AD-A068 629 NAVAL RESEARCH LAB WASHINGTON DC F/G 15/4

SOME REFINEMENTS FOR A SHIP TRACKING ALGORITHM.(U)

MAY 79 W W WILLMAN

UNCLASSIFIED NRL-MR-3991 NL

| OF |

AD
A068629

END
DATE
FILMED
7-79
DDC

LEVEL

42

NRL Memorandum Report 3991

Some Refinements for a Ship Tracking Algorithm

WARREN W. WILLMAN

Systems Research Branch
Space Systems Division

May 2, 1979

NAVAL RESEARCH LABORATORY
Washington, D.C.

(9) Memorandum rept.,

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER NRL Memorandum Report 3991	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) SOME REFINEMENTS FOR A SHIP TRACKING ALGORITHM		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL problem. 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Warren W. Willman		8. CONTRACT OR GRANT NUMBER(s) (16) RR00302
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NRL Problem B01-10 (17) Project RR003 02 41 6152 no PE
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, VA 22217		12. REPORT DATE (11) May [redacted]79 13. NUMBER OF PAGES 18
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 18p.		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

(14) NRL-MR-3991

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Adaptive driving noise
Command and control
Kalman filtering
Ocean surveillance
Recursive filtering
Ship tracking
Track smoothing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Some improvements are reported which have been developed for the ship tracking algorithm described in NRL Report 7969. Also, some errors which have been discovered in that report are listed.

251 950

CONTENTS

SOME REFINEMENTS FOR A SHIP TRACKING ALGORITHM

INTRODUCTION

Some refinements have been developed for the ship tracking algorithm in reference 1. Also, some errors in this reference have been discovered, and are listed in Appendix A here. The availability of reference 1, as so amended, is assumed in this report.

These refinements often make little difference when the input position reports of the ship being tracked have a large margin of error. In other cases, however, the difference can be dramatic, especially with accurate reports followed by inaccurate ones. The original algorithm does badly in such cases because certain pecularities of the underlying ship motion model cause it to adhere too rigidly to the "average velocity" estimated from early data. The refinements correct this tendency by adjusting the "state covariance matrix" whenever the estimated "driving noise" increases. This adjustment is theoretically exact for the case of only two position reports with no prior velocity information.

Upon the receipt of each new position report, the original algorithm operates in planar coordinates by first updating an estimate of the ship's state vector $(x, y, \dot{x}, \dot{y})$ with a Kalman filter, then updating an estimate of a 2 x 2 driving noise covariance matrix Q with the "residuals" from this filter. This Q matrix estimate is then modified to make it diagonal with respect to the current estimate of the velocity vector.

One refinement consists of subsequently updating the velocity components of the state covariance matrix. In the currently estimated "in-track" and "cross track" coordinates, this (2 x 2) submatrix is increased by $(t-t_o)^{-1}\tilde{Q}$, where each term in $\tilde{Q}$ is the greater of zero and the <u>change</u> in the corresponding term of Q from its preceding value, and where t is the current time and t_o is the time that tracking started.

The rationale for this procedure is that this is exactly the correction that would be needed in the state covariance matrix after two position reports if this matrix were generated by the Kalman filter using too low a value of the driving noise. In a sense, then, this refinement introduces a degree of coordination that was hitherto lacking between the updating of the Kalman filter and of the Q matrix estimate.

Note: Manuscript submitted February 26, 1979.

For good results, it was also found necessary to adopt a more exact procedure for diagonalizing Q (in the currently estimated in-track and cross-track coordinates) than was used in the original algorithm. As a practical safeguard against extremely bad data, it was found wise to limit the size of the increment in the velocity covariance submatrix to that submatrix's initially specified value.

The details of these refinements are specified for the case of planar tracking in the experimental FORTRAN implementation of Appendix B, which corresponds to the program listed in reference 1 (Figure 3, pg. 20-23) for the original algorithm. The section "UPDATE DRIVING NOISE ESTIMATE" is replaced entirely; also a few bookkeeping details are added in the "INITIALIZATION" section. These refinements extend in a straightforward way to the algorithms described in reference 1 for tracking on a sphere, because these are both derived from the planar tracking procedure. In this case, however, an even better procedure, using rhumb-line track projections, is described in reference 2.

REFINEMENTS FOR PLANAR TRACKING

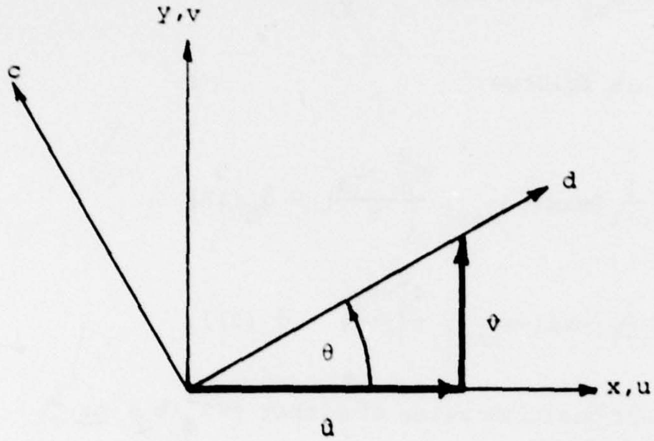
A. <u>New Method of Recursively Estimating the "Maneuvering Intensity" Matrix Q</u> (a 2 x 2 matrix)

Instead of computing the statistics $\hat{q}_{xx}$, $\hat{q}_{xy}$ and $\hat{q}_{yy}$ (p. 11 of reference 1), we compute two statistics, $\hat{q}_d$ and $\hat{q}_c$, which approximate the down-track and cross-track components of the "maneuvering noise" intensity. Assuming that these two components are uncorrelated, the sort of reasoning used in reference 1 to derive eqs. (27)-(29) leads to a similar recursive scheme for generating $\hat{q}_d$ and $\hat{q}_c$. In the notation of reference 1, this scheme consists of doing the following at time t_{i+1}, <u>after</u> updating the conditional mean and covariance matrix of the state vector according to eqs. (8)-(24):

(i) Compute the down-track and cross-track components of the "innovation" vector

$$\begin{bmatrix} \varepsilon_x \ (i+1) \\ \hline \varepsilon_x \ (i+1) \end{bmatrix} .$$

To do this, we use the following coordinate transformation:



$$\sin\theta = \frac{\hat{v}}{\sqrt{\hat{u}^2 + \hat{v}^2}} ,$$

$$\cos\theta = \frac{\hat{u}}{\sqrt{\hat{u}^2 + \hat{v}^2}} ,$$

$\hat{u}$, $\hat{v}$ are estimated velocity components at t^+_{i+1}

Hence, we have

$$\varepsilon_d = \varepsilon_x(i+1)\cos\theta + \varepsilon_y(i+1)\sin\theta$$

$$\varepsilon_c = -\varepsilon_x(i+1)\sin\theta + \varepsilon_y(i+1)\cos\theta$$

(ii) Compute the variance of ε_d and ε_c, which we denote by b_d and b_c.

The covariance matrix of $\begin{bmatrix} \varepsilon_x(i+1) \\ \hline \varepsilon_y(i+1) \end{bmatrix}$ is $\left[\begin{array}{c|c} b_{xx} & b_{xy} \\ \hline b_{xy} & b_{yy} \end{array}\right]$,

where, in the notation of reference 1

$$b_{xx} = p_{xx}(t_i^+) + 2\tau\, p_{xu}(t_i^+) + \tau^2 p_{uu}(t_i^+) + r_{xx}(i+1),$$

$$b_{xy} = p_{xy}(t_i^+) + \tau p_{xv}(t_i^+) + \tau p_{yu}(t_i^+) + \tau^2 p_{uv}(t_i^+) + r_{xy}(i+1) \text{ and}$$

$$b_{yy} = p_{yy}(t_i^+) + 2\tau\, p_{yv}(t_i^+) = \tau^2 p_{vv}(t_i^+) = r_{yy}(i+1).$$

Hence, from the coordinate transformation,

$$b_d = b_{xx} \cos^2\theta + 2\, b_{xy} \sin\theta \cos\theta + b_{yy}\sin^2\theta$$

$$b_c = b_{xx} \sin^2\theta - 2\, b_{xy} \sin\theta \cos\theta + b_{yy}\cos^2\theta .$$

(iii) Update $\hat{q}_d$ and $\hat{q}_c$ as follows:

$$\hat{q}_d(i+1) = \hat{q}_d(i) + \frac{1}{i+1}\left[\max\left(-e_{max}, \frac{\varepsilon_d^2 - b_d}{\tau}\right) - \hat{q}_d(i)\right]$$

$$\hat{q}_c(i+1) = \hat{q}_c(i) + \frac{1}{i+1}\left[\max\left(-e_{max}, \frac{\varepsilon_c^2 - b_c}{\tau}\right) - \hat{q}_c(i)\right],$$

where e_{max} is 0 or the greatest positive value of either $\frac{1}{\tau}(\varepsilon_d^2 - b_d)$ <u>or</u> $\frac{1}{\tau}(\varepsilon_c^2 - b_c)$ generated in any iteration so far (including this one). The use of e_{max} keeps $\hat{q}_d$ and $\hat{q}_c$ from occasionaly achieving unrealistically large negative values.

This recursion scheme is carried out for $i \geq 0$ (i.e., after the <u>second</u> observation, because the first is z_0 in the notation of reference 1). It is initiated by setting $\hat{q}_d(0) = \hat{q}_c(0) = e_{max} = 0$.

Until the second observation time t_1, we use $Q = 0$ (or some other selected initial value) for the "maneuvering matrix." For t in the interval (t_i, t_{i+1}), we use an approximation for Q constructed from $\hat{q}_d(i)$ and $\hat{q}_c(i)$ as follows:

$$c_d = \max[\hat{q}_d(i), 0] \qquad \text{so } C = \begin{bmatrix} c_d & 0 \\ 0 & c_c \end{bmatrix} \text{ is}$$

$$c_c = \max[\hat{q}_c(i), 0]$$

positive-semidefinite.

$Q = \begin{bmatrix} q_{xx} & q_{xy} \\ q_{yy} & q_{yy} \end{bmatrix}$ is the result of transforming C from down-track, cross-track coordinates (as currently estimated) to x,y coordinates, so

$$q_{xx} = c_d \cos^2\theta + c_c \sin^2\theta$$

$$q_{xy} = (c_d - c_c) \sin\theta \cos\theta$$

$$q_{yy} = c_d \sin^2\theta + c_c \cos^2\theta .$$

B. Updating Velocity Covariance Submatrix (2 x 2)

Immediately after updating Q as above (say at t_i^+, $i \geq 1$), we also compensate by adjusting the submatrix

$$\left[\begin{array}{c|c} p_{uu}(t_i^+) & p_{uv}(t_i^+) \\ \hline p_{uv}(t_i^+) & p_{vv}(t_i^+) \end{array}\right]$$

as follows:

(i) Let a_d and a_c denote the values of c_d and c_c obtained on the previous iteration (initially, $a_d = a_c = 0$). Then compute

$$d_d = \min \left[p_{uu}(t_o^+), \quad \frac{\max(0,\ c_d-a_d)}{t_i-t_o} \right]$$

and

$$d_c = \min \left[p_{uu}(t_o^+), \quad \frac{\max(0,\ c_c-a_c)}{t_i-t_o} \right] .$$

$\left[\begin{array}{c|c} d_d & 0 \\ \hline 0 & d_c \end{array}\right]$ is the covariance submatrix increment in down-track, cross-track coordinates. Also note: $p_{uu}(t_o^+) = \frac{1}{2}V^2 = p_{vv}(t_o^+)$.

(ii) Transforming the increment to x,y coordinates gives the updated covariance submatrix components as

$$p_{uu}(t_i^+) = p_{uu}(t_i^+) + d_d\cos^2\theta + d_c\sin^2\theta,$$

$$p_{uv}(t_i^+) = p_{uv}(t_i^+) + (d_d-d_c)\ \sin\theta\ \cos\theta, \text{ and}$$

$$p_{vv}(t_i^+) = p_{vv}(t_i^+) + d_d\sin^2\theta + d_c\cos^2\theta,$$

where "=" is used as in FORTRAN.

C. Summary

Thus the "Final Algorithm" on pp. 12-13 of reference 1 is changed as follows:

1. q_{xx}, $\dot{q}_{xy}$ and $\dot{q}_{yy}$ are not used; replaced by $\dot{q}_d$ and $\dot{q}_c$.

   Initial values are $\dot{q}_d(0) = \dot{q}_c(0) = 0$. Another initial value is $e_{max} = 0$.

2. The first step is replaced by two steps:

   - Generate q_{xx}, q_{xy}, q_{yy} (for $i \geq 1$, these are trivially 0 for $i = 0$) from $\dot{q}_d(i)$ and $\dot{q}_c(i)$ as described above in (A). Also update e_{max}.

   - Adjust $p_{uu}(t_i^+)$, $p_{uv}(t_i^+)$ and $p_{vv}(t_i^+)$ as described in above in (B).

3. The next two steps stay the same. The last step changes to:

   - Compute $\dot{q}_d(i+1)$ and $\dot{q}_c(i+1)$ as described above in (A).

RHUMB-LINE TRACK PROJECTION

The preceding modifications for planar tracking can be incorporated into algorithms for tracking on a sphere in the same way that is described in reference 1. For tracking on a sphere with geographical coordinates (i.e., latitude and longitude), however, an even better procedure, due to T. G. Bugenhagen of the Johns Hopkins University Applied Physics Laboratory [2], is to propagate the estimated target position between observations along a rhumb-line instead of a great-circle path, again by dead reckoning with the estimated average velocity. This modification eliminates the need for coordinate rotations, and is probably a better approximation to actual ship motion anyway. The "track propagation" procedure specified on pp. 25 and 26 of reference 1 then simplifies to the following:

- Beginning at time t_i^+ with the y axis oriented toward local north and the x axis toward local east, let ϕ_i and ψ_i be the latitude and longitude of the estimated position at that time. In these local rectangular coordinates, the velocity estimates $\hat{u}_i$, $\hat{v}_i$, the (4 x 4) state covariance matrix P_i, and the maneuvering parameter estimates $\dot{q}_d(i)$ and $\dot{q}_c(i)$ (see part A of preceding section) are also available. For convenience here, denote south latitudes and west longitudes as negative.

- Propagate the ship's position by dead reckoning to the time t_{i+1} of the next observation. The latitude $\hat{\phi}_{i+1}$ and longitude $\hat{\psi}_{i+1}$

of this dead-reckoned position can be computed analytically as follows:

$$\tilde{\phi}_{i+1} = \phi_i + \frac{\hat{v}_i}{R_e}(t_{i+1} - t_i);\ R_e = \text{earth radius}$$

(unless $|\tilde{\phi}_{i+1}| \geq \frac{1}{2}\pi$, in which case this procedure is no longer valid.)

$$\tilde{\psi}_{i+1} = \begin{cases} \psi_i + \dfrac{\hat{u}_i(t_{i+1} - t_i)}{R_e \cos\phi_i} & \text{if } \hat{v}_i(t_{i+1} - t_i) = 0 \\[2ex] \psi_i + \left(\dfrac{\hat{u}_i}{\hat{v}_i}\right) \ln\left[\left(\dfrac{\cos\phi_i}{\cos\tilde{\phi}_{i+1}}\right)\cdot\left(\dfrac{1+\sin\tilde{\phi}_{i+1}}{1+\sin\phi_i}\right)\right] & \text{otherwise} \end{cases}$$

- Compute q_{xx}, q_{xy} and q_{yy} from $\hat{q}_d(i)$ and $\hat{q}_c(i)$ as in part A of the preceding section.

- Compute the matrix M_{i+1} from eqs. 12 through 21 of reference 1.

- Set

$$\left.\begin{aligned} \hat{x}(t_{i+1}^-) &= 0 \\ \hat{y}(t_{i+1}^-) &= 0 \\ \hat{u}(t_{i+1}^-) &= \hat{u}_i \\ \hat{v}(t_{i+1}^-) &= \hat{v}_i \end{aligned}\right\} \text{values at time } t_{i+1}^- \text{ in local coordinates}$$

- Perform the rest of the procedure as in reference 1, except that P_{i+1} and $\hat{q}_d(i+1)$ and $\hat{q}_c(i+1)$ (instead of Q_{i+1}) are computed as described in the preceding section of this report.

REFERENCES

[1] W. W. Willman, "Recursive Filtering Algorithms for Ship Tracking," NRL Report 7969, April 6, 1976.

[2] T. G. Bugenhagen and L. B. Carpenter, "OTH/DC&T Engineering Analysis, Vol. 5 - A Kalman Filter Rhumb-Line Ship Tracking Algorithm," APL/JHU Report FS-79-032, February 1979.

APPENDIX A

ERRATA

NRL Report 7969
Recursive Filtering Algorithms for Ship Tracking
Warren W. Willman
April 6, 1976

p. 10: v_i in second expression from bottom should be $\hat{v}_i$.

pp. 10 & 11: m_{xx}, m_{xy} and m_{yy} should be replaced by the respective expressions

$$p_{xx}(t_i^+) + 2p_{xu}(t_i^+)\,\tau + p_{uu}(t_i^+)\,\tau^2,$$

$$p_{xy}(t_i^+) + \{p_{xv}(t_i^+) + p_{yu}(t_i^+)\}\tau + p_{uv}(t_i^+)\,\tau^2 \text{ and}$$

$$p_{yy}(t_i^+) + 2p_{yv}(t_i^+)\,\tau + p_{vv}(t_i^+)\,\tau^2.$$

p. 11: These same expressions, with the index i changed to j-1, should replace $p_{xx}(t_j^-)$, $p_{xy}(t_j^-)$ and $p_{yy}(t_j^-)$, respectively.

$\hat{q}_{xx}(i-1)$ and $\hat{q}_{xy}(i-1)$ in the second two equations should be $\hat{q}_{xx}(i+1)$ and $\hat{q}_{xy}(i+1)$.

p. 21: Second FORTRAN statement from bottom should be replaced by the four statements

```
RI  = FLOAT(I-1)
GXX = GXX-QXX*TAU        (because index I starts
GXY = GXY-QXY*TAU         at 1 instead of 0)
GYY = GYY-QYY*TAU
```

p. 25: The coordinates (ϕ_{i+1}, ψ_{i+1}) at the end of line 7 should be $(\tilde{\phi}_{i+1}, \tilde{\psi}_{i+1})$. Eq. (51) should read $\tilde{\phi}_{i+1} = \sin^{-1}(\sin\phi_i \cos\gamma + \frac{\hat{v}_i}{f}\cos\phi_i \sin\gamma)$.

p.38: Right hand side of equation (A3b) should read

$$P(t_i^-) - P(t_i^-)H_i^T(H_iP(t_i^-)H_i^T + R_i)^{-1}H_iP(t_i^-).$$

p. 45: v_i in middle alternative of (C9) should be $\hat{v}_i$; the same for u_i and $\hat{u}_i$.

p. 46: $\hat{q}_{xy}$ in last equation should be $\hat{q}_{yy}$.

APPENDIX B

MODIFICATIONS IN PROGRAM LISTING OF FIGURE 3 IN NRL REPORT 7969

```
      PROGRAM TEST
C
C                    KALMAN FILTER WITH ADAPTIVE DRIVING NOISE
C
      DIMENSION T(999),XX(999),YY(999),XS(999),YS(999),PXX(999),AX(999)
      DIMENSION AY(999),PXY(999),PYY(999),ARXX(999),ARXY(999),ARYY(999)
      DIMENSION SMAJ(999),SMIN(999),TH(999)
C
C                    READ PARAMETER VALUES
C
C                    N = NUMBER OF DETECTIONS (= NO. OF DATA CARDS)
C                    VELVAR = PRIOR SPEED VARIANCE
C
      READ 5071,N,VELVAR
 5071 FORMAT(I3,F10.5)
C
C                    READ (AND STORE) DATA FOR EACH DETECTION
C
C        T = TIME
C        AX,AY = OBSERVED LOCATION COORDINATES
C        SMA = SEMIMAJOR AXIS OF 86 PERCENT CONTAINMENT ELLIPSE
C              FOR OBSERVATION
C        SMI = SEMIMINOR AXIS OF CONTAINMENT ELLIPSE
C        THT = ORIENTATION OF SEMIMAJOR AXIS (DEGREES CLOCKWISE
C              FROM Y-AXIS)
C
 5070 FORMAT(6F10.4)
      DO 9 I=1,N
      READ 5070,T(I),AX(I),AY(I),SMA,SMI,THT
      THT=THT/57.3
      ARXX(I)=((SMA*SIN(THT))**2+(SMI*COS(THT))**2)/4.
      ARXY(I)=SIN(THT)*COS(THT)*(SMA*SMA-SMI*SMI)/4.
    9 ARYY(I)=((SMI*SIN(THT))**2+(SMA*COS(THT))**2)/4.
C
C                    INITIALIZATION
C
      XX(1)=AX(1)
      YY(1)=AY(1)
      PXX(1)=ARXX(1)
      PXY(1)=ARXY(1)
      PYY(1)=ARYY(1)
      C1=PXX(1)+PYY(1)
      C2=SQRT((PXX(1)-PYY(1))**2+4.*PXY(1)**2)
      C1=.5*(C1+C2)
      C2=C1-C2
      SMAJ(1)=2.*SQRT(C1)
      SMIN(1)=2.*SQRT(C2)
```

```
      IF(PXY(1).NE.0.) GO TO 71
      TH(1)=0.
      GO TO 51
   71 TH(1)=57.3*ATAN((PXX(1)-C1)/PXY(1))+90.
   51 CONTINUE
      HI=0.
      HC=0.
      U=0.
      V=0.
      PXU=0.
      PXV=0.
      PYU=0.
      PYV=0.
      PUU=.5*VELVAR
      PUV=0.
      PVV=PUU
      QXX=0.
      QXY=0.
      QYY=0.
      CI=0.
      CC=0.
      EMAX=0.
      RI=0.
      PRINT 10
      PRINT 11
   10 FORMAT(50X,19HPREDICTED POSITIONS,//)
   11 FORMAT(30X,66HTIME              X              Y              SMAJ AX    SM
     1IN AX    ORIENT,//)
C
C                      RECURSIVE STATE VECTOR ESTIMATION
C
      DO 1 I=2,N
      ZX=AX(I)
      ZY=AY(I)
      RXX=ARXX(I)
      RXY=ARXY(I)
      RYY=ARYY(I)
      TAU=T(I)-T(I-1)
      XBAR=XX(I-1)+U*TAU
      YBAR=YY(I-1)+V*TAU
      GXX=PXX(I-1)+2.*PXU*TAU+PUU*TAU*TAU+QXX*TAU
      GXY=PXY(I-1)+(PXV+PYU)*TAU+PUV*TAU*TAU+QXY*TAU
      GYY=PYY(I-1)+2.*PYV*TAU+PVV*TAU*TAU+QYY*TAU
      GXU=PXU+PUU*TAU
      GXV=PXV+PUV*TAU
      GYU=PYU+PUV*TAU
      GYV=PYV+PVV*TAU
C
C     CURRENT PREDICTED POSITION OUTPUT
C
C        I = OBSERVATION INDEX
C        T = TIME
C        (XX,YY) = CURRENT PREDICTED POSITION IN X-Y COORDINATES
C        SMAJ = SEMIMAJOR AXIS OF 86 PERCENT CONTAINMENT ELLIPSE FOR
C               CURRENT POSITION
C        SMIN = SEMIMINOR AXIS OF CONTAINMENT ELLIPSE
C        TH = ORIENTATION OF SEMIMAJOR AXIS (DEG. CLOCKWISE FROM Y-AXIS)
C
```

```
      C1=GXX+GYY
      C2=SQRT((GXX-GYY)**2+4.*GXY*GXY)
      C1=.5*(C1+C2)
      C2=C1-C2
      SMAJ(I)=2.*SQRT(C1)
      SMIN(I)=2.*SQRT(C2)
      IF(GXY.NE.0.) GO TO 70
      TH(I)=0.
      GO TO 50
   70 TH(I)=57.3*ATAN((GXX-C1)/GXY)+90.
   50 PRINT 7,T(I),XBAR,YBAR,SMAJ(I),SMIN(I),TH(I)
C
      DET=(GXX+RXX)*(GYY+RYY)-(GXY+RXY)**2
      DETI=0.
      IF(DET.GT.0.) DETI=1./DET
      HXX=(GYY+RYY)*DETI
      HXY=-(GXY+RXY)*DETI
      HYY=(GXX+RXX)*DETI
      PXX(I)=GXX-GXX*GXX*HXX-2.*GXX*GXY*HXY-GXY*GXY*HYY
      PXY(I)=GXY-GXX*GXY*HXX-(GXX*GYY+GXY*GXY)*HXY-GXY*GYY*HYY
      PYY(I)=GYY-GYY*GYY*HYY-2.*GYY*GXY*HXY-GXY*GXY*HXX
      C1=PXX(I)+PYY(I)
      C2=SQRT((PXX(I)-PYY(I))**2+4.*PXY(I)**2)
      C1=.5*(C1+C2)
      C2=C1-C2
      SMAJ(I)=2.*SQRT(C1)
      SMIN(I)=2.*SQRT(C2)
      IF(PXY(I).NE.0.) GO TO 72
      TH(I)=0.
      GO TO 52
   72 TH(I)=57.3*ATAN((PXX(I)-C1)/PXY(I))+90.
   52 CONTINUE
      PXU=GXU-GXX*GXU*HXX-(GXX*GYU+GXY*GXU)*HXY-GXY*GYU*HYY
      PXV=GXV-GXX*GXV*HXX-(GXX*GYV+GXY*GXV)*HXY-GXY*GYV*HYY
      PYU=GYU-GYY*GYU*HYY-(GYY*GXU+GXY*GYU)*HXY-GXY*GXU*HXX
      PYV=GYV-GYY*GYV*HYY-(GYY*GXV+GXY*GYV)*HXY-GXY*GXV*HXX
      PUU=PUU-GXU*GXU*HXX-2.*GXU*GYU*HXY-GYU*GYU*HYY
      PUV=PUV-GXU*GXV*HXX-(GXU*GYV+GYU*GXV)*HXY-GYU*GYV*HYY
      PVV=PVV-GXV*GXV*HXX-2.*GXV*GYV*HXY-GYV*GYV*HYY
      DET=RXX*RYY-RXY*RXY
      DETI=0.
      IF(DET.GT.0.) DETI=1./DET
      HXX=RYY*DETI
      HXY=-RXY*DETI
      HYY=RXX*DETI
      XX(I)=XBAR+(PXX(I)*HXX+PXY(I)*HXY)*(ZX-XBAR)
      XX(I)=XX(I)+(PXX(I)*HXY+PXY(I)*HYY)*(ZY-YBAR)
      YY(I)=YBAR+(PXY(I)*HXX+PYY(I)*HXY)*(ZX-XBAR)
      YY(I)=YY(I)+(PXY(I)*HXY+PYY(I)*HYY)*(ZY-YBAR)
      U=U+(PXU*HXX+PYU*HXY)*(ZX-XBAR)+(PXU*HXY+PYU*HYY)*(ZY-YBAR)
      V=V+(PXV*HXX+PYV*HXY)*(ZX-XBAR)+(PXV*HXY+PYV*HYY)*(ZY-YBAR)
C
C     UPDATE DRIVING NOISE ESTIMATE AND VELOCITY COVARIANCE SUBMATRIX
C
      IF(TAU.LE.0.) GO TO 1
      RI=RI+1.
```

```
      DI=HI
      DC=HC
      PI=0.
      PC=PI
      SQINV=PI
      U2=U*U
      V2=V*V
      UV=U*V
      SQ=SQRT(U2+V2)
      IF(SQ.GT.0.) SQINV=1./SQ
      QINV=SQINV*SQINV
      EX=ZX-XBAR
      EY=ZY-YBAR
      SIT=(U*EX+V*EY)*SQINV
      SCT=(U*EY-V*EX)*SQINV
      GXX=GXX-QXX*TAU+RXX
      GXY=GXY-QXY*TAU+RXY
      GYY=GYY-QYY*TAU+RYY
      GIT=(U2*GXX+2.*UV*GXY+V2*GYY)*QINV
      GCT=(V2*GXX-2.*UV*GXY+U2*GYY)*QINV
      ECI=(SIT*SIT-GIT)/TAU
      ECC=(SCT*SCT-GCT)/TAU
      IF(ECI.GT.EMAX) EMAX=ECI
      IF(ECC.GT.EMAX) EMAX=ECC
      IF(ECI.LT.-EMAX) ECI=-EMAX
      IF(ECC.LT.-EMAX) ECC=-EMAX
      CI=CI+(ECI-CI)/RI
      CC=CC+(ECC-CC)/RI
      HI=0.
      HC=0.
      IF(CI.GT.0.) HI=CI
      IF(CC.GT.0.) HC=CC
      QXX=(U2*HI+V2*HC)*QINV
      QXY=UV*(HI-HC)*QINV
      QYY=(V2*HI+U2*HC)*QINV
      DI=HI-DI
      DC=HC-DC
      IF(DI.GT.0.) PI=DI/(T(I)-T(1))
      IF(DC.GT.0.) PC=DC/(T(I)-T(1))
      IF(PI.GT..5*VELVAR) PI=.5*VELVAR
      IF(PC.GT..5*VELVAR) PC=.5*VELVAR
      PUU=PUU+(U2*PI+V2*PC)*QINV
      PUV=PUV+UV*(PI-PC)*QINV
      PVV=PVV+(V2*PI+U2*PC)*QINV
    1 CONTINUE
C
C                   TRACKER OUTPUT
C
C         I = OBSERVATION INDEX
C         T = TIME
C         (XX, YY) = CURRENT POSITION ESTIMATE IN X-Y COORDINATES
C         SMAJ = SEMIMAJOR AXIS OF 86 PERCENT CONTAINMENT ELLIPSE FOR
C                CURRENT POSITION
C         SMIN = SEMIMINOR AXIS OF CONTAINMENT ELLIPSE
C         TH = ORIENTATION OF SEMIMAJOR AXIS (DEG. CLOCKWISE FROM Y-AXIS)
C
```

```
      PRINT 8
      PRINT 12
      PRINT 11
      DO 3 I=1,N
    3 PRINT 7,T(I),XX(I),YY(I),SMAJ(I),SMIN(I),TH(I)
      PRINT 8
    7 FORMAT(25X,F10.2,5X,2F10.2,5X,3F10.2,/)
    8 FORMAT(20X,///)
   12 FORMAT(50X,17HTRACKED POSITIONS,//)
C
C                       TRACK SMOOTHER
C
      XS(N)=XX(N)
      YS(N)=YY(N)
      NM1=N-1
      DEN=T(N)-T(1)
      QXX=QXX+PUU*DEN
      QXY=QXY+PUV*DEN
      QYY=QYY+PVV*DEN
      DO 2 K=1,NM1
      I=N-K
      TAU=T(I+1)-T(I)
      P1=PXX(I)
      P2=PXY(I)
      P3=PYY(I)
      DEN=(P1+QXX*TAU)*(P3+QYY*TAU)-(P2+QXY*TAU)**2
      DENI=0.
      IF(DEN.GT.0.) DENI=1./DEN
      HXX=P1*(P3+QYY*TAU)-P2*(P2+QXY*TAU)
      HXY=P2*(P1+QXY*TAU)-P1*(P2+QXY*TAU)
      HYX=P2*(P3+QYY*TAU)-P3*(P2+QXY*TAU)
      HYY=P3*(P1+QXX*TAU)-P2*(P2+QXY*TAU)
      XS(I)=XX(I)+(HXX*(XS(I+1)-XX(I)-U*TAU)+HXY*(YS(I+1)-YY(I)-V*TAU))*
     1DENI
2     YS(I)=YY(I)+(HYX*(XS(I+1)-XX(I)-U*TAU)+HYY*(YS(I+1)-YY(I)-V*TAU))*
     1DENI
      SXX=PXX(N)
      SXY=PXY(N)
      SYY=PYY(N)
      C1=SXX+SYY
      C2=SQRT((SXX-SYY)**2+4.*SXY**2)
      C1=.5*(C1+C2)
      C2=C1-C2
      SMAJ(N)=2.*SQRT(C1)
      SMIN(N)=2.*SQRT(C2)
      IF(SXY.NE.0.) GO TO 73
      TH(N)=0.
      GO TO 53
   73 TH(N)=57.3*ATAN((SXX-C1)/SXY)+90.
   53 CONTINUE
      DO 5 L=2,N
      I=N-L+1
      TAU=T(I+1)-T(I)
```

```
      P1=PXX(I)
      P2=PXY(I)
      P3=PYY(I)
      A1=P1+QXX*TAU
      A2=P2+QXY*TAU
      A3=P3+QYY*TAU
      DET=A1*A3-A2*A2
      DETI=0.
      IF(DET.GT.0.) DETI=1./DET
      H1=A3*DETI
      H2=-A2*DETI
      H3=A1*DETI
      B1=P1*H1+P2*H2
      B2=P1*H2+P2*H3
      B3=P2*H1+P3*H2
      B4=P2*H2+P3*H3
      D1=SXX-A1
      D2=SXY-A2
      D3=SYY-A3
      E1=B1*D1+B2*D2
      E2=B1*D2+B2*D3
      E3=B3*D1+B4*D2
      E4=B3*D2+B4*D3
      SXX=P1+E1*B1+E2*B2
      SXY=P2+E1*B3+E2*B4
      SYY=P3+E3*B3+E4*B4
      C1=SXX+SYY
      C2=SQRT((SXX-SYY)**2+4.*SXY**2)
      C1=.5*(C1+C2)
      C2=C1-C2
      SMAJ(I)=2.*SQRT(C1)
      SMIN(I)=2.*SQRT(C2)
      IF(SXY.NE.0.) GO TO 74
      TH(I)=0.
      GO TO 5
   74 TH(I)=57.3*ATAN((SXX-C1)/SXY)+90.
    5 CONTINUE
C
C                    SMOOTHER OUTPUT
C
C          I = OBSERVATION INDEX
C          T = TIME
C          (XS, YS) = SMOOTHED POSITION IN X-Y COORDINATES
C          SMAJ = SEMIMAJOR AXIS OF 86 PERCENT CONTAINMENT ELLIPSE FOR
C                 SMOOTHED POSITION
C          SMIN = SEMIMINOR AXIS OF CONTAINMENT ELLIPSE
C          TH = ORIENTATION OF SEMIMAJOR AXIS (DEG. CLOCKWISE FROM Y-AXIS)
C
      PRINT 13
      PRINT 11
   13 FORMAT(50X,18HSMOOTHED POSITIONS,//)
      DO 4 I=1,N
    4 PRINT 7,T(I),XS(I),YS(I),SMAJ(I),SMIN(I),TH(I)
      END
```

DEPARTMENT OF THE NAVY

NAVAL RESEARCH LABORATORY
Washington, D.C. 20375

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

POSTAGE AND FEES PAID
DEPARTMENT OF THE NAVY
DoD-316
THIRD CLASS MAIL

U.S. MAIL