deviation Number of Netmind net spread		
<i>Alaska Knight</i>	186	19.111	399	0.357	18.417	271	0.242
<i>Alaska Knight</i>	187	19.241	401	0.230	18.651	120	0.506
<i>Alaska Knight</i>	189	19.357	399	0.274	18.654	199	0.264
<i>Alaska Knight</i>	190	19.224	405	0.260	18.646	180	0.388
<i>Alaska Knight</i>	191	18.969	413	0.243	18.420	227	0.287
<i>Alaska Knight</i>	192	19.376	375	0.438	18.497	220	0.348
<i>Alaska Knight</i>	193	19.063	418	0.255	18.339	267	0.197
<i>Alaska Knight</i>	194	18.655	421	0.214	18.035	275	0.275
<i>Alaska Knight</i>	195	19.169	422	0.255	18.427	271	0.223
<i>Alaska Knight</i>	196	18.570	423	0.229	18.028	253	0.326
<i>Alaska Knight</i>	197	18.786	404	0.257	18.106	260	0.380
<i>Alaska Knight</i>	199	18.577	358	0.605	18.366	249	0.708
<i>Alaska Knight</i>	201	19.414	405	0.804	19.152	216	0.913
<i>Alaska Knight</i>	202	18.134	406	0.288	17.895	228	0.484
<i>Alaska Knight</i>	203	17.408	414	0.329	17.210	152	0.429
<i>Alaska Knight</i>	204	17.761	412	0.526	16.936	265	0.180
<i>Alaska Knight</i>	205	17.494	398	0.594	17.042	109	0.345
<i>Alaska Knight</i>	206	18.748	416	0.670	18.348	142	0.705
<i>Alaska Knight</i>	207	17.558	394	0.396	17.460	64	0.512
<i>Alaska Knight</i>	211	18.887	416	0.260	18.418	96	0.501
<i>Alaska Knight</i>	212	18.965	368	0.304	18.372	128	0.533
<i>Alaska Knight</i>	213	19.327	396	0.295	18.677	143	0.396
<i>Alaska Knight</i>	214	18.751	399	0.212	18.198	169	0.377
<i>Alaska Knight</i>	215	19.017	394	0.534	18.347	133	0.269
<i>Alaska Knight</i>	216	19.054	389	0.381	18.299	104	0.471
<i>Alaska Knight</i>	218	17.101	387	0.619	16.648	46	0.733
<i>Alaska Knight</i>	219	14.757	380	0.553	14.827	58	0.939
<i>Alaska Knight</i>	221	15.886	396	0.787	15.377	178	0.543
<i>Alaska Knight</i>	224	16.762	396	0.646	16.072	96	1.593
<i>Aldebaran</i>	6	16.799	476	0.802	15.969	258	0.320
<i>Aldebaran</i>	10	17.518	296	0.905	16.231	240	0.320
<i>Aldebaran</i>	13	16.361	290	1.033	15.915	132	0.867
<i>Aldebaran</i>	16	18.486	415	0.911	17.939	309	0.367
<i>Aldebaran</i>	17	17.743	1392	1.745	17.018	240	0.417
<i>Aldebaran</i>	18	18.021	495	0.715	17.150	235	0.600
<i>Aldebaran</i>	19	16.158	508	0.689	15.589	107	0.779
<i>Aldebaran</i>	20	16.901	110	0.776	16.014	116	0.493
<i>Aldebaran</i>	22	18.130	302	0.485	17.682	153	0.987
<i>Aldebaran</i>	26	18.399	177	0.882	17.411	250	0.431
<i>Aldebaran</i>	27	17.511	173	0.896	16.172	190	0.395
<i>Aldebaran</i>	29	17.135	230	0.984	15.852	246	0.945
<i>Aldebaran</i>	34	16.332	481	0.479	15.475	308	0.452
<i>Aldebaran</i>	35	16.585	422	0.947	15.729	168	1.016
<i>Aldebaran</i>	39	16.552	446	0.912	16.144	110	0.430
<i>Aldebaran</i>	45	17.323	376	1.104	16.271	285	0.317
<i>Aldebaran</i>	46	15.718	456	0.784	14.961	275	0.330
<i>Aldebaran</i>	47	15.238	488	0.569	14.732	119	0.384

Appendix 2. Continued.

Vessel	Haul	Mean Marport net spread	Number of pings	Standard deviation Marport net spread	Mean Netmind net spread	Number of pings	Standard deviation Netmind net spread
<i>Aldebaran</i>	49	17.698	367	1.125	16.926	171	0.314
<i>Aldebaran</i>	50	18.438	79	1.322	17.753	99	0.378
<i>Aldebaran</i>	51	17.625	221	1.564	16.981	291	0.286
<i>Aldebaran</i>	52	18.312	175	1.085	17.428	241	0.300
<i>Aldebaran</i>	53	17.552	430	0.620	16.992	250	0.360
<i>Aldebaran</i>	56	16.850	445	0.602	16.232	199	0.341
<i>Aldebaran</i>	57	16.582	343	0.876	15.639	228	0.326
<i>Aldebaran</i>	64	15.866	313	0.879	15.027	277	0.483
<i>Aldebaran</i>	65	16.413	1352	0.870	15.861	77	0.321
<i>Aldebaran</i>	67	16.114	354	0.974	15.390	38	0.569
<i>Aldebaran</i>	68	15.988	223	1.634	14.736	179	0.337
<i>Aldebaran</i>	69	16.243	298	1.017	15.042	175	0.389
<i>Aldebaran</i>	71	15.689	425	0.952	14.564	208	0.378
<i>Aldebaran</i>	75	17.152	465	0.615	16.589	60	0.328
<i>Aldebaran</i>	78	17.896	341	0.930	16.987	188	0.422
<i>Aldebaran</i>	79	17.947	336	0.814	17.176	311	0.293
<i>Aldebaran</i>	80	17.704	116	0.810	17.229	285	0.364
<i>Aldebaran</i>	81	20.344	1708	1.593	19.316	264	1.307
<i>Aldebaran</i>	83	19.462	163	0.759	18.324	309	0.867
<i>Aldebaran</i>	84	19.457	238	0.997	18.222	261	0.895
<i>Aldebaran</i>	88	18.702	273	0.772	17.981	308	0.347
<i>Aldebaran</i>	89	18.312	365	0.563	17.286	273	0.397
<i>Aldebaran</i>	91	17.278	333	0.770	16.319	243	0.369
<i>Aldebaran</i>	92	17.541	183	0.787	16.609	215	0.468
<i>Aldebaran</i>	93	18.167	145	0.917	17.124	223	0.373
<i>Aldebaran</i>	94	17.778	232	0.628	16.862	264	0.501
<i>Aldebaran</i>	97	15.910	398	0.790	15.155	134	0.639
<i>Aldebaran</i>	98	16.469	442	0.605	15.940	147	0.590
<i>Aldebaran</i>	100	15.885	473	0.612	15.029	51	0.699
<i>Aldebaran</i>	103	15.558	495	0.652	15.150	108	0.510
<i>Aldebaran</i>	107	17.050	236	0.777	16.110	277	0.700
<i>Aldebaran</i>	108	16.108	479	0.624	15.114	280	0.463
<i>Aldebaran</i>	109	15.953	485	0.925	15.559	165	0.509
<i>Aldebaran</i>	110	14.943	476	0.758	14.399	169	0.461
<i>Aldebaran</i>	112	16.312	380	0.896	15.273	300	0.291
<i>Aldebaran</i>	113	16.417	414	0.702	16.157	64	0.604
<i>Aldebaran</i>	117	16.903	207	0.608	15.924	282	0.628
<i>Aldebaran</i>	120	15.753	505	0.475	14.992	287	0.451
<i>Aldebaran</i>	121	16.192	424	0.743	15.326	293	0.532
<i>Aldebaran</i>	122	16.765	181	0.779	15.525	298	0.304
<i>Aldebaran</i>	123	15.461	473	0.868	14.906	68	0.291
<i>Aldebaran</i>	125	17.259	511	0.732	16.679	247	0.341
<i>Aldebaran</i>	126	17.585	236	1.558	16.968	310	0.596
<i>Aldebaran</i>	127	18.282	339	0.700	17.350	176	0.714
<i>Aldebaran</i>	128	18.312	321	0.689	17.759	154	0.592
<i>Aldebaran</i>	129	17.022	423	0.832	15.972	232	0.674
<i>Aldebaran</i>	130	18.001	352	0.767	17.341	244	0.413

Appendix 2. Continued.

Vessel	Haul	Mean	Number of pings	Standard deviation	Mean	Number of pings	Standard deviation
		Marport net spread		Marport net spread	Netmind net spread		Netmind net spread
<i>Aldebaran</i>	131	17.109	496	0.495	16.578	133	0.725
<i>Aldebaran</i>	132	16.746	453	0.728	16.111	289	0.398
<i>Aldebaran</i>	133	16.303	185	1.331	15.141	130	0.601
<i>Aldebaran</i>	135	17.352	261	0.799	16.481	266	0.537
<i>Aldebaran</i>	137	17.429	442	1.029	16.510	308	0.324
<i>Aldebaran</i>	138	19.187	348	0.753	18.600	295	0.442
<i>Aldebaran</i>	140	18.100	325	0.845	17.190	247	0.323
<i>Aldebaran</i>	141	17.806	491	0.812	17.196	166	0.502
<i>Aldebaran</i>	142	18.244	447	0.958	17.432	292	0.274
<i>Aldebaran</i>	143	17.410	468	0.968	16.517	282	0.324
<i>Aldebaran</i>	144	18.259	424	1.234	17.676	244	0.649
<i>Aldebaran</i>	145	17.871	364	1.186	16.770	282	0.308
<i>Aldebaran</i>	146	16.746	421	0.992	15.933	276	0.458
<i>Aldebaran</i>	147	18.684	408	0.791	17.915	291	0.281
<i>Aldebaran</i>	148	17.403	446	0.772	16.656	191	0.347
<i>Aldebaran</i>	149	17.903	371	0.937	17.270	249	0.540
<i>Aldebaran</i>	150	19.279	438	0.629	18.647	280	0.455
<i>Aldebaran</i>	153	16.968	126	1.885	16.114	124	0.348
<i>Aldebaran</i>	154	18.596	430	0.965	17.860	299	1.025
<i>Aldebaran</i>	155	18.943	234	0.719	17.744	147	0.675
<i>Aldebaran</i>	157	21.621	352	1.052	20.360	226	0.796
<i>Aldebaran</i>	158	17.904	482	0.568	17.090	300	0.292
<i>Aldebaran</i>	159	22.783	116	0.756	21.635	106	0.787
<i>Aldebaran</i>	160	17.973	445	0.723	17.116	228	0.281
<i>Aldebaran</i>	161	18.110	352	0.608	17.347	255	0.365
<i>Aldebaran</i>	162	18.082	492	0.378	17.229	315	0.296
<i>Aldebaran</i>	163	18.608	484	0.606	17.661	307	0.318
<i>Aldebaran</i>	164	18.242	371	0.680	17.317	291	0.365
<i>Aldebaran</i>	165	18.313	418	0.926	17.386	274	0.288
<i>Aldebaran</i>	166	20.048	245	0.695	19.178	300	0.279
<i>Aldebaran</i>	167	19.073	301	0.452	18.462	202	0.631
<i>Aldebaran</i>	168	19.484	279	0.613	18.313	309	0.413
<i>Aldebaran</i>	169	19.507	374	0.536	18.372	258	0.338
<i>Aldebaran</i>	170	19.617	391	0.537	18.608	271	0.550
<i>Aldebaran</i>	171	19.765	348	0.488	18.583	326	0.382
<i>Aldebaran</i>	173	19.405	498	0.391	18.369	252	0.648
<i>Aldebaran</i>	174	19.169	443	0.512	18.055	317	0.324
<i>Aldebaran</i>	175	18.551	486	0.474	17.922	296	0.365
<i>Aldebaran</i>	176	19.120	287	0.573	18.043	318	0.351
<i>Aldebaran</i>	177	19.934	310	0.602	18.927	335	0.342
<i>Aldebaran</i>	179	18.709	169	0.306	17.806	114	0.266
<i>Aldebaran</i>	180	19.319	416	0.511	18.372	328	0.345
<i>Aldebaran</i>	181	18.731	181	0.407	17.796	99	0.878
<i>Aldebaran</i>	182	19.271	485	0.455	18.260	184	1.292
<i>Aldebaran</i>	184	19.360	393	0.539	18.510	202	1.183
<i>Aldebaran</i>	185	19.354	418	0.514	18.328	333	0.349
<i>Aldebaran</i>	186	20.044	437	0.699	19.269	234	0.816

Appendix 2. Continued.

Vessel	Haul	Mean Marport net spread	Number of pings	Standard deviation Marport net spread	Mean Netmind net spread	Number of pings	Standard deviation Netmind net spread
<i>Aldebaran</i>	188	20.084	213	0.617	19.028	287	0.516
<i>Aldebaran</i>	189	20.267	459	0.359	19.513	309	0.431
<i>Aldebaran</i>	190	19.801	194	0.535	18.780	314	0.347
<i>Aldebaran</i>	191	19.325	152	0.582	18.223	333	0.290
<i>Aldebaran</i>	192	19.283	391	0.422	18.269	322	0.277
<i>Aldebaran</i>	193	19.518	359	0.451	18.644	325	0.283
<i>Aldebaran</i>	196	19.146	475	0.460	18.527	98	1.859
<i>Aldebaran</i>	197	19.793	455	0.309	18.850	156	0.853
<i>Aldebaran</i>	198	19.281	516	1.088	18.282	229	1.025
<i>Aldebaran</i>	199	17.686	526	0.528	16.844	136	1.265
<i>Aldebaran</i>	200	16.896	432	0.648	16.720	96	1.329
<i>Aldebaran</i>	201	16.531	476	0.566	15.694	144	1.224
<i>Aldebaran</i>	202	19.756	204	0.757	18.938	253	0.689
<i>Aldebaran</i>	204	17.662	198	0.551	17.206	250	0.641
<i>Aldebaran</i>	205	18.055	439	0.723	17.657	103	1.613
<i>Aldebaran</i>	206	19.608	239	1.196	18.305	96	1.770
<i>Aldebaran</i>	207	17.960	500	0.372	17.396	313	0.430
<i>Aldebaran</i>	208	18.012	342	0.660	17.090	302	0.393
<i>Aldebaran</i>	209	18.124	283	0.677	17.373	313	0.379
<i>Aldebaran</i>	210	18.036	494	0.315	17.344	317	0.220
<i>Aldebaran</i>	211	17.343	453	0.418	16.838	109	0.505
<i>Aldebaran</i>	212	18.513	130	0.569	17.228	297	0.339
<i>Aldebaran</i>	213	18.452	544	0.437	17.819	270	0.570
<i>Aldebaran</i>	214	18.813	463	0.527	17.771	324	0.283