

Final Technical Report of Grant ONR N00014-03-1-00465

**Entitled:
Computational Methods for Target-Tracking Problems**

Submitted by

Anthony G. Warrack, PI

Alexandra Kurepa

Department of Mathematics

North Carolina A&T State University

1601 East Market Street, Greensboro NC 27411

warrack@ncat.edu (336) 285-2092 kurepa@ncat.edu

to

Mathematical, Computer and Information Sciences, Code 311

Office of Naval Research, 875 North Randolph St, Suite 1118

Arlington, VA 22203-1995

Rabinder N. Madan, Ph.D., Technical Director (madanr@onr.navy.mil)

(703) [696-4217](tel:7036964217)

September 2007

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 28-09-2007		2. REPORT TYPE Final Technical Report		3. DATES COVERED (From - To) 3/1/2003-6/30/2007	
4. TITLE AND SUBTITLE Computational Methods for Probabilistic Target Tracking Problems				5a. CONTRACT NUMBER N/A	
				5b. GRANT NUMBER N00014-03-1-00465	
				5c. PROGRAM ELEMENT NUMBER N/A	
6. AUTHOR(S) Warrack, Anthony, G				5d. PROJECT NUMBER N/A	
				5e. TASK NUMBER N/A	
				5f. WORK UNIT NUMBER N/A	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NORTH CAROLINA A&T STATE UNIVERSITY 1601 E MARKET ST GREENSBORO NC 27411				8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) DR. RABINDER MADAN (CODE 311) OFFICE OF NAVAL RESEARCH 875 North Randolph St, Suite 1118 Arlington, VA 22203-1995				10. SPONSOR/MONITOR'S ACRONYM(S) ONR	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) N/A	
12. DISTRIBUTION / AVAILABILITY STATEMENT UU					
13. SUPPLEMENTARY NOTES None					
14. ABSTRACT The grant started in 2003. Initially a cohort of two graduate students and four sophomore undergraduate students was recruited. The students received special training in probabilistic and statistical methods pertaining to target -tracking problems. Particular topics included Kalman filtering, the EM Algorithm, smoothing methods, and density estimation. In the summer of 2004 the graduate students accompanied Dr. Warrack to The Naval Undersea Warfare Center, Newport, RI (NUWC-Newport), for a 10 week internship working under the supervision of Dr Roy Streit. This resulted in a presentation at NUWC, <i>Applying Density Estimation and Nonparametric Smoothing Techniques to Tracking Problems</i> . In the summer of 2005 the undergraduates accompanied Dr. Warrack to NUWC-Newport for a 10 week internship under the direction of Dr. Marcus Graham. A presentation <i>“Using Parametric and Nonparametric Smoothing Techniques to Improve Estimation with the EM Algorithm</i> was given at NUWC. All six of the students have graduated with high grade point averages. Three received NAVSEA job offers, one of whom is working at NSWCDD-Dahgren, VA. During an extension year two graduate students and two undergraduates were supported.					
15. SUBJECT TERMS Density Estimation, Kalman Filter, EM Algorithm, Linear Regression, Nonparametric Regression, Smoothing, Gaussian Mixtures, Bayesian Statistics.					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 45	19a. NAME OF RESPONSIBLE PERSON Dr. A.G. Warrack
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (include area code) 336-285-2092

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998-, xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

Ba. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER.

Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES).

Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S).

Enter report number as assigned by the sponsoring/monitoring agency, if available, e.g. BRL-TR-829; -21 5.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

Table of Contents

Participants	3
Activities and Findings	4
Research Summary	4
Research Presentations	5
Library of R/S-Plus Programs	6
Education and Results	6
Initial Training	6
Graduate Student Research at NUWC	7
Undergraduate Student Research at NUWC	7
Conclusions Regarding NUWC Internships	8
Further Graduate Student Research at NC A&T	9
Effect on Career and Professional Development	9
Demographic Data	10
APPENDIX A	
R/S-Plus Kalman Filter Program	11
EM Algorithm for Normal Mixture Estimation	13
Program for Unscented Kalman Filter	16
APPENDIX B	
Graduate Student Presentation NUWC 2004	20
APPENDIX C	
Undergraduate Student Presentation NUWC 2005	37

Participants

The following personnel were the main participants in this grant:

Principal Investigator: Dr. Anthony Giles Warrack

Co-PI: Dr. Alexandra Kurepa

ONR Technical Director: Dr. Rabinder Madan

NUWC-Newport Points of Contact: Dr. Roy L. Streit, Dr. Marcus L. Graham

Graduate Students: Ms. Latoya Silochan, Ms. Kashonda Bynum, Mr. Rodolfo Bernal, Ms. Alisha Williams

Undergraduate Students: Ms. Angela Edwards, Mr. Bryahn Ivery, Mr. Dustin Lupton, Mr. James Pender, Mr. Terrell Felder, Ms. Krystal Knight

Under the terms of the original proposal, submitted in 2003, a cohort of two graduate students (Silochan and Bynum) and four undergraduate sophomore students (Edwards, Lupton, Ivery, and Pender) was selected. Due to the fact that some funding was delayed, and also that there were extra funds available as in-state rather than out-of-state students had been recruited, a no-cost one year extension was applied for, and granted, enabling support for two more graduate students, Mr. Ricardo Bernal and Ms Alisha Williams, and two more undergraduate students, Ms Krystal Knight and Mr. Terrell Felder. Thus the grant has supported 6 undergraduate and 4 graduate students.

The grant was undertaken with close cooperation at The Naval Undersea Warfare Center, Newport, RI (NUWC-Newport) where Dr. Warrack had held ONR/ASEE Summer Faculty Research Fellowships. On March 30, 2004 Dr. Rabinder Madan, the ONR Technical Director for the grant with Dr. Roy L. Streit of NUWC, visited the A&T campus to confer with the PI's and the students, prior to the two graduate students summer internship at NUWC. When Dr. Streit resigned from NUWC-Newport in 2005,

Dr. Marcus L. Graham became the point of contact. Dr. Errol G. Rowe at NUWC also cooperated.

Activities and Findings

1. Research Summary

The research has concentrated in the area of parameter estimation in mixtures of normal probability distributions, in particular as it pertains to problems of multi-target tracking. We have been involved in the implementation of algorithms and estimation methods, such as the E-M Algorithm, the Kalman filter, and smoothing methods, both parametric and non-parametric. We have also investigated the relative merits of various statistics in evaluating statistical models, such as the Akaike Information Criterion (AIC), and the Bayes Information Criterion (BIC), particularly in situations where the number of populations (targets) is one of the parameters to be estimated. Another area of interest has been in the field of obtaining good initial “guesses” of parameters, where we have compared various clustering techniques and algorithms.

Another problem of interest has been that of maintaining “target identity”, when two target trajectories either cross or pass very close to each other. We attempted to combine smoothing methods with the EM Algorithm, and were interested in comparing standard methods, e.g. linear or polynomial regression, with methods that make fewer model assumptions, such as smoothing splines, Loess regression, and Kernel Regression. In the later case we were interested in comparing the combination of different kernel functions combined with different bandwidths.

2. Research Presentations

At all stages the students have been encouraged to present their work on a formal basis. The following is a list of presentations that have been made by both the faculty and student participants in the grant:

"Applying the Unscented Kalman Filter to Problems in Submarine Tracking"

Dr. Giles Warrack and Dr Alexandra Kurepa, The Third National Ronald E. McNair Symposium on Science and Technology : North Carolina Agricultural and Technical State University, January 28, 2004

"Simulating a Target Tracking Problem Using the Kalman Filter" ,Latoya Silochan, Math Awareness Mini-Conference, North Carolina Agricultural and Technical State University

April 22, 2004

"Searching Algorithms using the Kendall-Wei Algorithm", Kashonda Bynum, Math Awareness Mini-Conference, North Carolina Agricultural and Technical State University

April 22, 2004

"Applying Density Estimation and Nonparametric Smoothing Techniques to Tracking Problems" Latoya Silochan and Kashonda Bynum, presentation to NUWC-Newport Code 22 members, July 23, 2004

"Multiple Target Tracking Using Functional Density Estimation", Kashonda Bynum, Latoya Silochan, The Fourth National Ronald E. McNair Symposium on Science and Technology : North Carolina Agricultural and Technical State University, January 29, 2005

“Using Parametric and Nonparametric Smoothing Techniques to Improve Estimation with the EM Algorithm”, Angela Edwards, Dustin Lupton, Bryahn Ivery, and James Pender. Presentation given at Chaffee Auditorium, NUWC-Newport, July 22, 2005

“Submarine Target Tracking Simulations Using Mathematical Modelling”, Dustin Lupton and James Pender. Math Awareness Mini-Conference, North Carolina Agricultural and Technical State University, April 24, 2006

“Using Tree Based Methods to Classify Messages”, Terrell A. Felder, Math Awareness Mini-Conference, North Carolina Agricultural and Technical State University, April 19, 2007

“Applying Logistic regression to Message Classification”, Krystal A. Knight, Math Awareness Mini-Conference, North Carolina Agricultural and Technical State University, April 19, 2006

3). Library of R/S-Plus Programs

During the course of the research a collection of computer programs for the E-M Algorithm, the Kalman Filter, and the Unscented Kalman Filter were written, many in collaboration with Dr. Errol G. Rowe of NUWC-Newport. These are included in Appendix A.

Education and Results

1). Initial Training

All students in the initial cohort were given special instruction through a series of lectures and seminars, as well as being required to take courses in Probability, Linear Models, and Statistical Inference. The lectures and seminars covered more specialized topics than would generally be included in standard courses. The topics included

Bayesian Estimation, Filtering methods, Nonparametric Regression, Monte-Carlo simulation, Bootstrapping, Maximum Likelihood Estimation using the E-M Algorithm, and Kernel Density Estimation. The students were also taught to program in MATLAB and/or R/S-Plus.

2). Graduate Student Research at NUWC-Newport

In the summer of 2004 the two graduate students, Latoya Silochan and Kashonda Bynum, accompanied Dr. Warrack on a 10 week internship to NUWC-Newport, where they worked under the supervision of Dr. Warrack, and Dr. Roy L. Streit (Code 22). The students worked on a project involving the application of functional density estimation techniques to multiple target tracking. The work combined Bayesian updating estimation over time, with comparisons of different kernels (e.g. Gaussian, Epachnikov), in density estimation, and different choices of bandwidth (the “Bias-Variance” tradeoff). At the end of the 10 week period the students made a presentation to members of Code 22 at NUWC-Newport entitled “*Applying Density Estimation and Nonparametric Smoothing Techniques to Tracking Problems*”. The students attended seminars and lectures given at NUWC. They also sat in on ILIR sessions, in which NUWC researchers made presentations for the process of in-house research funding.

3). Undergraduate Student Research at NUWC-Newport

In the summer of 2005 the four undergraduate students, Angela Edwards, Dustin Lupton, Bryahn Ivery, and James Pender accompanied Dr. Warrack for a 10 week internship at NUWC-Newport. The NUWC point of contact was Dr. Marcus L. Graham (Dr. Streit having left NUWC). Again the students attended in-house NUWC lectures and presentations on non-classified material and ILIR sessions. They collaborated on a project using simulated data and along with all the other student interns made a presentation in the Chaffee Auditorium at NUWC entitled “*Using Parametric and*

Nonparametric Smoothing Techniques to Improve Estimation with the EM Algorithm". Using simulated data, the students attempted to track varying numbers of targets using the E-M (Expectation-Maximisation) Algorithm, both when the numbers of targets are known, and when they are unknown. One problem addressed was that of attempting to use various types of smoothing routines to maintain target identity when target trajectories either cross, or pass very close to each other. As well as trying standard methods, such as linear or polynomial regression, they also experimented with more "model free" methods such as cubic splines, Loess regression, and kernel regression (comparing different kernels and bandwidths). An attempt was made to tackle the much more difficult problem of tracking when the number of targets varies and is one of the parameters to be estimated. This problem was treated by a "penalized likelihood" approach which compensates for the fact that more complicated models will have higher likelihoods than smaller sub-models by penalizing models with larger numbers of parameters. Two standard statistics that do this are Akaike Information Criterion (AIC), and the Bayes Information Criterion (BIC). In so far as it was possible to compare the two, empirical evidence based on simulated data seemed to indicate that the BIC was marginally superior, in others words it had a slightly higher probability of selecting the correct model. It also has the attractive feature that it can be shown to be the posterior probability for a model, given the data.

4). Conclusions Regarding NUWC Internships

All six students are on record as saying that they believe they benefited enormously from the summer internships. They were very impressed by the professionalism of the staff at NUWC, and felt they had been very well treated there. It certainly gave them exposure to a professional working environment. They also developed a certain intellectual initiative, and the ability to work on problems on their own for extended periods.

5). Further Graduate Student Research at NC A&T

On returning to North Carolina A&T from NUWC, Latoya Silochan wrote a Master's Degree project under the supervision of Dr. Warrack entitled "*An E-M Based Algorithm for Maintaining Target Identity*", in which she combined parametric polynomial smoothing with the E-M Algorithm. This was presented in April 2005. Ms Bynum wrote an MS project under the supervision of Dr. Bolindrah Borah entitled "*Solution Methods of Non-homogeneous Partial Differential Equations*". Rodolfo Bernal did a project under the supervision of Dr. Warrack, "*Improving Estimation in the E-M Algorithm by PAVA Smoothing*". In this he considered the estimation of the parameters of a normal mixture distribution when the mixing probabilities, and the means of the mixtures are known to have the same orderings, e.g. $\mu_1 \leq \mu_2 \leq \dots \leq \mu_k$, and $p_1 \leq p_2 \leq \dots \leq p_k$. He attempted to incorporate the techniques of Isotonic Regression using the "Pool Adjacent Violators Algorithm" (PAVA) into the estimation process. While the estimation appeared to me marginally improved, the resulting algorithm was considerably slower. Ms. Alisha Williams is currently working with Dr. Kurepa on applications of Monte Carlo methods to Partial Differential Equations.

Effect on Career and Professional Development

Both Ms Silochan and Ms Bynum graduated with MS degrees in May 2005. Ms Silochan applied for a position at the Naval Surface Warfare Center (NSWCDD) , Dahlgren, VA She was offered, and accepted a position as a Mathematician/Statistician, which she accepted. However she was unable to obtain the required security clearance, and so accepted a position as a Mathematical Statistician with the US Postal Service, where she now works. Ms Bynum was also offered a position with NSWCDD but she too was unable to obtain security clearance. She is working as a lecturer and academic counselor at the Center for Academic Excellence at North Carolina A&T. Mr Rodolfo Bernal graduated with an MS degree in May 2006. He was offered a job as Systems Engineer at NSWC, but was then told this would have to be put on hold because of a hiring freeze.

He has also applied for a position with the US Census Bureau. He is currently working for Your Choice Health Services in Raleigh, NC.

James Pender, Dustin Lupton, Bryahn Ivery, and Angela Edwards all graduated in 2006 with BS (Mathematics) degrees. James Pender has been working at NSWCDD since January 2007. At the moment he is working with the Aegis Ballistic Missile Defense System (ABMD) in the Command and Decision (C&D) section. He has recently been selected from a competitive pool of candidates for a position doing baseline testing for the Japan Ballistic Missile Defense (JABMD) System. Dustin Lupton is employed by Progress Energy as an Auxiliary Operator at the Brunswick Nuclear Plant, in Southport NC. He is studying to qualify as an NRC licensed nuclear reactor operator. Bryahn Ivery, after teaching for a year has also applied to NSWCDD, also the NSA and the Census Bureau. Angela Edwards has had health problems which has made permanent employment difficult.

There is no doubt that the grant significantly impacted the career choices of most of the students involved, as well as giving them exposure to areas of applied mathematics, probability and statistics they would not otherwise have encountered.

Demographic Data

Of the four original undergraduates, all of whom graduated with GPA's between 3.00 and 3.75, three were African American, and one White. Three were male, and one female. Of the three graduate students who graduated, two were African American, and one Hispanic. Of the students currently enrolled, the graduate student is female and African American, the two undergraduates are both African American, one male, the other female. All three are performing excellently in their classes.

APPENDIX A

```
#Kalman Filter R/S-Plus program for 2-Dimensional Tracking
#Instructions: To load the file into R's workspace, type
#           source("/home/... path to your code ... /kalman.R")
#
#           To run the program, type
#
#           (a) library(MASS) #Just once per R session.
#                   #contains ginv(generalized matrix inverse)
#
#           (b) kalmanR(N)  #N is number of observations.
#
#
#library(MASS) #contains ginv - generalized matrix inverse:
kalmanR <- function(N) {
  # N - the number of observations.
  #
  m <- numeric; m <- 2;
  n <- numeric; n <- 4;
  dt <- numeric; dt <- 1;

  accel <- 0.5;
  obsStd <- .75;

  xHat <- numeric(n);
  xHat <- c(0.0, 0.0, 0.0, 0.0); #(x_pos, x_velocity, y_pos, y_velocity)

  Path <- matrix(0,2,N)
  path_Hat <- matrix(0,2,N)

  Soln <- numeric(n);
  Soln <- xHat;

  Phi <- diag(n); #Identity 4-by-4 matrix:
  Phi[1,2] <- dt; #x-component update
  Phi[3,4] <- dt; #y-component update

  P <- rnorm(n*n);
  dim(P) <- c(n,n);

  Q <- diag(n);
  Q[1,1] <- dt^4 / 4; Q[1,2] <- dt^3 / 2; Q[2,1] <- dt^3 / 2; Q[2,2] <- dt^2;
  Q[3,3] <- dt^4 / 4; Q[3,4] <- dt^3 / 2; Q[4,3] <- dt^3 / 2; Q[4,4] <- dt^2;
  Q <- accel^2 * Q;
  P <- Q;
```

```

M <- matrix(0,m,n);
M[1,1] <- 1.0;
M[2,3] <- 1.0;

R <- obsStd * obsStd * diag(m);

Phi_P <- matrix(0,n,n);
Phi_P_Mprime <- matrix(0,n,m);
B_Mprime_plusR <- matrix(0,m,m);
MP <- matrix(0,m,n);

for (i in 1:N) {
  eps1 <- rnorm(1);
  eps2 <- rnorm(1);
  processNoise = accel * c(eps1*dt^2 / 2, eps1*dt, eps2*dt^2 / 2, eps2*dt);
  Soln <- Phi %*% Soln + processNoise;

  w <- obsStd * rnorm(m);
  z <- M %*% Soln + w; #This is our observation:
  Innovation <- z - M %*% xHat; #Difference between approximation & observation:

  B_Mprime_plusR <- M %*% P %*% t(M) + R;
  w <- solve(B_Mprime_plusR,Innovation);
  xHat <- Phi %*% xHat + Phi %*% P %*% t(M) %*% w;

  Path[1,i] <- Soln[1]; Path[2,i] <- Soln[3]
  path_Hat[1,i] <- xHat[1]; path_Hat[2,i] <- xHat[3]

  P <- Phi %*% P %*% t(Phi) - Phi %*% P %*% t(M) %*%
    ginv(B_Mprime_plusR) %*% M %*% P %*% t(Phi) + Q
}
results <- data.frame(Path[1,],Path[2,],path_Hat[1,],path_Hat[2,])
plot(results[,1],results[,2],col='blue',type='l',xlab="Truth = blue, approx = red",ylab=" ")
points(results[,3],results[,4],col='red',type='l')
results
}

```

R/S/Plus Program For Estimation of Parameters in k Gaussian Mixtures

```
#####
***##
## Program "EMkMixEstimation.R" . This programe contains 3 subroutines:  ##
## 1). KmixGenerate generates a mixture of k normals, sample size n      ##
## 2). StartMixK uses sample quantiles to generate startinf means, sds    ##
## 3). emK computes EM estimates of mu's sd's probs for any k           ##
#####
#####
***##
## Subroutine to Generate a sample of size n of a mixture of k normals  ##
#####
KmixGenerate <- function(n,means,sds,probs) {
  k    <- length(means)
  nk   <- rmultinom(1,n,probs) ##USE MULTINOMIAL TO GENERATE
MIXTURES
  x    <- numeric(0)
  for (i in 1:k) {
    x  <- c(x,rnorm(nk[i],means[i],sds[i]))
    print(nk[i])
  }
  return(x)
}

#####
## Now Generate Starting Values  ##
#####

StartMixK  <- function(x,k) {
  y    <- sort(x)
  means1 <- numeric(k)
  sds1  <- numeric(k)

  ## Uniform prior probs assumed, but any could be used  ##

  probs1 <- rep(1,k)/k
  breaks <- c(0,cumsum(probs1))

  ## Computes required number of quantiles ##

  quants <- quantile(x,breaks)

  ## This loop computes the starting means and std devs ##

  for (i in 1:k) {
    a    <- quants[i]
```

```

        b      <- quants[i+1]
        means1[i] <- mean(y[y >= a & y <= b])
        sds1[i]  <- sd(y[y >= a & y <= b])
    }
    return(list(means1,sds1,probs1))
}

#####
***##
##          Routine to do EM estimation          ##
##          Compute Estimates using EM with function emK          ##
#####
***##

emK<-function(x,means,sds,probs) {
    n      <- length(x)
    k      <- length(means)
    MAXiter<-500 #Maximum number of iterations
    numITS <- 0; ERR <- 1

    ## Create n by k matrix for posterior probabilities ##

    TX      <-matrix(0,n,k)

    while ((ERR > .00005) & (numITS < MAXiter)) {
        numITS  <- numITS + 1
        oldmeans <- means

#####
***##
## Compute column numerators for TX, n by k posterior probabilities matrix ##
#####
***##

        for (i in 1:k) {
            TX[,i] <- probs[i]*dnorm(x,means[i],sds[i])
        }

        TXrowsum      <- apply(TX,1,sum)

    ## Now divide by row sums ##

    TX      <-TX/TXrowsum

```

```

##Update probabilities, means, sds##

      for (i in 1:k) {
        probs[i] <-mean(TX[,i])
        means[i] <-sum(TX[,i]*x)/(n*probs[i])
        sds[i]  <-sqrt(sum(TX[,i]*(x-means[i])^2)/(n*probs[i]))
      }

      ERR      <- sum((oldmeans-means)^2)
    }

    print("Number of Iterations, Convergence Error");print(c(numITS,ERR))
    return(list(means,sds,probs)) ## Evidently R not happy about returning
values this way ##
}

#####
***##
##                               End of function  emK                               ##
#####
#####
***##
##                               MAIN PROGRAM                               ##
#####
***##

n <- 200
means<-c(5,10,12,15)
sds<-c(2,4,4,2)
probs<-c(.3,.2,.3,.2)
k <- length(means)
x <- KmixGenerate(n,means,sds,probs)
#print(x)
startvals <- StartMixK(x,k)
EMestimates <- emK(x,startvals[[1]],startvals[[2]],startvals[[3]])
print("TRUE MEANS, SDS, & PROBS WITH EM ESTIMATES")
print(means)
print(estmeans<-EMestimates[[1]])
print(sds)
print(estsds<-EMestimates[[2]])
print(probs)
print(estprobs<-EMestimates[[3]])

```

R/S/Plus Program to Track with Unscented Kalman Filter

```
#
#
#
ukf <- function(N,x0=10,alpha=0.5,beta=25,gamma=8,sd1=1.732051,std2=1.0) {
  #
  #N The number of time samples.
  #
  #
  #To Load into Memory: source("/home/...path-to-program.../ukf.R")
  #
  #To Run: ukf(100)

  #Generate state and observation values: MM[,1] and MM[,3], respectively.
  #State and observation noise also generated: MM[,2] and MM[,4], respectively.
  MM <- processesTruth(alpha, beta, gamma, sd1, sd2, x0, N)

  lx <- 1 #Length of process x
  lv <- 1 #Length of noise v

  ly <- 1 #Length of observation process
  ln <- 1 #Length of observation noise n

  X_x <- matrix(0,lx,2*lx+1)
  X_v <- matrix(0,lv,2*lx+1)
  X_n <- matrix(0,ln,2*lx+1)

  x <- MM[1,1]
  y <- MM[1,3]
  v <- MM[1,2]
  n <- MM[1,4]

  myPts <- x

  P_k <- (sd1*sd1)*diag(lx)

  W <- numeric(2*lx + 1)

  X <- matrix(0,lx,2*lx+1)
  Y <- matrix(0,ly,2*lx+1)
  K <- matrix(0,lx,ly)

  W[1] <- .5
  for (i in 2:(2*lx + 1)) { #Weights
```

```

W[i] <- (1 - W[1])/(2*lx)
}

for (kk in 1:N) {
  #
  #

  X[,1] <- x
  j <- 0
  for (i in 2:(2*lx+1)) {
    j <- j + 1
    X[,i] <- x + (-1)**(j+1) * sqrt(1/(2*W[i])*P_k)
  }

  for (i in 1:(2*lx + 1)) {
    X_x[1:lx,i] <- X[1:lx,i]
  }

  X_x <- systemProcess_noNoise(X_x, lx, kk, alpha, beta, gamma)

  x_mean <- matrix(0, lx, 1)
  dim(x_mean) <- c(length(x_mean), 1)
  for (i in 1:(2*lx + 1)) {
    x_mean <- x_mean + W[i]*X_x[1:lx,i]
  }

  P_kMean <- sd1*sd1 * diag(lx)
  for (i in 1:(2*lx + 1)) {
    P_kMean <- P_kMean + W[i] * (X_x[1:lx,i] - x_mean) %*% t(X_x[1:lx,i] -
x_mean)
  }

  Y <- observationProcess_noNoise(Y, X_x, lx, ly, alpha, beta, gamma)

  y_mean <- matrix(0, ly, 1)
  dim(y_mean) <- c(length(y_mean), 1)
  for (i in 1:(2*lx + 1)) {
    y_mean <- y_mean + W[i]*Y[1:ly,i]
  }
}

```

```

P_yy <- sd2*sd2 * diag(ly)
for (i in 1:(2*lx + 1)) {
  P_yy <- P_yy + W[i] * (Y[1:ly,i] - y_mean) %*% t(Y[1:ly,i] - y_mean)
}

P_xy <- matrix(0, lx, ly)
for (i in 1:(2*lx + 1)) {
  P_xy <- P_xy + W[i] * (X_x[1:lx,i] - x_mean) %*% t(Y[1:ly,i] - y_mean)
}

K <- P_xy[1,1] / P_yy[1,1] #P_xy %*% ginv(P_yy)
x <- x_mean + K %*% (MM[kk,3] - y_mean)
P_k <- P_kMean - K %*% P_yy %*% t(K)

myPts <- c(myPts,x)
}

errMat <- numeric(N)
MSE <- 0
for (k in 1:N) {
  errMat[k] <- abs(MM[k,1] - myPts[k])
  MSE <- MSE + errMat[k]*errMat[k]
}
MSE <- MSE/N
print(MSE)
timeStamps <- 1:length(myPts)
plot(timeStamps, myPts, ylab="Average Particle Pos.", type="l")
}

processesTruth <- function(alpha, beta, gamma, sd1, sd2, x0, N) {
  #
  #
  #
  # M[,1] ... the state values:
  # M[,2] ... the state noise:
  #
  # M[,3] ... observation values:
  # M[,4] ... observation noise:
  #
  #
  M <- matrix(0,N,4)
  M[,2] <- rnorm(N,0,sd1) #Process Noise

```

```

M[,4] <- rnorm(N,0,sd2) #Observation Noise
#
M[1,1] <- x0
for (n in 2:N) {
  M[n,1] <- alpha * M[n-1,1] +
    beta * M[n-1,1]/(1 + M[n-1,1]*M[n-1,1]) +
    gamma * cos(1.2*n) + M[n,2] #Process
  M[n,3] <- M[n,1]*M[n,1]/20.0 + M[n,4] #Observation
}
timeStamps <- 1:N
#split.screen(c(2,1))
layout(matrix(c(1,2,3), 3, 1))
#screen(1)
#erase.screen()
plot(timeStamps, M[,1],ylab="True Particle Position",type='l')
return(M)
}

systemProcess_noNoise <- function(X_x,lx,kk,alpha,beta,gamma) {
#
#
for (n in 1:(2*lx + 1)) { #Process
  X_x[1:lx,n] <- alpha * X_x[1:lx,n] +
    beta * X_x[1:lx,n]/(1 + X_x[1:lx,n]*X_x[1:lx,n]) +
    gamma * cos(1.2*kk)
}
return(X_x)
}

observationProcess_noNoise <- function(Y,X_x,lx,ly,alpha,beta,gamma) {
#
#
for (n in 1:(2*lx + 1)) { #Observation
  Y[1:ly,n] <- X_x[1:lx,n]*X_x[1:lx,n]/20.0
}
return(Y)
}

#
#
# END OF CODE

```

APPENDIX B

Graduate Students Presentation at NUWC

In the summer of 2004 the two graduate students, Ms Latoya Silochan and Ms Kashonda Bynum accompanied Dr. Warrack for a 10 week internship at NUWC-Newport, where they worked under Dr Roy L. Streit. At the end of the internship the students made a presentation to selected member of Code 22 at NUWC entitled “Applying Density Estimation Techniques to Tracking Problems”. Frequency Azimuth (“FRAZ”) data was simulated, and various different kernel density estimators were used. The effect of different bandwidths was also considered.

Applying Density Estimation and Nonparametric Smoothing Techniques to Tracking Problems

A.G. Warrack, K. Bynum, and L. Silochan

July 23, 2004

1 Introduction

We consider tracking problems in which multiple targets are observed over time. At each time period beam measurements are observed, being the sum of measurements over frequency bins (FRAZ or Frequency Azimuth data). Data was simulated for two scenarios. In the first the number of targets is fixed at three, and the trajectories cross at the halfway point. In the second there are initially three targets, with a fourth entering at the halfway point. It is assumed that the number of targets is unknown, and that this is a parameter to be estimated, along with the target trajectories.

It is assumed that the data generated by k targets at time t follows a mixture of k Gaussian distributions

$$f(x; t) = \sum_{i=1}^k p_i \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}\left(\frac{x-\mu_i(t)}{\sigma_i}\right)^2} \quad (1)$$

If the number of mixtures, k is given, a well known algorithm for estimating the μ_i, σ_i , and p_i is the E-M (Expectation-Maximization) Algorithm. In this paper we attempt to see if k and the $\mu_i(t)$ can be estimated by smoothing the data using a kernel density estimate at each time point, and then smoothing the modal points by means of nonparametric smoothing (e.g. splines) to estimate the target trajectories. We also attempt to test hypotheses about the number of targets at any times point by using a bootstrap test based on the minimum bandwidth.

2 Density Estimation

There are two main components in the estimation of probability sdensity functions, the selection of a *kernel* function, and the selection of a bandwidth.

2.1 Kernel Functions

A *kernel density estimate* for a probability density function $f(x)$ is a function of the form

$$\hat{f}(x) = \frac{1}{nh} \sum_{j=1}^n K\left(\frac{x - X_j}{h}\right) \quad (2)$$

for a sample $X_1, X_2, \dots, X_n$, a fixed kernel function $K()$, and a bandwidth h . The kernel is normally chosen to be a probability density function with $\int xK(x)dx = 0$ and $\int x^2K(x)dx = \sigma_K^2 < \infty$. Two common choices for $K()$ are the Gaussian kernel

$$K(x) = (2\pi)^{-1/2} e^{-x^2/2}$$

and the Epanechnikov kernel

$$K(x) = \frac{3}{4} (1 - x^2/5) / \sqrt{5}, |x| \leq \sqrt{5} \quad (3)$$

$$K(x) = 0, |x| > \sqrt{5} \quad (4)$$

When using the Gaussian kernel one may think of the estimator in the words of Efron and Tibshirani as "adding up n little Gaussian density curves centered at each point x_i each having standard deviation h ". It is generally acknowledged that while the selection of a kernel function is not important in estimating a density, the selection of the bandwidth, h , is more crucial. If h is chosen too small, $\hat{f}(x)$ will have low bias, that is $E[\hat{f}(x)]$ will be close to $f(x)$, but high variance, but if h is chosen too large this situation will be reversed. Ideally h should be chosen to minimize the mean square error (MSE), which is the sum of the variance and the square of the bias. Two other noteworthy facts regarding the bandwidth are:

- 1). the number of modes, m , of $\hat{f}(x)$ is a monotonically decreasing function of h
- 2). If h_m is the smallest bandwidth producing a density estimator with m modes, then the log-likelihood $\sum_{i=1}^n \hat{f}(X_i; K, h_m)$ is maximized over all h producing density estimates with m modes

Another way of looking at h_m is to think of it as the bandwidth that produces m modes with the least amount of smoothing.

2.2 Bandwidth Selection

One way to automatically compute the bandwidth is based on computing the mean integrated square error (MISE) here the expectation is taken with regard to h)

$$MISE = E \int |\hat{f}(x; h) - f(x)|^2 dx = \int E |\hat{f}(x; h) - f(x)|^2 dx$$

and choosing the smallest value as a function of h . We can expand MISE as

$$MISE = E \int \hat{f}(x; h)^2 dx - 2E \hat{f}(X; h) + \int f(x)^2 dx$$

The third term is a constant and can be dropped.

One approach is to make an asymptotic expansion of the MISE of the form

$$MISE = \frac{1}{nh} \int K^2 + \frac{1}{4} h^4 \int (f'')^2 \left\{ \int x^2 K \right\}^2 + O(1/nh + h^4)$$

where K^2 is the convolution of the kernel function with itself. If we neglect the remainder, the optimal bandwidth, the one minimizing $MISE$, would be

$$h_{optimal} = \left[\frac{\int K^2}{n \int (f'') \left\{ \int x^2 K \right\}^2} \right]^{1/5} \quad (5)$$

Of course the function on the right hand side of (4) involves the unknown density function f . One solution to this problem, generally known as the "plug-in" method, is to choose a pilot bandwidth, k , and approximate f'' with

$$\hat{f}''(x) = \frac{1}{nk^3} \sum_{i=1}^n K'' \left(\frac{X_i - x}{k} \right) \quad (6)$$

Of course this merely replaces one problem with another, and estimates of f'' can vary greatly according to the choice of k . Sheather and Jones come up with a commonly use solution assuming that $k(h) = Ch^{5/7}$, where C depends on the data, but not on h .

3 Testing for the Number of Modes

Suppose we wish to test for number of modes, m , of a probability distribution. Specifically consider the hypotheses:

$$H_0 : n_{mode}(f) = m \quad vs \quad H_a : n_{mode}(f) > m \quad (7)$$

Where $n_{mode}(f)$ is the number of mixtures, or modes in the density. We compute a nonparametric kernel density estimate based on the observed data $X_1, X_2, \dots, X_n$

$$\hat{f}_{K,h}(x) = \frac{1}{hn} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right)$$

It may be shown that the number of modes of $\hat{f}_{K,h}(x)$ decreases monotonically as h increases. Let H_m be the minimal bandwidth for which $\hat{f}_{K,H_m}(x)$ has m modes. That is

$$n_{mode}(\hat{f}_{K,H_m}(x)) = m \quad and \quad n_{mode}(\hat{f}_{K,h}(x)) > m, h > H_m \quad (8)$$

Since a density with a high number of modes needs a higher degree of smoothing (i.e. using a larger bandwidth) to produce a density estimate with a smaller number of modes, we may use the minimal bandwidth as a test statistic. Thus if h_m were the observed minimal bandwidth for m modes, we could reject the null hypothesis at significance level α if the p-value is smaller than α , that is

$$P_0(H_m > h_m) \leq \alpha$$

A high minimal bandwidth h_m would indicate that the data has to undergo a large degree of smoothing to produce a kernel density estimate with m modes. The sampling distribution of H_m is unknown. However we may approximate the p-value by using the parametric bootstrap in which we repeatedly resample (with replacement) from the data, and approximate the p-value as follows by the proportion of times the resampled data produces a minimal bandwidth for m modes, which is larger than the one we observed. The steps of the algorithm may be stated as follows:

1. Draw B bootstrap samples of size n from $\hat{f}(\cdot; \hat{h}_m)$
2. For each sample compute $\hat{h}_m^*$, the smallest bandwidth that produces m modes, obtaining $\hat{h}_m^*(1), \hat{h}_m^*(2), \dots, \hat{h}_m^*(B)$
3. Approximate p-value with $\sum_{b=1}^n \mathbf{I}[\hat{h}_m^*(b) > \hat{h}_m] / B$

The first step may be done as follows: Let $y_1^*, y_2^*, \dots, y_n^*$ be a sample taken with replacement from the data $x_1, x_2, \dots, x_n$. Now set

$$x_i^* = \bar{y}^* + (1 + \hat{h}_m^2 / \hat{\sigma}^2)^{-1/2} (y_i^* - \bar{y}^* + \hat{h}_m \epsilon_i); i = 1, 2, \dots, n \quad (9)$$

where $\bar{y}^*$ is the mean of the y_i^* , $\hat{\sigma}^2$ is the sample estimate of the variance of the x_i , and the ϵ_i are standard normal random variables. It was hoped that this test might serve to estimate the number of targets, however it proved to be computationally incredibly slow on the computers available, and therefore of little practical use. Computer programs were written to compute the minimum bandwidth h_m giving a smoothed estimate of m modes at each time point (see tables 1 and 2).

4 Bayesian Estimation of Beam probabilities

At each time point, t , and at each beam, b , we wish to estimate a posterior signal probability $\theta_{t,b}$ based on the observed measurement $X_{t,b}$, and the previous estimate $\hat{\theta}_{t-1,b}$. We use the relationship between the Dirichlet and Multinomial distributions: The Dirichlet density is a k -dimensional version of the beta density. If $x = (x_1, x_2, \dots, x_k)$, $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$ and $A = \sum \alpha_i$, the Dirichlet density is

$$f(x|\alpha) = \frac{\Gamma(A)}{\prod_{i=1}^k \Gamma(\alpha_i)} x_1^{\alpha_1-1} x_2^{\alpha_2-1} \dots x_k^{\alpha_k-1}, x_i > 0, \alpha_i > 0, \sum x_i = 1$$

where $\Gamma(t)$ is Euler's gamma function

$$\Gamma(t) = \int_0^{\infty} x^{t-1} e^{-x} dx, t > 0$$

The means, variances, and covariances of the X_i are $E(X_i) = \alpha_i/A$, $Var(X_i) = \frac{\alpha_i(A - \alpha_i)}{A^2(A + 1)}$, and $Cov(X_i X_j) = -\frac{\alpha_i \alpha_j}{A^2(A + 1)}$. It is well known that the Dirichlet distribution is a conjugate prior for the multinomial distribution. Let $Y = (Y_1, Y_2, \dots, Y_k)$ have multinomial distribution with parameters $n, \theta_1, \theta_2, \dots, \theta_k, \sum_{i=1}^k \theta_i = n$ and $\sum_{i=1}^k \theta_i = 1$. If $\theta = (\theta_1, \theta_2, \dots, \theta_k)$ has prior Dirichlet distribution with parameters $\alpha_1, \alpha_2, \dots, \alpha_k$, then the posterior distribution of $\theta|Y = y, y = (y_1, y_2, \dots, y_k)$ is Dirichlet with parameters $y_1 + \alpha_1, y_2 + \alpha_2, \dots, y_k + \alpha_k$. Dirichlet random vector may be simulated in the following way:

To generate random vector $\theta = (\theta_1, \theta_2, \dots, \theta_k)$ having a Dirichlet distribution with parameter $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$:

- 1). Generate $X_1, X_2, \dots, X_k$ independent from gamma densities

$$f(x) = \frac{e^{-x} x^{\alpha_i - 1}}{\Gamma(\alpha_i)}, 1 \leq i \leq k$$

- 2). Set $\theta_i = \frac{X_i}{X_1 + X_2 + \dots + X_k}, 1 \leq i \leq k$

5 Estimation of Target Trajectories

At time $t, t = 1, 2, \dots, T$, we observe histogram count $X_{t,b}$ at beam $b, b = 1, 2, \dots, B$. The counts are generated from a mixture of k targets, according to Gaussian distributions.

$$f(x) = \sum_{i=1}^k p_i \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i}\right)^2} \quad (10)$$

Let n be the total number of observations also the sum of the beam counts at time $t, \sum_{b=1}^B X_{t,b} = n$.

- 1). Let $\hat{p}_{t-1,1}, \hat{p}_{t-1,2}, \dots, \hat{p}_{t-1,B}$ be estimates of the histogram probabilities at time $t-1, \sum_{b=1}^B \hat{p}_{t-1,b} = 1$

2). Using the relationship between the Dirichlet distribution and the multinomial distribution use the histogram data at time t to and the probabilities $\hat{p}_{t-1,b}$ obtain posterior probabilities

$$\tilde{p}_{t,b} = \frac{\hat{p}_{t-1,b} + X_{t,b}}{1 + \sum_{j=1}^B X_{t,b}} \quad (11)$$

- 3). Generate "posterior" histogram counts according to a Multinomial($n, \tilde{p}_{t,1}, \tilde{p}_{t,2}, \dots, \tilde{p}_{t,B}$) distribution
- 4). Smooth these counts using a Gaussian kernel density estimator, and use the smoothed data to:
 - a). Obtain updated beam probabilities $\hat{p}_{t,1}, \hat{p}_{t,2}, \dots, \hat{p}_{t,B}$
 - b). Obtain estimates of the number of targets and the position of the targets at time t from the number and position of the modes of the smoothed distribution
- 5). Use a spline to smooth the point estimates in cases where a track can be discriminated

6 Data Simulation

Two scenarios were simulated. In one in which three target start out in from separate positions, and then cross over. In the second there are initially three targets, which are joined by a fourth. There is no crossing of paths. The results of simulations are shown in figures 1 through 5.

References

- [Streit] Roy L. Streit. *Tracking on Intensity-Modulated Data Streams*. NUWC-NPT Technical Report 11221, May 1, 2000
- [Streit and Luginbuhl] Roy L. Streit and T.E. Luginbuhl. *Probabilistic Multi-Hypothesis Tracking*. NUWC-NPT Technical report 10428, February 15, 1995
- [Efron] Bradley Efron and Rob Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, 1993
- [Schaether and Jones] S.J. Scheather and M.C. Jones. A Reliable Data Base Bandwidth selection Method for Kernel Density Estimation *Journal of the Royal Statistical Society, Series B* 48, 683-690
- [Silverman] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986

TABLE 1

Tracking 3 targets over 50 time points (bwmin5 output) BW-SJ is adaptive bandwidth, modes is estimated number of targets at time t , $h(i)$ is minimum bandwidth producing i modes at time t

time	BW-SJ	modes	$h(1)$	$h(2)$	$h(3)$	$h(4)$	$h(5)$
[1,]	0.616	3	5.000	2.607	0.397	0.330	0.301
[2,]	0.637	3	5.000	2.406	0.445	0.358	0.320
[3,]	0.646	3	5.000	2.454	0.272	0.263	0.253
[4,]	0.626	3	5.000	2.138	0.406	0.320	0.282
[5,]	0.590	3	5.000	1.966	0.368	0.282	0.253
[6,]	0.574	3	5.000	1.832	0.339	0.282	0.272
[7,]	0.569	3	4.636	1.804	0.492	0.473	0.349
[8,]	0.599	3	5.000	1.507	0.291	0.282	0.263
[9,]	0.560	3	5.000	1.354	0.291	0.272	0.244
[10,]	0.547	3	4.713	1.067	0.291	0.253	0.244
[11,]	0.556	3	4.761	1.019	0.291	0.244	0.234
[12,]	0.525	3	4.282	0.579	0.416	0.416	0.378
[13,]	0.529	2	4.148	0.378	0.311	0.263	0.253
[14,]	0.517	2	4.081	0.330	0.330	0.263	0.263
[15,]	0.480	2	3.909	0.272	0.215	0.196	0.186
[16,]	0.464	2	3.497	0.282	0.263	0.244	0.244
[17,]	0.445	3	3.363	0.531	0.339	0.234	0.234
[18,]	0.452	2	3.124	0.435	0.425	0.311	0.253
[19,]	0.416	3	2.895	0.521	0.311	0.244	0.234
[20,]	0.418	2	2.751	0.263	0.244	0.205	0.196
[21,]	0.415	2	2.502	0.311	0.291	0.234	0.215
[22,]	0.402	2	2.196	0.215	0.196	0.186	0.157
[23,]	0.430	2	1.564	0.272	0.234	0.224	0.196
[24,]	0.465	2	1.449	0.387	0.349	0.301	0.263
[25,]	0.441	2	0.885	0.263	0.244	0.224	0.215
[26,]	0.415	2	0.569	0.311	0.253	0.234	0.234
[27,]	0.394	2	0.483	0.320	0.311	0.244	0.224
[28,]	0.448	1	0.311	0.301	0.282	0.263	0.215
[29,]	0.460	1	0.368	0.320	0.301	0.291	0.224
[30,]	0.354	2	0.378	0.320	0.253	0.224	0.186
[31,]	0.346	2	0.674	0.291	0.263	0.244	0.234
[32,]	0.367	3	0.875	0.397	0.205	0.196	0.186
[33,]	0.357	2	1.354	0.301	0.272	0.244	0.196
[34,]	0.348	3	1.708	0.397	0.301	0.224	0.215
[35,]	0.374	2	1.966	0.301	0.272	0.253	0.244
[36,]	0.451	2	2.119	0.358	0.320	0.311	0.311
[37,]	0.465	2	2.292	0.282	0.263	0.263	0.234
[38,]	0.490	3	2.359	0.789	0.492	0.330	0.291
[39,]	0.574	3	2.483	0.732	0.378	0.301	0.282
[40,]	0.463	3	2.962	1.306	0.425	0.339	0.301

[41,]	0.442	4	3.134	1.574	0.492	0.349	0.282
[42,]	0.468	3	3.182	1.804	0.311	0.301	0.282
[43,]	0.517	3	3.450	2.186	0.464	0.378	0.301
[44,]	0.527	3	3.593	2.359	0.311	0.301	0.263
[45,]	0.584	3	3.746	2.732	0.272	0.253	0.253
[46,]	0.578	3	3.813	2.981	0.311	0.282	0.282
[47,]	0.584	3	3.986	3.373	0.349	0.320	0.311
[48,]	0.643	3	4.215	3.679	0.291	0.263	0.244
[49,]	0.642	3	4.359	3.871	0.636	0.368	0.339
[50,]	0.664	3	4.483	4.139	0.272	0.244	0.196

TABLE 2

Tracking 3 targets over $t=1,\dots,25$, 4 targets over $t=26,\dots,50$ (bwmin6 output)
 BW-SJ is adaptive bandwidth, modes is estimated number of targets at time t ,
 $h(i)$ is minimum bandwidth producing i modes at time t

time	BW-SJ	modes	$h(1)$	$h(2)$	$h(3)$	$h(4)$	$h(5)$
[1,]	0.440	1	0.368	0.272	0.234	0.224	0.186
[2,]	0.487	1	0.378	0.349	0.272	0.234	0.224
[3,]	0.484	1	0.349	0.320	0.263	0.263	0.224
[4,]	0.496	1	0.387	0.311	0.291	0.263	0.234
[5,]	0.509	1	0.397	0.330	0.330	0.263	0.224
[6,]	0.495	2	0.502	0.425	0.272	0.244	0.234
[7,]	0.456	2	0.693	0.378	0.349	0.272	0.263
[8,]	0.490	2	0.569	0.272	0.272	0.244	0.244
[9,]	0.552	2	0.722	0.320	0.320	0.291	0.244
[10,]	0.536	1	0.397	0.397	0.320	0.282	0.234
[11,]	0.474	2	0.550	0.368	0.358	0.272	0.224
[12,]	0.478	2	0.531	0.416	0.301	0.263	0.253
[13,]	0.501	2	0.550	0.339	0.253	0.253	0.215
[14,]	0.474	2	0.732	0.445	0.406	0.311	0.282
[15,]	0.501	3	0.952	0.521	0.320	0.291	0.253
[16,]	0.454	2	1.028	0.339	0.301	0.224	0.224
[17,]	0.501	2	1.564	0.473	0.320	0.282	0.234
[18,]	0.453	3	1.181	0.540	0.358	0.282	0.282
[19,]	0.440	3	1.536	0.636	0.301	0.282	0.253
[20,]	0.481	3	1.593	0.732	0.330	0.311	0.301
[21,]	0.455	3	1.583	0.559	0.330	0.301	0.244
[22,]	0.442	3	1.737	0.875	0.349	0.301	0.224
[23,]	0.471	4	1.899	0.713	0.579	0.339	0.339
[24,]	0.475	3	1.985	0.913	0.425	0.358	0.282
[25,]	0.461	4	2.043	0.894	0.531	0.368	0.301
[26,]	0.619	4	3.201	2.359	1.028	0.358	0.244
[27,]	0.636	4	3.478	2.062	0.904	0.358	0.291
[28,]	0.617	4	3.201	1.985	0.636	0.425	0.406
[29,]	0.646	3	3.392	2.138	0.550	0.349	0.339
[30,]	0.639	3	3.852	2.024	0.607	0.311	0.311
[31,]	0.646	3	3.641	1.938	0.464	0.330	0.311
[32,]	0.642	3	3.899	1.909	0.454	0.406	0.282
[33,]	0.626	3	3.756	2.081	0.550	0.387	0.244
[34,]	0.627	3	4.215	2.167	0.358	0.330	0.272
[35,]	0.667	3	4.148	2.043	0.397	0.349	0.320
[36,]	0.638	3	4.493	2.043	0.349	0.282	0.272
[37,]	0.660	3	4.292	2.158	0.291	0.291	0.282
[38,]	0.652	3	4.531	2.043	0.378	0.311	0.282
[39,]	0.648	3	4.694	1.985	0.358	0.339	0.301
[40,]	0.649	3	4.789	2.158	0.263	0.263	0.244

[41,]	0.689	3	5.000	2.158	0.540	0.387	0.330
[42,]	0.693	4	5.000	2.234	0.894	0.368	0.301
[43,]	0.708	3	5.000	2.339	0.358	0.320	0.234
[44,]	0.735	3	5.000	2.196	0.311	0.282	0.215
[45,]	0.735	3	5.000	2.071	0.320	0.311	0.291
[46,]	0.769	3	5.000	2.387	0.425	0.358	0.301
[47,]	0.772	3	5.000	2.493	0.291	0.272	0.253
[48,]	0.768	3	5.000	2.512	0.301	0.263	0.263
[49,]	0.796	3	5.000	2.368	0.502	0.358	0.339
[50,]	0.833	3	5.000	2.435	0.339	0.291	0.291

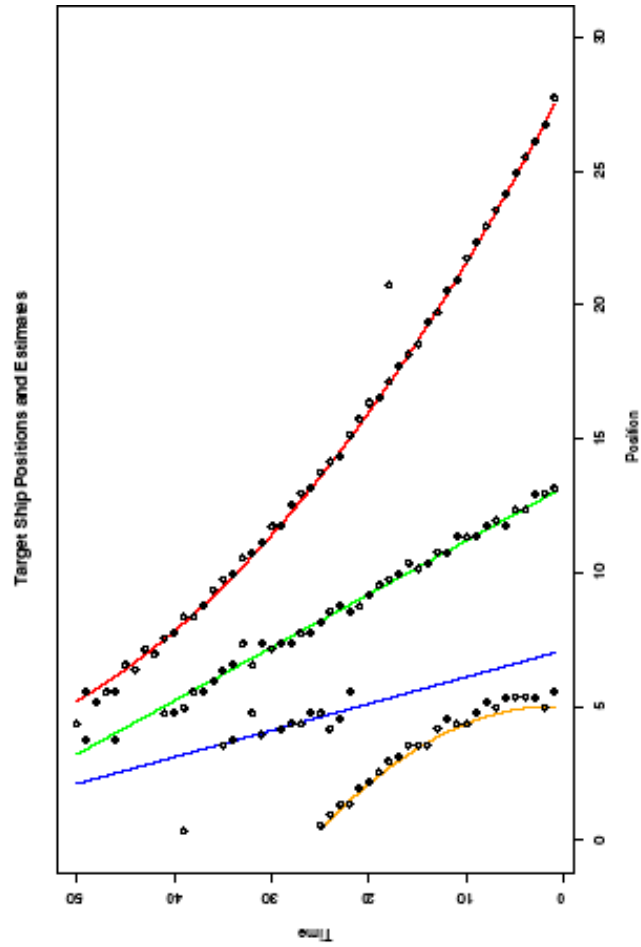


Figure 1. True target positions are red, blue, green, and orange lines. Dots are modal points of density estimators at each time period. Number of targets changes from 3 to 4.

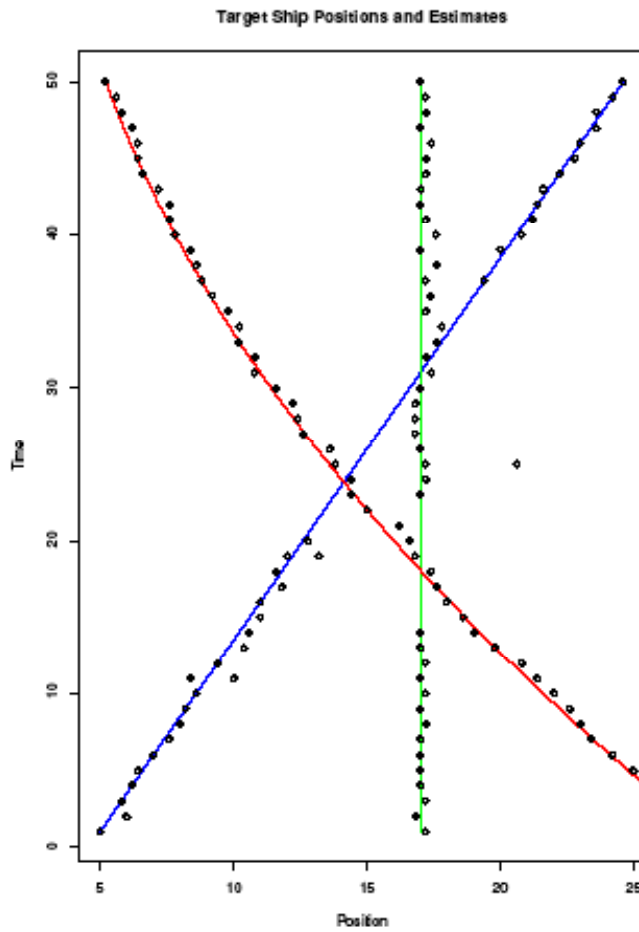


Figure 2. True target positions are red, blue and green lines. Dots are modal points of density estimators at each time period. Number of targets constant at 3.

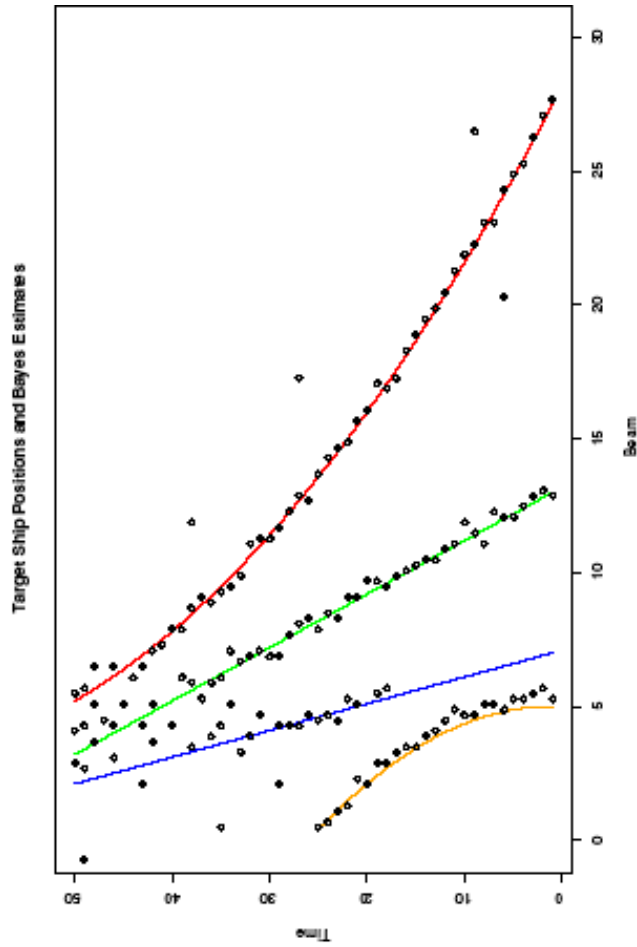


Figure 3. True target positions are red, blue, green, and orange lines. Dots are modal points of density estimators at each time period using Bayesian updates. Number of targets changes from 3 to 4.

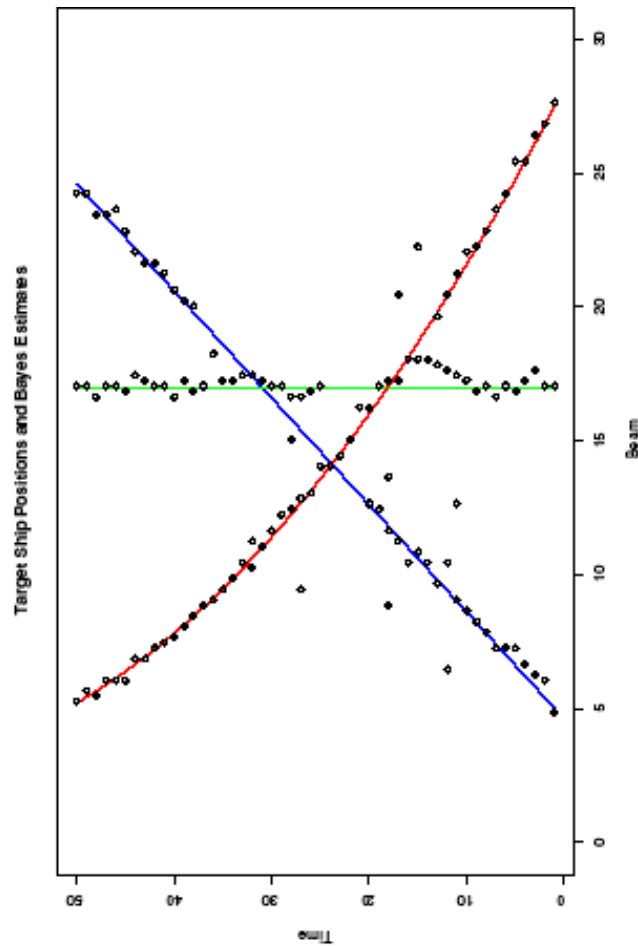


Figure 4. True target positions are red, blue and green lines. Dots are modal points of density estimators at each time period using Bayesian probability updates. Number of targets constant at 3

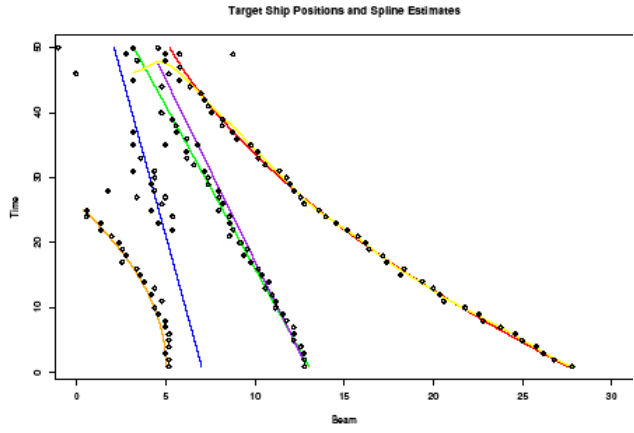


Figure 5. Smooth Spline estimates of targets are yellow and purple lines

APPENDIX C

Undergraduate Student Presentation NUWC, 2005

Using Parametric and Nonparametric Smoothing Techniques to Improve Estimation with the EM Algorithm

Angela EDWARDS

Bryahn IVERY

Dustin LUPTON

James PENDER

Department of Mathematics
North Carolina A&T State University

Mentored by

Dr. A.G. Warrack
Department of Mathematics
North Carolina A&T State University
warrack@ncat.edu

Dr. Marcus Graham
Naval Undersea Warfare Center,
Newport, RI

GrahamML@npt.nuwc.navy.mil

This work was supported by grant number N00014-03-1-0465 from the Office of Naval Research.

July 22, 2005

Introduction

In this paper we consider the problem of tracking two targets, and maintaining the separate track when either

- The two tracks cross,
- The two tracks approach each other, then after passing very close, diverge.

The data used was simulated using the R statistical programming language. R is the open source version of the S-Plus programming language developed at Bell Labs by John Chambers.

Type of Data

At each time point, $t, t=1,2,\dots,T$, we observe data in the form of a sample of n observations $x_1, x_2, \dots, x_n$. These data are generated according to a “mixture” of k normal distributions, each with mean and standard deviation (μ_j, σ_j) , $j=1,2,\dots,k$. Each observation, x_i , is sampled with probability p_j from distribution j , where

$p_1 + p_2 + \dots + p_k = 1$. It can be shown that the probability distribution for each x_i is

$$f(x) = \sum_{j=1}^k p_j \phi(x_i; \mu_j, \sigma_j)$$

Where $\phi(x_i; \mu_j, \sigma_j)$ is the normal density with mean and standard deviation (μ_j, σ_j) .

The means at time t represent the respective positions of k targets. In fact we should write $\mu_j(t)$, representing the true position of target number j at time t . In this presentation we will assume the number of targets, k , is known.

Estimation of Target Parameters

The parameters to be estimated at each time point t , are the means (positions), standard deviations and probabilities. Clearly the positions at time t should be incorporated into the estimation at time $t+1$, since we assume that all targets move in some reasonably smooth trajectory.

A standard statistical tool in estimating the parameters in a mixture distribution is the *Expectation-Maximization Algorithm* (generally known as the EM Algorithm), which seeks parameter estimates that maximize the likelihood function

$$L(\mu, \sigma, p; x) = \prod_{i=1}^n \sum_{j=1}^k p_j \phi(x_i; \mu_j, \sigma_j)$$

This is an iterative algorithm, which requires starting values, and then updates the parameter estimates at each iteration. Under fairly wide conditions this algorithm can be guaranteed to converge. It is notably useful in estimation for situations in which there is missing or censored data, as well as for mixture distributions.

In tracking situations it is sensible to incorporate the estimate at time t , or some function of them, to use as the initial estimates at time $t+1$.

In this study we consider various smoothing methods, both parametric and nonparametric, which we incorporate into the EM algorithm to improve target estimates, and to maintain target identity.

Smoothing

The concept of smoothing refers to the accurate fitting of a smooth curve to a set of “noisy” data (e.g. data full of error). A smooth estimate is an extremely accurate estimate of the original data point because smooth estimates greatly reduce noise, and help to prominently reveal the characteristics of the actual trajectory being tracked. In the research conducted this summer (May-July 2005), several smoothing techniques were used to smooth parametric as well as nonparametric regression estimates.

Parametric Regression

The parametric case is one in which the expected form of the function is known. In this instance, it is most accurate to perform a linear/multiple regression to estimate a specific, finite number of unknown parameters. The use of a weighted sum of the observations to retrieve our fitted values is pertinent to this case.

Simple Linear Regression Model

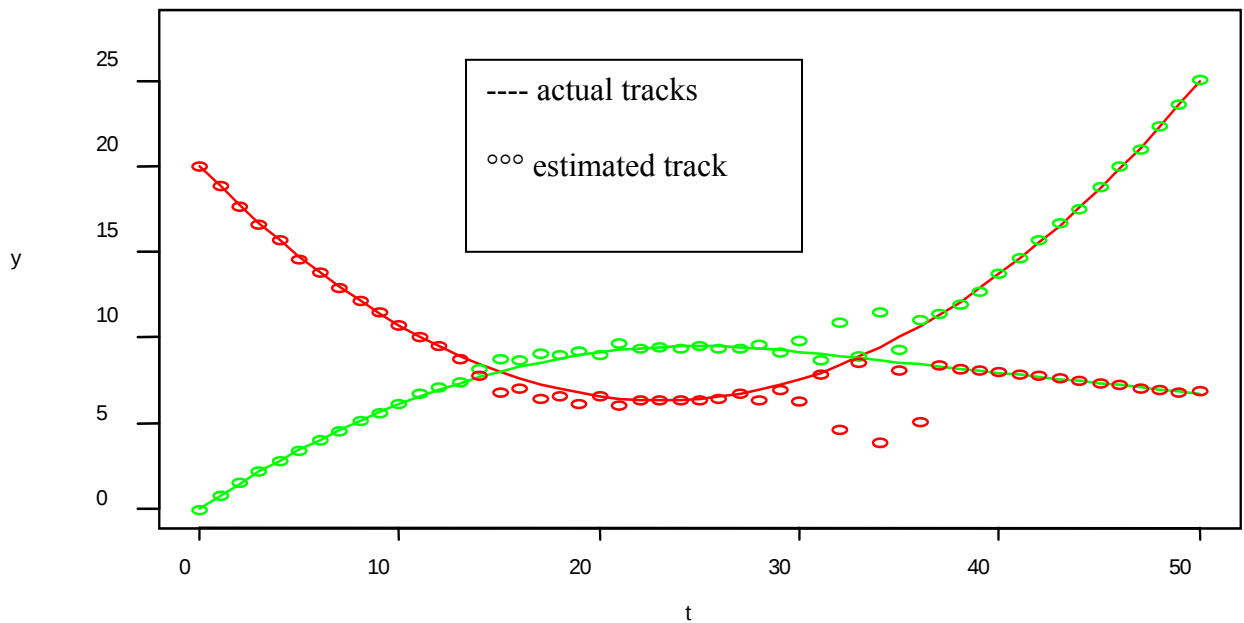
The first case explored was that of the linear regression model. The linear regression model is given by:

$$y_i = \alpha + \beta x_i + \varepsilon_i$$

Where α is the y-axis intercept, β is the slope (otherwise known as the regression coefficient), and the ε_i ’s are the corresponding error terms for each x_i . These error terms are considered to be independent and normally distributed with mean zero and standard deviation, σ^2 . The method of least squares is used to estimate the parameters α, β, σ^2 . For the trajectories that were studied, there was no need to use a linear regression fit. Though one of our trajectories has linear properties, it is more efficient and flexible to use a quadratic multiple regression analysis. This approach gives more flexibility because the quadratic regression model can track quadratic functions and linear functions, whereas the simple linear regression is used strictly for those functions that only display linear characteristics.

However, if we apply the linear regression model to the specific trajectories used, we can see specifically where this technique fails.

Fig. 1: Linear regression



In figure 1, the track is plotted as a function of time. The solid line denotes the user defined trajectory, and the hollow dots represent the estimated data points that were calculated and smoothed using simple linear regression. It is obvious that the program confuses the two tracks when they intersect the second time. Since the linear regression model is used for those trajectories that display only linear characteristics, the smoothing procedure will always estimate a data point that has a linear relationship with the previous estimate. For this reason, when the tracks cross the second time, the linear regression smoother wants to continue in a positive direction for the green track and a negative direction for the red track, thus causing the tracks to switch. This is not the case when a quadratic multiple regression smoother is used.

Nonparametric Regression

The nonparametric case is one in which the expected form of the function is unknown. In this instance, one must use an alternate way of determining the weights to be used in the regression. There are several different techniques that may be used. This summer three of these techniques were used (all of which have predetermined functions in R).

Kernel Regression

The Kernel regression smoother, most commonly known as the Nadaraya-Watson Kernel Regression Estimate, is used to determine the appropriate weights to use to yield

fitted values of a data set. The kernel, K , used this summer was that of the normal density:

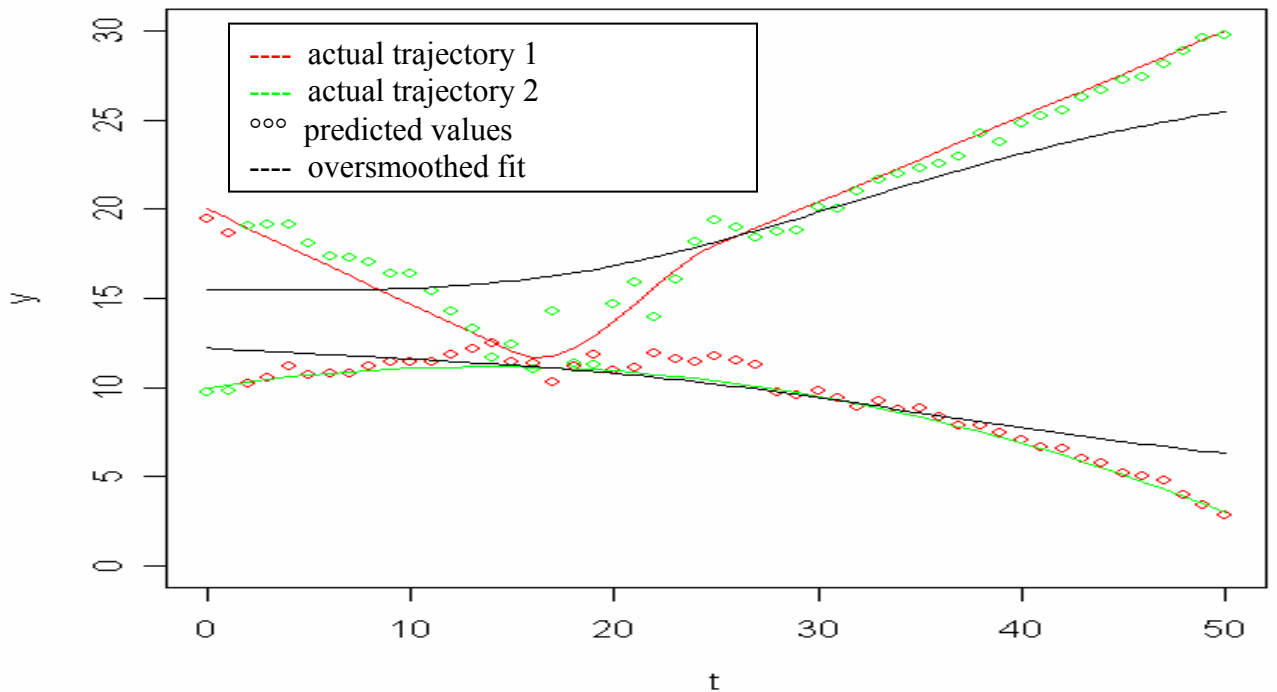
$$K = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

Using this kernel, R will evaluate the appropriate weights for each of n data points, assigning more weight to the estimates close to actual values and less weight to the estimates farther away from actual values. In R , the `ksmooth` function represents the kernel regression.

To accurately perform this regression one must also specify a bandwidth. The bandwidth is used to determine how fast the weights will decrease as the distance from the actual value increases. The choice of bandwidth is extremely important, as this value will determine how smooth the fitted values will be.

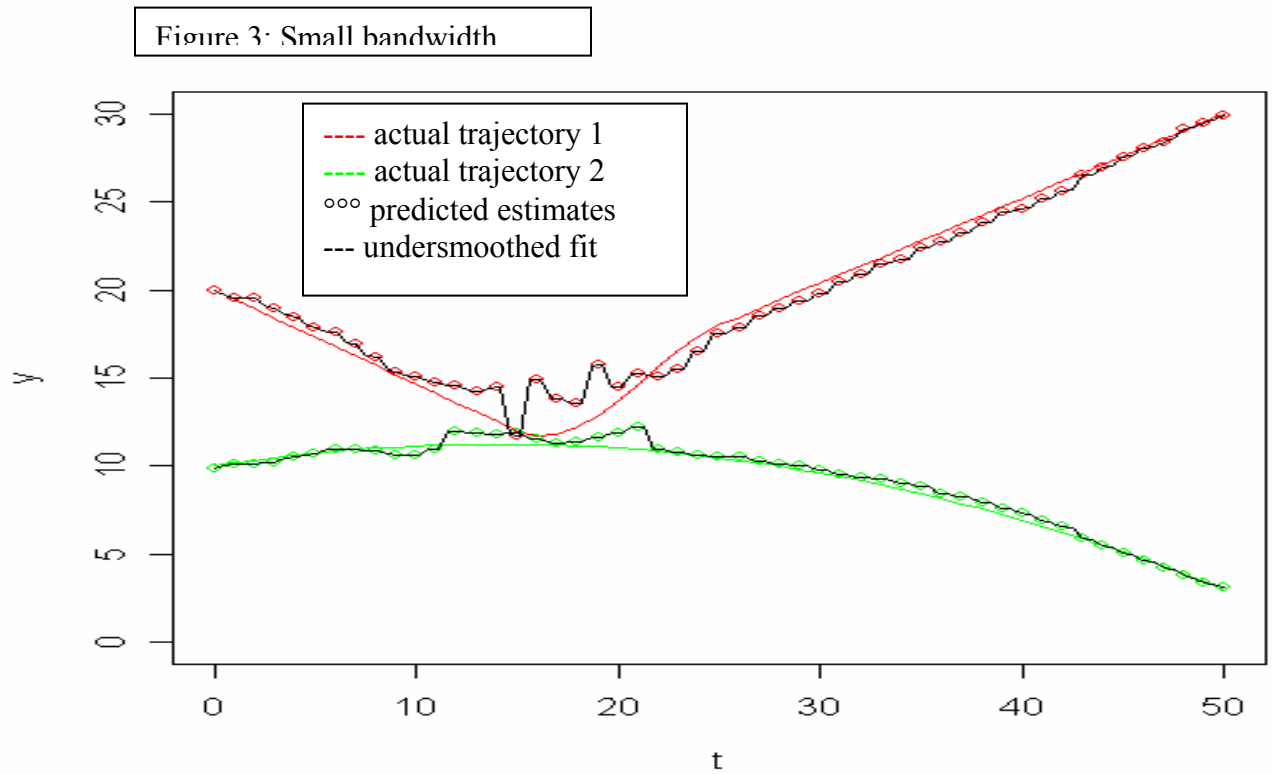
For example, choosing a bandwidth value that is too large (close to the actual sample size) will result in an over-smoothed fit. This is because when the bandwidth is large the weights are determined at a large number of points, thus they are virtually equal. The result is a set of smooth points that have a seemingly linear relationship as opposed to a relationship that closely resembles the actual trajectory. This is shown in Figure 2 where the bandwidth is set at 30.

Figure 2: Large Bandwidth



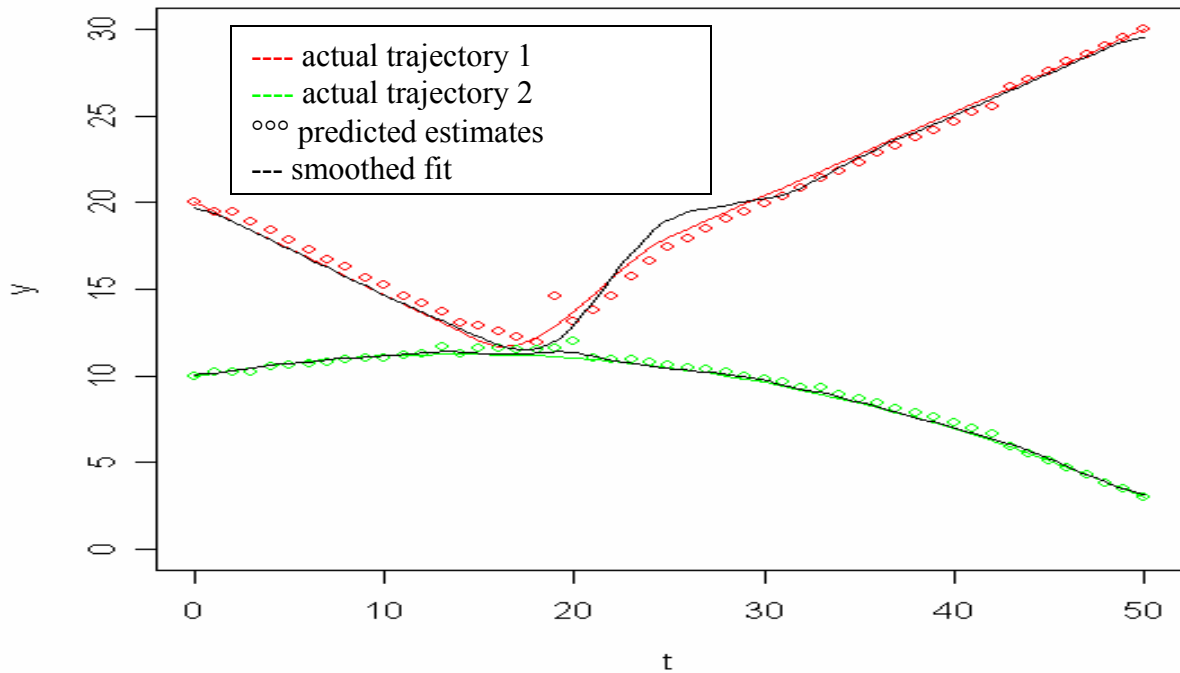
On the other hand, it is also possible to choose a bandwidth that is too small. When this occurs, the predicted point receives the most weight. In this case, each sample will yield a different fit because there is too much dependence on the individual data sets.

This results in unsmooth estimates with extremely high variances, as in Figure 3 where the bandwidth is set at 0.5.



However, using the “guess and check” process, one can eventually come up with an appropriate bandwidth value. For our purposes, it was most accurate to use a bandwidth of 3 for the first trajectory and a bandwidth of 2 for the second trajectory. Using these values produced the most accurate fit, as seen in figure 4.

Figure 4: Accurate Bandwidth



Though there is some error, it is obvious that with the appropriate bandwidth the kernel regression procedure can be an accurate smoothing algorithm.

II. Smoothing Splines

The smooth spline procedure is another function integrated by R to smooth data. This uses a combination of the ordinary least squares estimate and the loess smoothing procedure (loess procedure explained in more detail later). These smoothing splines adjust the level of smoothness by varying the curve from a least squares linear approximation to a cubic approximation, and using whichever approximation fits the original data set most appropriately.

A spline is a function that consists of several polynomial pieces joined together with certain smoothness conditions. A spline is calculated at several subintervals of an interval, I . The subintervals are determined by a certain number of knots, which we determine. The knots are the points of the original function at which the function changes its character (e.g. the function changes slope or changes direction).

For our purposes, we chose the number of knots for the first trajectory to be ten and we chose the number of knots for the second trajectory to be null. We chose ten for the first trajectory because there are several points at which the trajectory changes character. Specifying ten knots seemed to work the best, separating the trajectory into ten subintervals and calculating a spline at each interval. Because the second trajectory is simply a quadratic function, it worked best to specify the number of knots to be null. This is because the smooth spline function can make an accurate determination of how

many knots are needed if the function is “simple” (simple meaning strictly linear, strictly quadratic, strictly cubic etc.).

Figure 5: Smooth Spline

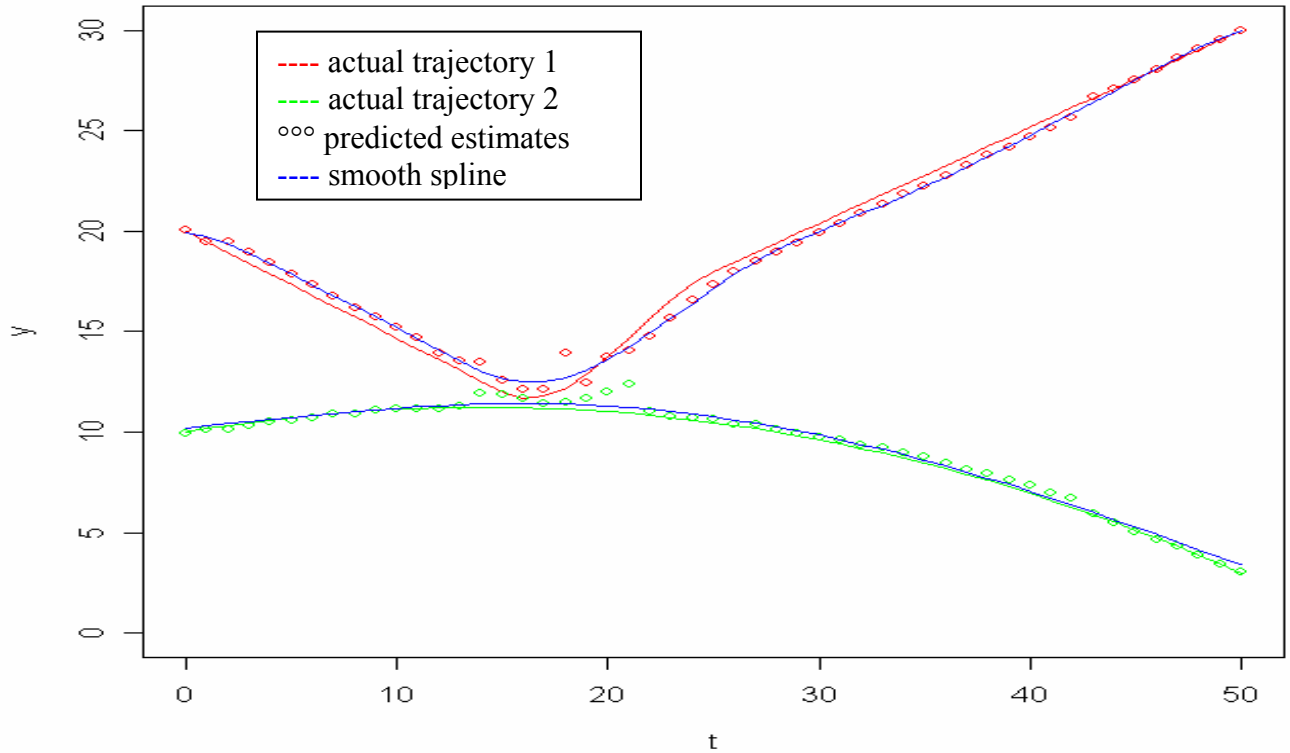


Figure 5 shows our interpolation of the smooth spline procedure. From figure 5 we see that the smooth spline procedure is extremely accurate, with the blue line representing our smooth spline estimate. Compared to the other procedures we used, the smooth spline estimate proved to be the most accurate procedure at tracking the original trajectory.

Loess regression

“Loess” stands for “locally weighted scatter plot smoother”. The procedure is complicated, but may be roughly outlined as follows: given data $(x_i, y_i), i = 1, 2, \dots, n$, we wish to estimate y for some given value of x . This estimate of y is obtained by fitting the weighted quadratic regression model:

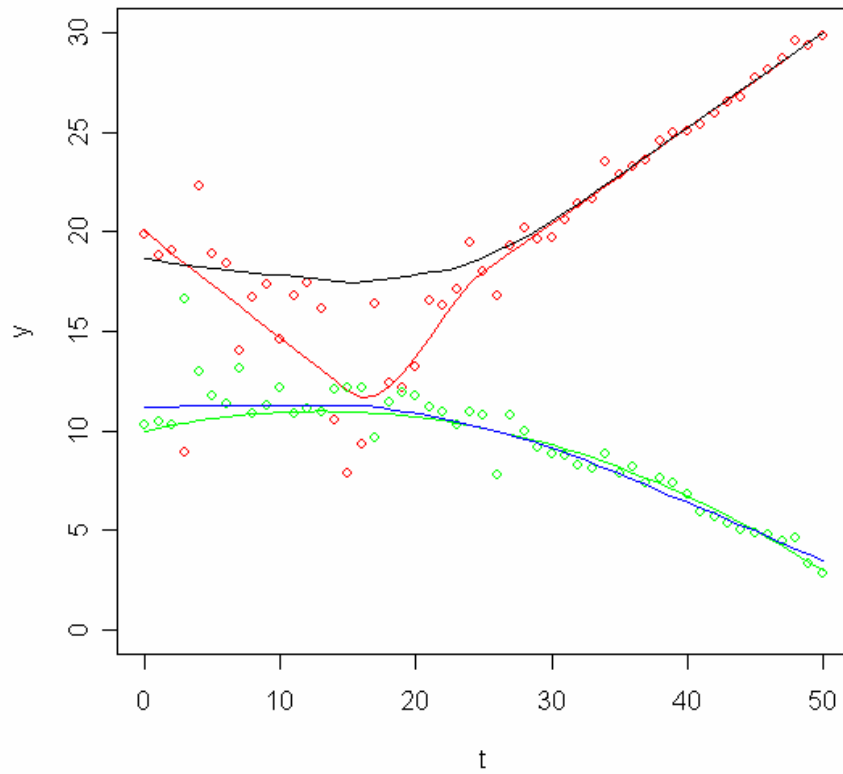
$$y_i = \beta_0 + \beta_1(x_i - x) + \beta_2(x_i - x)^2 + \varepsilon_i$$

Using the “tricube” function to determine weights

$$w_i = \left(1 - \left|\frac{x - x_i}{h}\right|^3\right)^3$$

where h is known as the “span”. The choice of a large value of h will general produce a very smooth curve, and that of a small value a more jagged one.

In this tracking program we used loess regression.



In the figure the loess regression doesn't read the estimates in the beginning as we would like, but follows well at the end. The red and green lines are the true trajectories that we are trying to find. The dots are the estimates from the EM algorithm, and the black and blue lines the smoothed estimate using the nonparametric loess regression.