Appendices

Appendix A
The computer program

A-1 Source Code

The source code for the program for the Kalman filter, written with Delphi programming language

```
unit kalman;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

  // definition of the variables used

      x3d =array[1..40,1..40,1..40] of real;
     x2d =array[1..10,1..10] of real;
 rec=record
 values:x3d;
 row,col:integer;
 end;
var
  Form1: TForm1;
a,b,b1,c,cx,cx1,cxhat,s,N_1,g,M,z,z1,x,XDASH,w,u,wm,d,e,e1,f1,f2,f3,f
4,h,h1,h2,xhat,identity:rec;
   ncol,nrow,ncol2,i,j,k,m11,n11,iter,iteration,step,ms:integer;
   t,t1,t2,t3:textfile;
    nr1,nc1:integer;

implementation

{$R *.dfm}


// matrix add procedure

procedure add_mat(nr,nc,i0:integer;a1,b1:rec;var CC1:rec);
var
i1,j1:integer;
```

```
begin
CC1.ROW:=NR; CC1.col:=Nc;
for i1:= 1 to nr do
for j1:=1 to nc do
CC1.values[i0,i1,j1]:=a1.values[i0,i1,j1]+b1.values[i0,i1,j1];
 end;

 // matrix substraction procedure

procedure subt_mat(nr,nc,i0:integer;a1,b1:rec; var c1 :rec);
var
i1,j1:integer;
begin
C1.ROW:=NR; C1.col:=Nc;
for i1:= 1 to nr do
for j1:=1 to nc do
c1.values[i0,i1,j1]:=a1.values[i0,i1,j1]-b1.values[i0,i1,j1];
end;

// matrix transpose procedure

procedure transpose_mat(nr,nc,i0:integer;a1:rec; var c1:rec);
var
i1,j1:integer;
begin
C1.ROW:=Nc; C1.col:=Nr;
for i1:=1 to nr do
for j1:=1 to nc do
c1.values[i0,i1,j1]:=a1.values[i0,j1,i1];
end;

// matrix multiplication procedure

procedure mat_mul(nr,nc1,nc2,i0:integer;a1,b1:rec; var c1:rec);
var
i1,j1,k1:integer;
sum:real;
begin
C1.ROW:=nr; C1.col:=Nc2;
for i1:=1 to nr do
for j1:=1 to nc2 do BEGIN
sum:=0;
for k1:=1 to nc1 do
sum:= sum+a1.values[i0,i1,k1]*b1.values[i0,k1,j1];
c1.values[i0,i1,j1]:=sum;
END;
  end;

  // matrix inverse procedure

 procedure inverse_mat(nr,i0:integer;a1:rec; var c1:rec;deter:real);
var
i1,j1,k1:integer;sum:real;
q:rec;
begin
C1.ROW:=NR; C1.col:=Nr;
for i1:=1 to nr do begin
for j1:=1 to nr do begin
q.values[i0,i1,j1]:=a1.values[i0,i1,j1];
if (i1=j1) then q.values[i0,i1,j1+nr]:=1 else
q.values[i0,i1,j1+nr]:=0;
end;
```

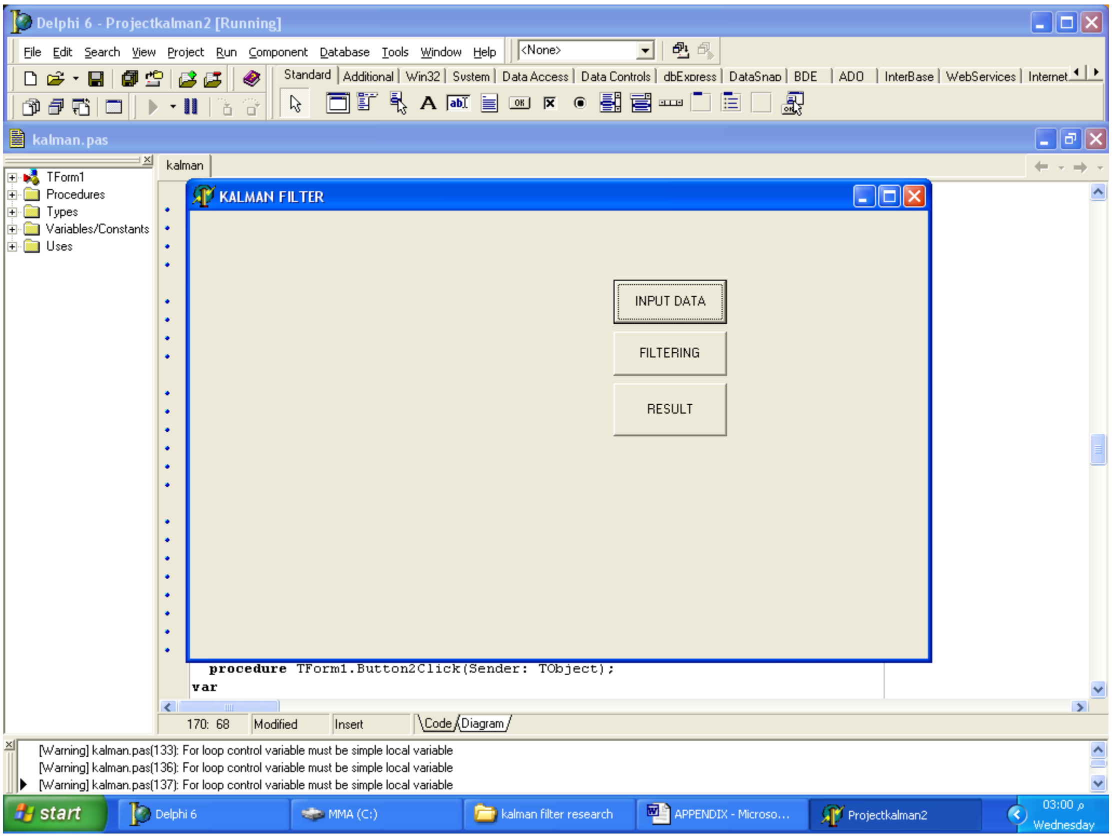
```
end;
for i1:=1 to nr do begin
for j1:=1 to nr do begin
if (i1<>j1) then begin
sum:=q.values[i0,j1,i1]/q.values[i0,i1,i1];
for k1:=1 to 2*nr do begin
q.values[i0,j1,k1]:= q.values[i0,j1,k1]-sum*q.values[i0,i1,k1];
end;
end;
end;
end;
for i1:=1 to nr do begin
for j1:=nr+1 to 2*nr do  begin
 c1.values[i0,i1,j1-nr]:= q.values[i0,i1,j1]/q.values[i0,i1,i1];
 end;
 end;
   end;

// idendtity matrix procedure

procedure identity_mat(nr,i0:integer;var ident:rec);
var
i1,j1:integer;
begin
  ident.row:=nr;   ident.col:=nr;
for i1:=1 to nr do
for j1:=1 to nr do
if i1=j1 then ident.values[i0,i1,j1]:=1 else
ident.values[i0,i1,j1]:=0;
 end;

 //input procedure

procedure TForm1.Button1Click(Sender: TObject);
VAR
M1,N1:INTEGER;
 begin
assignfile (t,'c:\input1.txt');
reset(t);

//input x

readln(t,m1,n1);x.row:=m1;x.col:=1;
for i:=1 to m1 do readln(t,x.values[1,i,1]);

//input w

W.row:=N1;W.col:=N1;
for i:=1 to W.row do
for j:=1 to W.col do
read(t,w.values[1,i,j]);

//input wm

WM.row:=M1;WM.col:=M1;
for i:=1 to WM.row do
for j:=1 to WM.COL do
 read(t,wm.values[1,i,j]);

//input N

n_1.row:=M1;n_1.col:=M1;
```

```
for i:=1 to n_1.row do
for j:=1 to n_1.col do
read(t,n_1.values[1,i,j]);

//input A

a.row:=n1;a.col:=M1;
for i:=1 to a.row do
for j:=1 to a.col do
 read(t,a.values[1,i,j]);

//input M

m.row:=m1; m.col:=m1;
for i:=1 to m.row do
for j:=1 to m.col do
read(t,M.values[1,i,j]);
  m11:=m1;n11:=n1;
 closefile(t);

 //input b

 assignfile (t2,'c:\input2.txt');
 reset(t2);
 readln(t2,iteration);
  b.row:=n1; b.col:=1;
for iter:=1 to iteration do
for i:=1 to n1 do readln(t2,b.values[iter,i,1]);
   closefile (t2);
  end;

// The main program

procedure TForm1.Button2Click(Sender: TObject);
var
deter:real;
begin
step:=0;
repeat
step:=step+1;
b1.row:=b.row;
b1.col:=b.col;
for i:=1 to b.row do
for j:=1 to b.col do
b1.values[1,i,j]:=b.values[step,i,j];

// predict the state vectot
mat_mul(m11,M11,1,1,m,x,xdash);
// predict the covarience matrix
mat_mul(m.row,M.col,n_1.col,1,m,n_1,d);
transpose_mat(m.row,m.col,1,m,e);
mat_mul(d.row,d.col,e.col,1,d,e,cx1);
add_mat(cx1.row,cx1.col,1,cx1,wm,cx);

//compute the gain matrix
transpose_mat(a.row,a.col,1,a,e1);
mat_mul(cx.row,cx.col,e1.col,1,cx,e1,f1);
mat_mul(a.row,a.col,f1.col,1,a,f1,f2);
add_mat(w.row,f2.col,1,w,f2,f3);
inverse_mat(f3.row,1,f3,f4,deter);
mat_mul(f1.row,f1.col,f4.col,1,f1,f4,g);

// compute the state vector
```

```
 mat_mul(a.row,a.col,xdash.col,1,a,xdash,h);
 subt_mat(b1.row,h.col,1,b1,h,h1);
 mat_mul(g.row,g.col,h1.col,1,g,h1,h2);
 add_mat(xdash.row,xdash.col,1,xdash,h2,xhat);

// compute the covarience matrix of the state vector
 mat_mul(g.row,g.col,a.col,1,g,a,z);
identity_mat(z.row,1,identity);
subt_mat(identity.row,z.col,1,identity,z,z1);
mat_mul(z1.row,z1.col,cx.col,1,z1,cx,cxhat);

// out put the final estimated values of the coordinates
assignfile(t1,'c:\ouput2.txt');
append(t1);
writeln(t1,xhat.values[1,1,1]:10:6,'   ',xhat.values[1,2,1]:10:6);
closefile(t1);
assignfile(t2,'c:\ouput3.txt');
append(t2);
writeln(t2,'                    ');
write(t2,step,'  ',n_1.VALUES[1,1,1]:10:6);
 closefile(t2);

 // prepare for the nexet iteration
  x.values[1,1,1]:=xhat.values[1,1,1];
 x.values[1,2,1]:=xhat.values[1,2,1];
  n_1:=cxhat;
 until step=iteration;
 end;
procedure TForm1.Button3Click(Sender: TObject);
var
i,J:integer;
begin
// The main output

assignfile(t,'c:\ouput1.txt');
rewrite(t);
writeln(t,'THE KALMAN FILTER');
writeln(t,'                             ');
writeln(t,'THE STATE VECTOR AT ',step ,' RECURSION');
 writeln(t,'                                 ');
 for i:=1 to XDASH.ROW do BEGIN
 for J:=1 to XDASH.col  do
 writeln(t,XDASH.VALUES[1,i,J]:10:6);
 END;
 writeln(t,'                          ');
writeln(t,'THE COVERIANCE MATRIX OF THE STATE VECTOR AT ', step,'
RECURSION');
 writeln(t,'                                      ');
for i:=1 to cx.row do BEGIN
for j:=1 to cx.col-1 do
 write(t,cx.VALUES[1,i,j]:10:6);
 writeLN(t,cx.VALUES[1,i,cx.COL]:10:6);
 END;
    writeln(t,'                         ');
writeln(t,'THE GAIN MATRIX');
writeln(t,'               ');
for i:=1 to G.row do BEGIN
for j:=1 to G.col-1 DO
 write(t,G.VALUES[1,i,j]:10:6);
 writeLN(t,G.VALUES[1,i,G.COL]:10:6);
 END;
  writeln(t,'                         ');
 writeln(t,'THE LEAST SQUARE ESTIMATE OF THE STATE VECTOR');
```

```
 writeln(t,'                   ');
for i:=1 to XHAT.row do BEGIN
for j:=1 to XHAT.col do
writeLN(t,XHAT.VALUES[1,i,J]:10:6);
END;
 writeln(t,'                          ');
writeln(t,'COVARIENCE MATRIX OF THE ESTIMATED STATE VECTOR ');
 writeln(t,'                          ');
for i:=1 to CXHAT.row do BEGIN
for j:=1 to CXHAT.col-1 DO
 write(t,CXHAT.VALUES[1,i,j]:10:6);
 writeLN(t,CXHAT.VALUES[1,i,CXHAT.COL]:10:6);
   END;
  closefile(t);

end;
end.
```

A-2 The Interactive desktop of the program



A-3 the initial data used

The initial state vector

1840624.748
537518.187
0
0.833333333333

The weight matrix

1 0
0 1

The model weight matrix

.414 0 0.0138 0
0 .414 0 0.0138 0
0.014 0 0.00047 0
0 0.014 0 0.00047

The initial variance covariance matrix

55.268658 0.000000 0.210857 0.000000
0.000000 55.268658 0.000000 0.210857
0.211196 0.000000 0.001843 0.000000

0.000000 0.211196 0.000000 0.001843

The design matrix

1 0 0 0
0 1 0 0

The transition matrix

1 0 60 0
0 1 0 60
0 0 1 0
0 0 0 1

A-4 the coordinates data used

True position data		Observed position data	
N	E	N	E
1840624.748	537518.187	1840619.748	537517.187
1840624.748	537568.187	1840628.748	537565.687
1840624.748	537618.187	1840614.748	537620.187
1840624.748	537668.187	1840622.748	537669.187
1840624.748	537718.187	1840620.748	537717.187
1840624.748	537768.187	1840636.748	537767.687
1840624.748	537818.187	1840626.748	537820.187
1840624.748	537868.187	1840630.748	537866.437
1840624.748	537918.187	1840634.748	537919.617
1840624.748	537968.187	1840614.748	537968.747
1840624.748	538018.187	1840626.748	538018.867
1840624.748	538068.187	1840619.748	538066.957
1840624.748	538118.187	1840636.748	538119.056
1840624.748	538168.187	1840634.108	538167.618
1840624.748	538218.187	1840625.645	538216.987
1840624.748	538268.187	1840634.378	538267.497
1840624.748	538318.187	1840635.108	538321.787
1840624.748	538368.187	1840619.058	538370.556
1840624.748	538418.187	1840617.098	538415.729
1840624.748	538468.187	1840618.158	538464.937
1840624.748	538518.187	1840612.058	538515.818
1840624.748	538568.187	1840627.117	538569.643
1840624.748	538618.187	1840615.09	538616.931
1840624.748	538668.187	1840623.512	538671.443

1840624.748	538718.187	1840629.317	538720.547
1840624.748	538768.187	1840619.065	538765.818
1840624.748	538818.187	1840618.803	538815.831
1840624.748	538868.187	1840632.428	538869.643
1840624.748	538918.187	1840617.152	538915.827
1840624.748	538968.187	1840629.308	538969.44
1840624.748	539018.187	1840630.438	539020.547
1840624.748	539068.187	1840634.385	539066.951
1840624.748	539118.187	1840635.108	539115.827
1840624.748	539168.187	1840632.308	539169.423
1840624.748	539218.187	1840619.109	539215.827
1840624.748	539268.187	1840632.204	539265.827
1840624.748	539318.187	1840633.008	539320.547
1840624.748	539368.187	1840618.765	539363.587
1840624.748	539418.187	1840628.388	539420.547
1840624.748	539468.187	1840615.983	539463.497

A-5 The Kalman filter estimation for the input data

kalmn filter estimation	
N	**E**
1840624.744	537568.143
1840624.749	537618.073
1840624.73	537667.982
1840624.725	537717.855
1840624.711	537767.68
1840624.764	537817.46
1840624.775	537867.202
1840624.815	537916.864
1840624.893	537966.489
1840624.799	538016.045
1840624.82	538065.536
1840624.757	538114.939
1840624.924	538164.3
1840625.067	538213.574
1840625.077	538262.772
1840625.253	538311.914
1840625.457	538361.086
1840625.314	538410.18
1840625.116	538459.111
1840624.937	538507.976
1840624.586	538556.825
1840624.659	538605.751
1840624.368	538654.572
1840624.341	538703.518
1840624.506	538752.425
1840624.318	538801.165
1840624.122	538849.906

1840624.427	538898.795
1840624.152	538947.551
1840624.352	538996.464
1840624.592	539045.44
1840624.986	539094.294
1840625.4	539143.13
1840625.687	539192.146
1840625.411	539241.041
1840625.699	539289.971
1840626.011	539339.14
1840625.699	539388.04
1840625.816	539437.282
1840625.388	539486.248

A-6 Sample of statistical output of the program

THE STATE VECTOR AT 40 RECURSION

1840626.978619
539469.361795
0.000000
0.833333

THE COVERIANCE MATRIX OF THE STATE VECTOR AT 40 RECURSION

3.821108 0.000000 0.047476 0.000000
0.000000 3.821108 0.000000 0.047476
0.047728 0.000000 0.000867 0.000000
0.000000 0.047728 0.000000 0.000867

THE GAIN MATRIX

0.792579 0.000000
0.000000 0.792579
0.009900 0.000000
0.000000 0.009900

THE LEAST SQUARE ESTIMATE OF THE STATE VECTOR

1840618.263724
539464.713483
-0.108854
0.775273

COVARIENCE MATRIX OF THE ESTIMATED STATE VECTOR

0.792579	0.000000	0.009847	0.000000
0.000000	0.792579	0.000000	0.009847
0.009900	0.000000	0.000397	0.000000
0.000000	0.009900	0.000000	0.000397

Appendix B
Numerical Examples

B-1 off shore navigation example

Referring to 5-1 and figure 5-1, consider a ship carrying out a fix each $\Delta t = 60$ second, assuming that the standard error of the ship's random acceleration $\sigma = 0.0002$ and the covariance matrix of each fix as follow:

$$w_i = \begin{bmatrix} 91.60 & 42.70 \\ 42.70 & 91.60 \end{bmatrix}$$

Then, using 5-10 we obtain

$$w_m^{-1} = \begin{bmatrix} 0.1296 & 0 & 0.00432 & 0 \\ 0 & 0.1296 & 0 & 0.00432 \\ 0.00432 & 0 & 0.000144 & 0 \\ 0 & 0.00432 & 0 & 0.000144 \end{bmatrix}$$

At i recursion we have the following values of the state vector and its covariance matrix

$$\hat{x}'_{i-1} = \begin{bmatrix} 15969.933 \\ 25030.638 \\ 2.92214 \\ 2.00528 \end{bmatrix} \begin{matrix} easting \\ northing \\ eastvelocity \\ northvelocity \end{matrix}$$

$$N_{i-1}^{-1} = C_{\hat{x}'_{i-1}} = \begin{bmatrix} 29.020576 & 11.740694 & 0.092973 & 0.029312 \\ 11.740694 & 20.661862 & 0.029312 & 0.072305 \\ 0.092973 & 0.029312 & 0.000655 & 0.000111 \\ 0.029312 & 0.072305 & 0.000111 & 0.000576 \end{bmatrix}$$

At the point i the observations are

$$E_i^o = 16145.292$$
$$N_i^o = 25158.442$$

<u>Filtering:</u>

First we predict the state vector using 4-55 as

$$\hat{x}'_i = \begin{bmatrix} 1 & 0 & 60 & 0 \\ 0 & 1 & 0 & 60 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 15969.933 \\ 25030.638 \\ 2.92214 \\ 2.00528 \end{bmatrix} = \begin{bmatrix} 16145.262 \\ 25150.955 \\ 2.92214 \\ 2.00528 \end{bmatrix}$$

Then we predict the covariance matrix of the state vector using 4-58 as

$$C_{\hat{x}'_i} = N_i'^{-1} = \begin{bmatrix} 42.66415 & 15.623718 & 0.136580 & 0.035686 \\ 15.623718 & 31.540946 & 0.035686 & 0.111173 \\ 0.136580 & 0.035686 & 0.000799 & 0.000111 \\ 0.035686 & 0.000073 & 0.000111 & 0.000720 \end{bmatrix}$$

Next we compute the gain matrix using 4-59

$$G = \begin{bmatrix} 0.336512 & -0.043163 \\ -0.043163 & 0.367242 \\ 0.001170 & -0.000351 \\ -0.000351 & 0.001419 \end{bmatrix}$$

The least squares estimate of the state vector using 4-60 is

$$\hat{x}_i = \begin{bmatrix} 16144.949 \\ 25153.703 \\ 2.91955 \\ 2.01590 \end{bmatrix}$$

With its covariance matrix given by

$$C_{\hat{x}_i} = \begin{bmatrix} 28.981496 & 11.727531 & 0.09215 & 0.028476 \\ 11.727531 & 20.632154 & 0.028476 & 0.071886 \\ 0.092159 & 0.028476 & 0.000652 & 0.000108 \\ 0.028476 & 0.071886 & 0.000108 & 0.000575 \end{bmatrix}$$

B-2 Traffic tracking example

Referring to 5-2 suppose we want to track a vehicle on a road. For example the truck in the images is moving with constant velocity and in each frame we can measure the position of a feature on the vehicle we want to track, so we can directly measure the position of the truck, but not its velocity.

Suppose the initial state vector as

$$\hat{x}_i' = \begin{bmatrix} 100 \\ 170 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} x\ coordinate \\ y\ coordinate \\ velocity\ in\ x\ direction \\ velocity\ in\ y\ direction \end{matrix}$$

And the initial covariance of the state vector as

$$N_{i-1}^{-1} = C_{\hat{x}_{i-1}'} = \begin{bmatrix} 9 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix}$$

The state update covariance is

$$w_m^{-1} = \begin{bmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0. & 0 & 0.25 \end{bmatrix}$$

The weight matrix is

$$w_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The transition matrix equals to

$$M_{i-1,i} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

First we predict the state vector using 4-55 as

$$\hat{x}_i' = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 100 \\ 170 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 100 \\ 170 \\ 0 \\ 0 \end{bmatrix}$$

Then we predict the covariance matrix of the state vector using 4-58 as

$$C_{\hat{x}_i'} = N_i'^{-1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 9 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix} + \begin{bmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0.25 \end{bmatrix}$$
$$= \begin{bmatrix} 34.25 & 0 & 25 & 0 \\ 0 & 34.35 & 0 & 25 \\ 25 & 0 & 35.25 & 0 \\ 0 & 25 & 0 & 25.25 \end{bmatrix}$$

At the point i the observations are

$$x_i^o = 103$$
$$y_i^o = 163$$

Next we compute the gain matrix using 4-59

$$G = \begin{bmatrix} 0.972 & 0 \\ 0 & 0.972 \\ 0.7090 & 0 \\ 0 & 0.709 \end{bmatrix}$$

The least squares estimate of the state vector using 4-60 is

$$\hat{x}_i = \begin{bmatrix} 100 \\ 170 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.972 & 0 \\ 0 & 0.972 \\ 0.709 & 0 \\ 0 & 0.709 \end{bmatrix} \left(\begin{bmatrix} 103 \\ 163 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 100 \\ 170 \\ 0 \\ 0 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 16144.949 \\ 25153.703 \\ 2.91955 \\ 2.01590 \end{bmatrix}$$

With its covariance matrix given by

$$C_{\hat{x}_i} = \begin{bmatrix} 34.25 & 0 & 25 & 0 \\ 0 & 34.25 & 0 & 25 \\ 25 & 0 & 25.25 & 0 \\ 0 & 25 & 0 & 25.25 \end{bmatrix} \begin{bmatrix} 0.972 & 0 \\ 0 & 0.972 \\ 0.709 & 0 \\ 0 & 0.709 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 34.25 & 0 & 25 & 0 \\ 0 & 34.25 & 0 & 25 \\ 25 & 0 & 25.25 & 0 \\ 0 & 25 & 0 & 25.25 \end{bmatrix}$$

$$= \begin{bmatrix} 28.981496 & 11.727531 & 0.09215 & 0.028476 \\ 11.727531 & 20.632154 & 0.028476 & 0.071886 \\ 0.092159 & 0.028476 & 0.000652 & 0.000108 \\ 0.028476 & 0.071886 & 0.000108 & 0.000575 \end{bmatrix}$$

REFERENCES AND BIBLIOGRAPHY

A., Bannister; S., Raymond & R., Baker (1992), “Surveying”, Longman Scientific & Technical Press.

Brown, R.G.; Hwang, P. (1996), “Introduction to Random Signals and Applied Kalman Filtering”, Second Edition, John Wiley, & Sons.

Cooper, M. A. R. (1987), “control surveys in civil engineering”, Department of Civil Engineering, the City University, London.

Cross, P.A (1983), "Advanced Least Squares Applied to Position Fixing", working paper no (6), north east London polytechnic publications.

Eiserloh, Peter, P. (2002), "Introduction to Kalman Filters and Applications", Naval Air Warfare Center, China Lake, CA.

Godaha, Saurabh (2003), "Non – linear Kalman filtering", Department of Geomatics Engineering, University of Calgary.

Hofmann-Wellenhof, B.; lichtenegger H. & Collins J. (1992) "GPS theory and practice", Springer-Verlag New York.

Lewis, F.L. (1986), "Optimal Estimation with an Introductory to Stochastic Control Theory", John Wiley & Sons.

Maybeck, P.S. (1979), "Stochastic Models, Estimation, and control", Vol. 1, Academic Press, New York.

Ryu, Jihan & Gerdes, J., Christian (2003), "Integrating Inertial Sensors with GPS for Vehicle", Dynamics Control Department of Mechanical Engineering, Design Division, Stanford University.

Salzmann, M. A. (1988), "Some Aspects of Kalman Filter, Department of Geodesy and Geomatics Engineering", University of New Brunswick.

Welch, G. & Bishop, G. (2003) "An Introduction to Kalman Filter", UNC-Chapel Hill, TR95-041.