

PART C

Attachments, References & Acknowledgements

Random error generation

```
clear all
close all
clc
load('TRACKS_10.mat')

errmax=[0.02 0.05 0.10 0.20 0.40];
for i=1:length(TRACKLIST_PROJ)
    for k=1:length(errmax)
        rmax=errmax(k);
        s1=['DATA(:,1)= ' char(TRACKLIST_PROJ(i)) ' .Est;']; eval(s1);
        s2=['DATA(:,2)= ' char(TRACKLIST_PROJ(i)) ' .Nord;']; eval(s2);
        s3=['DATA(:,3)= ' char(TRACKLIST_PROJ(i)) ' .H;']; eval(s3);
        [size1,size2]=size(DATA);
        err=randn(size1,2);
        DIFFX_err=max(err(:,1))-min(err(:,1));
        DIFFY_err=max(err(:,2))-min(err(:,2));
        DIFF=max(DIFFX_err,DIFFY_err);
        scale=(2*rmax)/DIFF;
        err=scale*err;
        DATA2(:,1)=DATA(:,1)+err(:,1);
        DATA2(:,2)=DATA(:,2)+err(:,2);
        DATA2(:,3)=DATA(:,3);
        name=char(TRACKLIST_PROJ(i));
        name(length(name))=[];
        name(length(name))=[];
        s4=[name num2str(rmax*100) ' .Est=DATA2(:,1);']; eval(s4);
        s5=[name num2str(rmax*100) ' .Nord=DATA2(:,2);']; eval(s5);
        s6=[name num2str(rmax*100) ' .H=DATA2(:,3);']; eval(s6);
        s7=[name num2str(rmax*100) ' .Random=err;']; eval(s7);
        clear DATA DATA2
    end
end
```


2 Local deviation angle

```

for k=1:length(TRACKLIST)
    s1=['DATA(:,1)=' char(TRACKLIST(k)) '.Est;']; eval(s1);
    s2=['DATA(:,2)=' char(TRACKLIST(k)) '.Nord;']; eval(s2);
    s3=['DATA(:,3)=' char(TRACKLIST(k)) '.H;']; eval(s3);
    [s1,s2]=size(DATA);
    for i=2:s1-1
        x1=DATA(i-1,1);
        y1=DATA(i-1,2);
        x2=DATA(i,1);
        y2=DATA(i,2);
        x3=DATA(i+1,1);
        y3=DATA(i+1,2);
        d1=sqrt((y2-y1)^2+(x2-x1)^2);
        d2=sqrt((y3-y2)^2+(x3-x2)^2);
        %    calcolo theta12
        deltaE=x2-x1;
        deltaN=y2-y1;
        th12=atan((x2-x1)/(y2-y1))*180/pi;
        if deltaN<0 && deltaE>0
            th12=th12+180;
        elseif deltaN>0 && deltaE<0
            th12=th12+360;
        elseif deltaN<0 && deltaE<0
            th12=th12+180;
        end
        %    calcolo theta23
        deltaE=x3-x2;
        deltaN=y3-y2;
        th23=atan((x3-x2)/(y3-y2))*180/pi;
        if deltaN<0 && deltaE>0
            th23=th23+180;
        elseif deltaN>0 && deltaE<0
            th23=th23+360;
        elseif deltaN<0 && deltaE<0
            th23=th23+180;
        end
        %    calcolo angolo di deviazione
        dev(i)=th23-th12;
    end
    dev(1)=NaN;
    dev(s1)=NaN;

    s4=[char(TRACKLIST(k)) '.Dev_Angle=dev'';']; eval(s4);
    clearvars -except ENDPTS STRPTS TRACKLIST TRACKLIST_PROJ TRACKS
end

```


3

Local curvature with polynomial fitting

```

clear all
close all
clc
load TRACKS_13.mat;
set(0,'defaulttextinterpreter','none');
wt1=waitbar(0,char(TRACKLIST(1)));
wt2=waitbar(0,'Attendere prego...');
pos_w1=get(wt1,'position');
pos_w2=[pos_w1(1) pos_w1(2)-2*pos_w1(4) pos_w1(3) pos_w1(4)];
set(wt2,'position',pos_w2);

sp=[2 4 6 8 10];
% sp=[8 10];
% sp=4;
gg=3;
tic
for a=1:length(TRACKLIST)
    s1=['DATA(:,1)= ' char(TRACKLIST(a)) '.Est;']; eval(s1);
    s2=['DATA(:,2)= ' char(TRACKLIST(a)) '.Nord;']; eval(s2);
    s3=['DATA(:,3)= ' char(TRACKLIST(a)) '.H;']; eval(s3);
    [s1,s2]=size(DATA);
    for b=1:length(sp)
        spa=sp(b);
        waitbar(a/length(TRACKLIST),wt1,[char(TRACKLIST(a)) ' - Fin. mobile:
' num2str(spa*2+1) ' epoche.']);
        for c=1+spa:s1-spa
            waitbar(c/(s1-2*spa),wt2);
            e=1;
            for d=c-spa:c+spa %coords acquisition around "i", width "sp"
                X(e)=DATA(d,1);
                Y(e)=DATA(d,2);
                Z(e)=DATA(d,3);
                e=e+1;
            end
            foX=X(1);
            foY=Y(1);
            foZ=Z(1);
            X=X-X(1); % starting point east as origin
            Y=Y-Y(1); % starting point north as origin
            Z=Z-Z(1); % starting point elevation as origin
            %local system application
            ang=atan2((Y(2*spa+1)-Y(1)),(X(2*spa+1))-X(1)); % rot. angle
            rotmatrix=[cos(ang) sin(ang); -sin(ang) cos(ang)]; %rot. matrix
            for f=1:length(X)
                NewPt(f,:)=(rotmatrix*[X(f);Y(f)]); %new rotated coords.
            end
            Xrot=NewPt(:,1);
            Yrot=NewPt(:,2);
        end
    end
end

```

```

    %polynomial evaluation in local system, and curvature
    P=polyfit(Xrot,Yrot,gg);      %fitting polynomial coeff. and errors
    P1=polyder(P);               %first derivative coeff.
    P2=polyder(P1);              %second derivative coeff.
    d1=polyval(P1,Xrot(length(Xrot)-spa)); %1st deriv. in central.pt.
    d2=polyval(P2,Xrot(length(Xrot)-spa)); %2nd deriv. in central.pt.
    Pcurv(b,c)=-d2/((1+d1^2)^(3/2)); %curvature evaluation
end
for g=1:spa
    Pcurv(b,g)=NaN; %initial point curvature denial
end
for m=s1-spa+1:s1
    Pcurv(b,m)=NaN; % final point curvature denial
end
end
Pcurv=Pcurv';
% curvature vector storage in various tracks
% s4=[char(TRACKLIST(a)) '.PolCurv=Pcurv;']; eval(s4);
% s4=[char(TRACKLIST(a)) '.PolCurv_2g_03Pts=Pcurv(:,1);']; eval(s4);
s5=[char(TRACKLIST(a)) '.PolCurv_3g_05Pts=Pcurv(:,1);']; eval(s5);
s6=[char(TRACKLIST(a)) '.PolCurv_3g_09Pts=Pcurv(:,2);']; eval(s6);
s7=[char(TRACKLIST(a)) '.PolCurv_3g_13Pts=Pcurv(:,3);']; eval(s7);
s8=[char(TRACKLIST(a)) '.PolCurv_3g_17Pts=Pcurv(:,4);']; eval(s8);
s9=[char(TRACKLIST(a)) '.PolCurv_3g_21Pts=Pcurv(:,5);']; eval(s9);
clearvars -except ENDPTS STRPTS TRACKLIST TRACKLIST_SHOT TRACKLIST_PROJ
TRACKS a b c d e f g m sp gg wt1 wt2
end
close(wt1)
close(wt2)
time=toc;
hours=fix(time/3600);
minutes=fix(time/60)-hours*60;
seconds=time-3600*hours-60*minutes;
msg=sprintf(['Sono stati elaborati correttamente ' num2str(length(TRACKLIST))
' tracciati.\nTempo necessario: ' num2str(hours) ' ore, ' num2str(minutes) '
minuti, ' num2str(seconds) ' secondi.']);
h=msgbox(msg,'Elaborazione conclusa','help');

```


A4

Local curvature with axis methodology

```
clear all
close all
clc
load TRACKS_13.mat;

for k=1:length(TRACKLIST)
    s1=['DATA(:,1)=' char(TRACKLIST(k)) '.Est;']; eval(s1);
    s2=['DATA(:,2)=' char(TRACKLIST(k)) '.Nord;']; eval(s2);
    s3=['DATA(:,3)=' char(TRACKLIST(k)) '.H;']; eval(s3);
    [s1,s2]=size(DATA);
    for i=2:s1-1
        x1=DATA(i-1,1);
        y1=DATA(i-1,2);
        z1=DATA(i-1,3);
        x2=DATA(i,1);
        y2=DATA(i,2);
        z2=DATA(i,3);
        x3=DATA(i+1,1);
        y3=DATA(i+1,2);
        z3=DATA(i+1,3);
        m3=(x1-x2)/(y2-y1); % 1st segment angular coefficient
        m4=(x2-x3)/(y3-y2); % 2nd segment angular coefficient
        xmp1=(x1+x2)/2; % medium first point coords. (x)
        ymp1=(y1+y2)/2; % medium first point coords. (y)
        xmp2=(x2+x3)/2; % medium second point coords. (x)
        ymp2=(y2+y3)/2; % medium second point coords. (y)
        k1=-m3*xmp1+ymp1;
        k2=-m4*xmp2+ymp2;
        CtPtx=(k2-k1)/(m3-m4);
        CtPty=m3*CtPtx+k1;
        r1=sqrt((CtPtx-x1)^2+(CtPty-y1)^2);
        r2=sqrt((CtPtx-x2)^2+(CtPty-y2)^2);
        r3=sqrt((CtPtx-x3)^2+(CtPty-y3)^2);
        r=(r1+r2+r3)/3;
        curv(i)=1/r;
        % curve orientation evaluation
        th12=atan((x2-x1)/(y2-y1))*180/pi; % azimuth 1st to 2nd point
        if (x2-x1)>0 && (y2-y1)<0
            th12=th12+180;
        elseif (x2-x1)<0 && (y2-y1)<0
            th12=th12+180;
        elseif (x2-x1)<0 && (y2-y1)>0
            th12=th12+360;
        end
        th23=atan((x3-x2)/(y3-y2))*180/pi; % azimuth 2nd to 3rd point
        if (x3-x2)>0 && (y3-y2)<0
            th23=th23+180;
        elseif (x3-x2)<0 && (y3-y2)<0
            th23=th23+180;
        elseif (x3-x2)<0 && (y3-y2)>0
            th23=th23+360;
        end
    end
end
```

```

end
if th12>270 && th23<90    %Northround correction
    th23=th23+360;
end
if th12>th23              % curve direction assignment
    curv(i)=-curv(i);
end
end
curv(1)=NaN;
curv(s1)=NaN;
s4=[char(TRACKLIST(k)) '.CircCurv=curv'';']; eval(s4);
clear curv DATA
end

```

5

Local curvature and deviation angle statistics

```

clear all
close all
clc
load TRACKS_14.mat;

for a=1:length(TRACKLIST_PROJ)
    clear DATA sq;
    s=char(TRACKLIST_PROJ(a));
    step=s(15);
    if s(16)~='_'
        step=[step s(16)];
    end
    if s(17)~='_' && s(16)~='_'
        step=[step s(17)];
    end
    if s(18)~='_' && s(17)~='_' && s(16)~='_'
        step=[step s(18)];
    end
    step=str2num(step)*0.01;
    s1=['DATA(:,1)=' s '.PolCurv_3g_05Pts;']; eval(s1);
    s2=['DATA(:,2)=' s '.PolCurv_3g_09Pts;']; eval(s2);
    s3=['DATA(:,3)=' s '.PolCurv_3g_13Pts;']; eval(s3);
    s4=['DATA(:,4)=' s '.PolCurv_3g_17Pts;']; eval(s4);
    s5=['DATA(:,5)=' s '.PolCurv_3g_21Pts;']; eval(s5);
    s6=['DATA(:,6)=' s '.PolCurv_2g_03Pts;']; eval(s6);
    s7=['DATA(:,7)=' s '.PolCurv_2g_05Pts;']; eval(s7);
    s8=['DATA(:,8)=' s '.PolCurv_2g_09Pts;']; eval(s8);
    s9=['DATA(:,9)=' s '.PolCurv_2g_13Pts;']; eval(s9);
    s10=['DATA(:,10)=' s '.PolCurv_2g_17Pts;']; eval(s10);
    s11=['DATA(:,11)=' s '.PolCurv_2g_21Pts;']; eval(s11);
    s12=['DATA(:,12)=' s '.CircCurv;']; eval(s12);
    [size1,size2]=size(DATA);
    for b=1:size2
        prog=NaN;
        for c=1:size1
            if c==1
                prog=0;
            else
                prog=(step*(c-1));
            end
            found=0;
            d=1;
            while found==0
                if REAL_CURV(d,1)>prog
                    x2=REAL_CURV(d,1);
                    y2=REAL_CURV(d,2);
                    x1=REAL_CURV(d-1,1);
                    y1=REAL_CURV(d-1,2);
                    found=1;
                end
            end
        end
    end
end

```

```

        d=d+1;
    end
    curvsperim=((prog-x1)*(y2-y1)/(x2-x1))+y1;
    curvteor=DATA(c,b);
    err(c,b)=abs(curvsperim-curvteor);
%     err(c,b)=abs(1/curvsperim-1/curvteor);
    end
end
s1=[s '.Errors_Curv=err;']; eval(s1);
clear err
end

%-----

clear all
close all
clc
load TRACKS_14.mat;

for a=1:length(TRACKLIST_PROJ)
    s1=['DATA=' char(TRACKLIST_PROJ(a)) '.Errors_Curv;']; eval(s1);
    [size1,size2]=size(DATA);
    for b=1:size2
        ErrParam(b,1)=nanmean(DATA(:,b));
        ErrParam(b,2)=nanstd(DATA(:,b));
        ErrParam(b,3)=nanmax(DATA(:,b));
        ErrParam(b,4)=nanmin(DATA(:,b));
    end
    s2=[char(TRACKLIST_PROJ(a)) '.ErrParam_Curv=ErrParam;']; eval(s2);
end

%-----

clear all
close all
clc
load TRACKS_14.mat;

for a=1:length(TRACKLIST_PROJ)
    clear DATA sq;
    s=char(TRACKLIST_PROJ(a));
    step=s(15);
    if s(16)~='_'
        step=[step s(16)];
    end
    if s(17)~='_' && s(16)~='_'
        step=[step s(17)];
    end
    if s(18)~='_' && s(17)~='_' && s(16)~='_'
        step=[step s(18)];
    end
    step=str2num(step)*0.01;
    s1=['DATA=' s '.Dev_Angle;']; eval(s1);
    size1=length(DATA);
    for c=1:size1
        if c==1
            prog=0;
        else
            prog=(step*(c-1));
        end
        found=0;

```

```

d=1;
if step==0.5
    stk=2;
elseif step==1
    stk=3;
elseif step==2
    stk=4;
elseif step==5
    stk=5;
elseif step==10
    stk=6;
elseif step==15
    stk=7;
elseif step==20
    stk=8;
end
while found==0
    if REAL_DEV(d,1)>prog
        x2=REAL_DEV(d,1);
        y2=REAL_DEV(d,stk);
        x1=REAL_DEV(d-1,1);
        y1=REAL_DEV(d-1,stk);
        found=1;
    end
    d=d+1;
end
devsperim=((prog-x1)*(y2-y1)/(x2-x1))+y1;
devteor=DATA(c);
err(c)=abs(devsperim-devteor);
%     err(c,b)=abs(1/curvsperim-1/curvteor);
end
s1=[s '.Errors_Dev=err'';']; eval(s1);
clear err
end

%-----

clear all
close all
clc
load TRACKS_14.mat;

for a=1:length(TRACKLIST_PROJ)
    s1=['DATA=' char(TRACKLIST_PROJ(a)) '.Errors_Dev;']; eval(s1);
    ErrParam(1)=nanmean(DATA);
    ErrParam(2)=nanstd(DATA);
    ErrParam(3)=nanmax(DATA);
    ErrParam(4)=nanmin(DATA);
    s2=[char(TRACKLIST_PROJ(a)) '.ErrParam_Dev=ErrParam'';']; eval(s2);
end

%-----

```


A6 Design-based grouping

```

clear all
close all
clc
load TRACKS_14.mat;
set(0, 'defaulttextinterpreter', 'none');
wt1=waitbar(0, char(TRACKLIST_PROJ(1)));
wt2=waitbar(0, 'Attendere prego...');
pos_w1=get(wt1, 'position');
pos_w2=[pos_w1(1) pos_w1(2)-2*pos_w1(4) pos_w1(3) pos_w1(4)];
set(wt2, 'position', pos_w2);

tic
for a=1:length(TRACKLIST_PROJ)
s=char(TRACKLIST_PROJ(a));
s1=[s '.PROJ_SEGM.ABSTRACT=zeros(25,7);']; eval(s1);
s11=[s '.PROJ_SEGM.ABSTRACT(:,1)=linspace(1,25,25)'';']; eval(s11);
waitbar(a/length(TRACKLIST_PROJ), wt1, char(TRACKLIST_PROJ(a)));
for b=1:length(GEOM_VERTEX)
waitbar(b/length(GEOM_VERTEX), wt2);
x1=GEOM_VERTEX(b,1); %B-th element start point (x1)
y1=GEOM_VERTEX(b,2); %B-th element start point (y1)
x2=GEOM_VERTEX(b,3); %B-th element end point (x2)
y2=GEOM_VERTEX(b,4); %B-th element end point (y2)
closerSTART=9999999999999999999999999999999; %starting distance = infinite
closerEND=9999999999999999999999999999999; %starting distance = infinite
STARTpos=0;
ENDpos=0;
s2=['lung=length(' s '.Est);']; eval(s2);
for c=1:lung
s3=['xtr=' s '.Est(c);']; eval(s3); %c-th track point
s4=['ytr=' s '.Nord(c);']; eval(s4); %c-th track point
dist1=sqrt((xtr-x1)^2+(ytr-y1)^2); %b-th point distance
dist2=sqrt((xtr-x2)^2+(ytr-y2)^2); %b-th point distance
if dist1<closerSTART
STARTpos=c;
closerSTART=dist1;
end
if dist2<closerEND
ENDpos=c;
closerEND=dist2;
end
end
s5=[s '.PROJ_SEGM.ABSTRACT(b,2)=STARTpos;']; eval(s5);
s6=[s '.PROJ_SEGM.ABSTRACT(b,3)=ENDpos;']; eval(s6);
s7=[s '.PROJ_SEGM.ABSTRACT(b,4)=' s '.Est(STARTpos);']; eval(s7);
s8=[s '.PROJ_SEGM.ABSTRACT(b,5)=' s '.Nord(STARTpos);']; eval(s8);
s9=[s '.PROJ_SEGM.ABSTRACT(b,6)=' s '.Est(ENDpos);']; eval(s9);
s10=[s '.PROJ_SEGM.ABSTRACT(b,7)=' s '.Nord(ENDpos);']; eval(s10);
for d=STARTpos:ENDpos

```

```

        s11=[s '.PROJ_SEGM.ELEMENT_' num2str(b) '(d-STARTpos+1,1)=' s
'.Est(d);']; eval(s11);
        s12=[s '.PROJ_SEGM.ELEMENT_' num2str(b) '(d-STARTpos+1,2)=' s
'.Nord(d);']; eval(s12);
    end
end
end
close(wt1)
close(wt2)
time=toc;
hours=fix(time/3600);
minutes=fix(time/60)-hours*60;
seconds=time-3600*hours-60*minutes;
msg=sprintf(['Sono stati elaborati correttamente '
num2str(length(TRACKLIST_PROJ)) ' tracciati.\nTempo necessario: '
num2str(hours) ' ore, ' num2str(minutes) ' minuti, ' num2str(seconds) '
secondi.']);
h=msgbox(msg,'Elaborazione conclusa','help');

```


A 7 Least squares, Landau and Huber fitting

```

clear all
close all
clc
load TRACKS_15.mat;

for a=1:length(TRACKLIST_PROJ)
    s=char(TRACKLIST_PROJ(a));
    for b=1:length(GEOMETRY) %tangents/curve 1st research cycle
        if strcmp(GEOMETRY(b),'Rettifilo')==1
            s1=['X=' s '.PROJ_SEGM.ELEMENT_' num2str(b) '(:,1)']; eval(s1);
            s2=['Y=' s '.PROJ_SEGM.ELEMENT_' num2str(b) '(:,2)']; eval(s2);
            if length(X)>2
                X=X-386887.4198; %local coordinates application
                Y=Y-4979559.0127; %local coordinates application
                A=[X ones(length(X),1)]; %design matrix for tangents
                T=Y; %tangent known term vector
                P=eye(length(X)); %weight matrix (identity)
            else
                s3=[s '.PROJ_SEGM=rmfield(' s '.PROJ_SEGM,'ELEMENT_'
num2str(b) ' ');]; eval(s3); %erase old array structure
                s4=[s '.PROJ_SEGM.El_' num2str(b) '.Est=X+386887.4198;'];
eval(s4); %storing x in "Est" vector
                s5=[s '.PROJ_SEGM.El_' num2str(b) '.Nord=Y+4979559.0127;'];
eval(s5); %storing y in "Nord" vector
                s6=[s '.PROJ_SEGM.El_' num2str(b) '.LS.Error=' 'Not enough
points';']; eval(s6); %saving a, b in Par vector
            end
        elseif strcmp(GEOMETRY(b),'Curva')==1
            s1=['X=' s '.PROJ_SEGM.ELEMENT_' num2str(b) '(:,1)']; eval(s1);
            s2=['Y=' s '.PROJ_SEGM.ELEMENT_' num2str(b) '(:,2)']; eval(s2);
            if length(X)>3
                X=X-386887.4198;
                Y=Y-4979559.0127;
                A=[X Y ones(length(X),1)]; %design matrix on curves
                T=-X.^2-Y.^2; %known term on curves
                P=eye(length(X)); %weight matrix (identity)
            else
                s3=[s '.PROJ_SEGM=rmfield(' s '.PROJ_SEGM,'ELEMENT_'
num2str(b) ' ');]; eval(s3); %erase old array structure
                s4=[s '.PROJ_SEGM.El_' num2str(b) '.Est=X+386887.4198;'];
eval(s4); %storing x in "Est" vector
                s5=[s '.PROJ_SEGM.El_' num2str(b) '.Nord=Y+4979559.0127;'];
eval(s5); %storing y in "Nord" vector
                s6=[s '.PROJ_SEGM.El_' num2str(b) '.LS.Error=' 'Not enough
points';']; eval(s6); %error message saving
            end
        end
        if exist('A','var')
            [x,Cxx,v]=LS(A,T,P);
            if length(x)==2 %if is tangent (2 parameters: a, b)

```

```

        s7=[s '.PROJ_SEGM=rmfield(' s '.PROJ_SEGM,'ELEMENT_'
num2str(b) '');']; eval(s7); %erase old array structure
        s8=[s '.PROJ_SEGM.El_' num2str(b) '.Est=X+386887.4198;'];
eval(s8); %storing x in "Est" vector
        s9=[s '.PROJ_SEGM.El_' num2str(b) '.Nord=Y+4979559.0127;'];
eval(s9); %storing y in "Nord" vector
        s10=[s '.PROJ_SEGM.El_' num2str(b) '.LS.Par=x;']; eval(s10);
%saving a,b in Par vector
        s11=[s '.PROJ_SEGM.El_' num2str(b) '.LS.Cxx=Cxx;'];
eval(s11); % saving a,b accuracy in vector Prec
        s12=[s '.PROJ_SEGM.El_' num2str(b) '.LS.v=v;']; eval(s12);
%saving accuracy vector
        elseif length(x)==3 %if is curve (3 parameters: a,b and c)
        s7=[s '.PROJ_SEGM=rmfield(' s '.PROJ_SEGM,'ELEMENT_'
num2str(b) '');']; eval(s7); %erase old array structure
        s8=[s '.PROJ_SEGM.El_' num2str(b) '.Est=X+386887.4198;'];
eval(s8); %storing x in "Est" vector
        s9=[s '.PROJ_SEGM.El_' num2str(b) '.Nord=Y+4979559.0127;'];
eval(s9); %storing y in "Nord" vector
        s10=[s '.PROJ_SEGM.El_' num2str(b) '.LS.Par=x;']; eval(s10);
        %saving a,b,c in Par vector
        s11=[s '.PROJ_SEGM.El_' num2str(b) '.LS.Cxx=Cxx;'];
eval(s11); % saving a,b,c accuracy in vector Prec
        s12=[s '.PROJ_SEGM.El_' num2str(b) '.LS.v=v;']; eval(s12);
%saving accuracy vector
        Cen=[(-x(1)/2)+386887.4198; (-x(2)/2)+4979559.0127];
%ctr.coord
        Rad=sqrt((-x(1)/2)^2+(-x(2)/2)^2-x(3)); %radius=Xc^2+Yc^2-c
        s13=[s '.PROJ_SEGM.El_' num2str(b) '.LS.Cen=Cen;'];
eval(s13); %saving center coords
        s14=[s '.PROJ_SEGM.El_' num2str(b) '.LS.Rad=Rad;'];
eval(s14); %saving radius
        end
    end
    clear A T P x Cxx v X Y Cen Rad
end
for d=1:length(GEOMETRY) %2nd cycle to search clothoids
    if strcmp(GEOMETRY(d),'Clotoide')==1
        s15=['X=' s '.PROJ_SEGM.ELEMENT_' num2str(d) '(:,1);'];
eval(s15);
        s16=['Y=' s '.PROJ_SEGM.ELEMENT_' num2str(d) '(:,2);'];
eval(s16);
        if strcmp(GEOMETRY(d-1),'Rettifilo')==1
            s17=['exRett=isfield(' s '.PROJ_SEGM.El_' num2str(d-1)
'.LS','Par');']; eval(s17);
            s18=['exCurv=isfield(' s '.PROJ_SEGM.El_' num2str(d+1)
'.LS','Rad');']; eval(s18);
            if exCurv==1 && exRett==1
                s19=['a=' s '.PROJ_SEGM.El_' num2str(d-1) '.LS.Par(1);'];
eval(s19);
                b=-1;
                s20=['c=' s '.PROJ_SEGM.El_' num2str(d-1) '.LS.Par(2);'];
eval(s20);
                s21=['R=' s '.PROJ_SEGM.El_' num2str(d+1) '.LS.Rad;'];
eval(s21);
                s22=['Cen=' s '.PROJ_SEGM.El_' num2str(d+1) '.LS.Cen;'];
eval(s22);
            end
        elseif strcmp(GEOMETRY(d+1),'Rettifilo')==1
            s17=['exRett=isfield(' s '.PROJ_SEGM.El_' num2str(d+1)
'.LS','Par');']; eval(s17);
            s18=['exCurv=isfield(' s '.PROJ_SEGM.El_' num2str(d-1)
'.LS','Rad');']; eval(s18);

```

```

        if exCurv==1 && exRett==1
            s19=['a=' s '.PROJ_SEG.M.E1_' num2str(d+1) '.LS.Par(1);'];
eval(s19);

            b=-1;
            s20=['c=' s '.PROJ_SEG.M.E1_' num2str(d+1) '.LS.Par(2);'];
eval(s20);

            s21=['R=' s '.PROJ_SEG.M.E1_' num2str(d-1) '.LS.Rad;'];
eval(s21);

            s22=['Cen=' s '.PROJ_SEG.M.E1_' num2str(d-1) '.LS.Cen;'];
eval(s22);

        end
    end
    if exist('Cen','var')
        Xm=Cen(1)-386887.4198;
        Ym=Cen(2)-4979559.0127;
        dist=abs((a*Xm+b*Ym+c)/sqrt(a^2+b^2)); %line-point distance
        deltaR=dist-R; %"scostamento" evaluation
        if deltaR>0
            A=nthroot(24*(R^3)*deltaR,4); %scale factor evaluation
            s23=[s '.PROJ_SEG.M.rmfield(' s '.PROJ_SEG.M,'ELEMENT_'
num2str(d) ''');']; eval(s23); %erase old array structure
            s24=[s '.PROJ_SEG.M.E1_' num2str(d) '.Est=X;']; eval(s24);
            %storing x in "Est" vector
            s25=[s '.PROJ_SEG.M.E1_' num2str(d) '.Nord=Y;'];
eval(s25); %storing y in "Nord" vector
            s26=[s '.PROJ_SEG.M.E1_' num2str(d) '.LS.A=A;'];
eval(s26); %scale factor storage
            s27=[s '.PROJ_SEG.M.E1_' num2str(d) '.LS.dR=deltaR;'];
eval(s27); %"scostamento" storage
        else
            s23=[s '.PROJ_SEG.M.rmfield(' s '.PROJ_SEG.M,'ELEMENT_'
num2str(d) ''');']; eval(s23); %cancello la vecchia struttura dell'array
            s24=[s '.PROJ_SEG.M.E1_' num2str(d) '.Est=X;']; eval(s24);
            %storing x in "Est" vector
            s25=[s '.PROJ_SEG.M.E1_' num2str(d) '.Nord=Y;'];
eval(s25); %storing y in "Nord" vector
            s26=[s '.PROJ_SEG.M.E1_' num2str(d) '.LS.Error=''deltaR is
negative!''']; eval(s26); %scale factor storage
        end
    else
        s23=[s '.PROJ_SEG.M.rmfield(' s '.PROJ_SEG.M,'ELEMENT_'
num2str(d) ''');']; eval(s23); %erase old array structure
        s24=[s '.PROJ_SEG.M.E1_' num2str(d) '.Est=X;']; eval(s24);
        %storing x in "Est" vector
        s25=[s '.PROJ_SEG.M.E1_' num2str(d) '.Nord=Y;']; eval(s25);
        %storing y in "Nord" vector
        s26=[s '.PROJ_SEG.M.E1_' num2str(d) '.LS.Error=''Adjacent
elements not defined properly''']; eval(s26); %error storage
    end
end
clear a b c R Cen Xm Ym dist deltaR A
end
end

%-----

```

```

clear all
close all
clc
load TRACKS_18.mat;

for a=1:length(TRACKLIST_SURV)
    s=char(TRACKLIST_SURV(a));
    for b=3:length(GEOMETRY) %tangents and curve research
        if strcmp(GEOMETRY(b),'Rettifilo')==1
            s1=['X=' s '.PROJ_SEGM.El_' num2str(b) '.Est;']; eval(s1);
            s2=['Y=' s '.PROJ_SEGM.El_' num2str(b) '.Nord;']; eval(s2);
            if length(X)>2
                X=X-386887.4198; %local coordinates system
                Y=Y-4979559.0127; %local coordinates system
                A=[X ones(length(X),1)]; %tangent design matrix
                T=Y; %tangent known vector
                P=eye(length(X)); %weight matrix (identity)
            else
                s6=[s '.PROJ_SEGM.El_' num2str(b) '.HB.Error=' 'Not enough
points';']; eval(s6); %error message: too few points
            end
            elseif strcmp(GEOMETRY(b),'Curva')==1
                s1=['X=' s '.PROJ_SEGM.El_' num2str(b) '.Est;']; eval(s1);
                s2=['Y=' s '.PROJ_SEGM.El_' num2str(b) '.Nord;']; eval(s2);
                if length(X)>3
                    X=X-386887.4198;
                    Y=Y-4979559.0127;
                    A=[X Y ones(length(X),1)]; %curves design matrix
                    T=-X.^2-Y.^2; %curves known vector
                    P=eye(length(X)); %weight matrix (identity)
                else
                    s6=[s '.PROJ_SEGM.El_' num2str(b) '.HB.Error=' 'Not enough
points';']; eval(s6); %error message: too few points
                end
            end
            if exist('A','var')
                [x,stats]=huber(A,T);
                [size1,size2]=size(A);
                if size2==2 %tangent: design matrix has 2 columns (a,b)
                    s10=[s '.PROJ_SEGM.El_' num2str(b) '.HB.Par=x;']; eval(s10);
                    %saving a,b in Par vector
                    s11=[s '.PROJ_SEGM.El_' num2str(b) '.HB.Stats=stats;'];
                    eval(s11); %saving statistics in vector Stats
                elseif size2==3 %curve: design matrix has 3 columns (a,b,c)
                    s10=[s '.PROJ_SEGM.El_' num2str(b) '.HB.Par=x;']; eval(s10);
                    %saving a,b,c in Par vector
                    s11=[s '.PROJ_SEGM.El_' num2str(b) '.HB.Stats=stats;'];
                    eval(s11); %saving statistics in vector Stats
                    Cen=[(-x(1)/2)+386887.4198;(-x(2)/2)+4979559.0127];
                    %ctr.coord
                    Rad=sqrt((-x(1)/2)^2+(-x(2)/2)^2-x(3)); %radius=Xc^2+Yc^2-c
                    s13=[s '.PROJ_SEGM.El_' num2str(b) '.HB.Cen=Cen;'];
                    eval(s13); %center coordinate storage
                    s14=[s '.PROJ_SEGM.El_' num2str(b) '.HB.Rad=Rad;'];
                    eval(s14); %radius storage
                end
            end
            clear A T P stats x X Y Cen Rad
        end
    for d=3:length(GEOMETRY) %2nd cycle searching clothoids
        if strcmp(GEOMETRY(d),'Clotoide')==1
            s15=['X=' s '.PROJ_SEGM.El_' num2str(d) '.Est;']; eval(s15);

```

```

s16=['Y=' s '.PROJ_SEG.M.El_' num2str(d) '.Nord;']; eval(s16);
if strcmp(GEOMETRY(d-1),'Rettifilo')==1
    s17=['exRett=isfield(' s '.PROJ_SEG.M.El_' num2str(d-1)
'.HB, 'Par');']; eval(s17);
    s18=['exCurv=isfield(' s '.PROJ_SEG.M.El_' num2str(d+1)
'.HB, 'Rad');']; eval(s18);
    if exCurv==1 && exRett==1
        s19=['a=' s '.PROJ_SEG.M.El_' num2str(d-1) '.HB.Par(1);'];
eval(s19);
        b=-1;
        s20=['c=' s '.PROJ_SEG.M.El_' num2str(d-1) '.HB.Par(2);'];
eval(s20);
        s21=['R=' s '.PROJ_SEG.M.El_' num2str(d+1) '.HB.Rad;'];
eval(s21);
        s22=['Cen=' s '.PROJ_SEG.M.El_' num2str(d+1) '.HB.Cen;'];
eval(s22);
    end
elseif strcmp(GEOMETRY(d+1),'Rettifilo')==1
    s17=['exRett=isfield(' s '.PROJ_SEG.M.El_' num2str(d+1)
'.HB, 'Par');']; eval(s17);
    s18=['exCurv=isfield(' s '.PROJ_SEG.M.El_' num2str(d-1)
'.HB, 'Rad');']; eval(s18);
    if exCurv==1 && exRett==1
        s19=['a=' s '.PROJ_SEG.M.El_' num2str(d+1) '.HB.Par(1);'];
eval(s19);
        b=-1;
        s20=['c=' s '.PROJ_SEG.M.El_' num2str(d+1) '.HB.Par(2);'];
eval(s20);
        s21=['R=' s '.PROJ_SEG.M.El_' num2str(d-1) '.HB.Rad;'];
eval(s21);
        s22=['Cen=' s '.PROJ_SEG.M.El_' num2str(d-1) '.HB.Cen;'];
eval(s22);
    end
end
if exist('Cen','var')
    Xm=Cen(1)-386887.4198;
    Ym=Cen(2)-4979559.0127;
    dist=abs((a*Xm+b*Ym+c)/sqrt(a^2+b^2)); %line-point distance
    deltaR=dist-R; %"scostamento" evaluation
    if deltaR>0
        A=nthroot(24*(R^3)*deltaR,4); %scale factor evaluation
        s26=[s '.PROJ_SEG.M.El_' num2str(d) '.HB.A=A;'];
eval(s26); %scale factor storage
        s27=[s '.PROJ_SEG.M.El_' num2str(d) '.HB.dR=deltaR;'];
eval(s27); %"scostamento" storage
    else
        s26=[s '.PROJ_SEG.M.El_' num2str(d) '.HB.Error='deltaR is
negative!';']; eval(s26); %saving scale factor
    end
    else
        s26=[s '.PROJ_SEG.M.El_' num2str(d) '.HB.Error='Adjacent
elements not defined properly!';']; eval(s26); %error communication
    end
end
clear a b c R Cen Xm Ym dist deltaR A
end
end
end
%-----

```

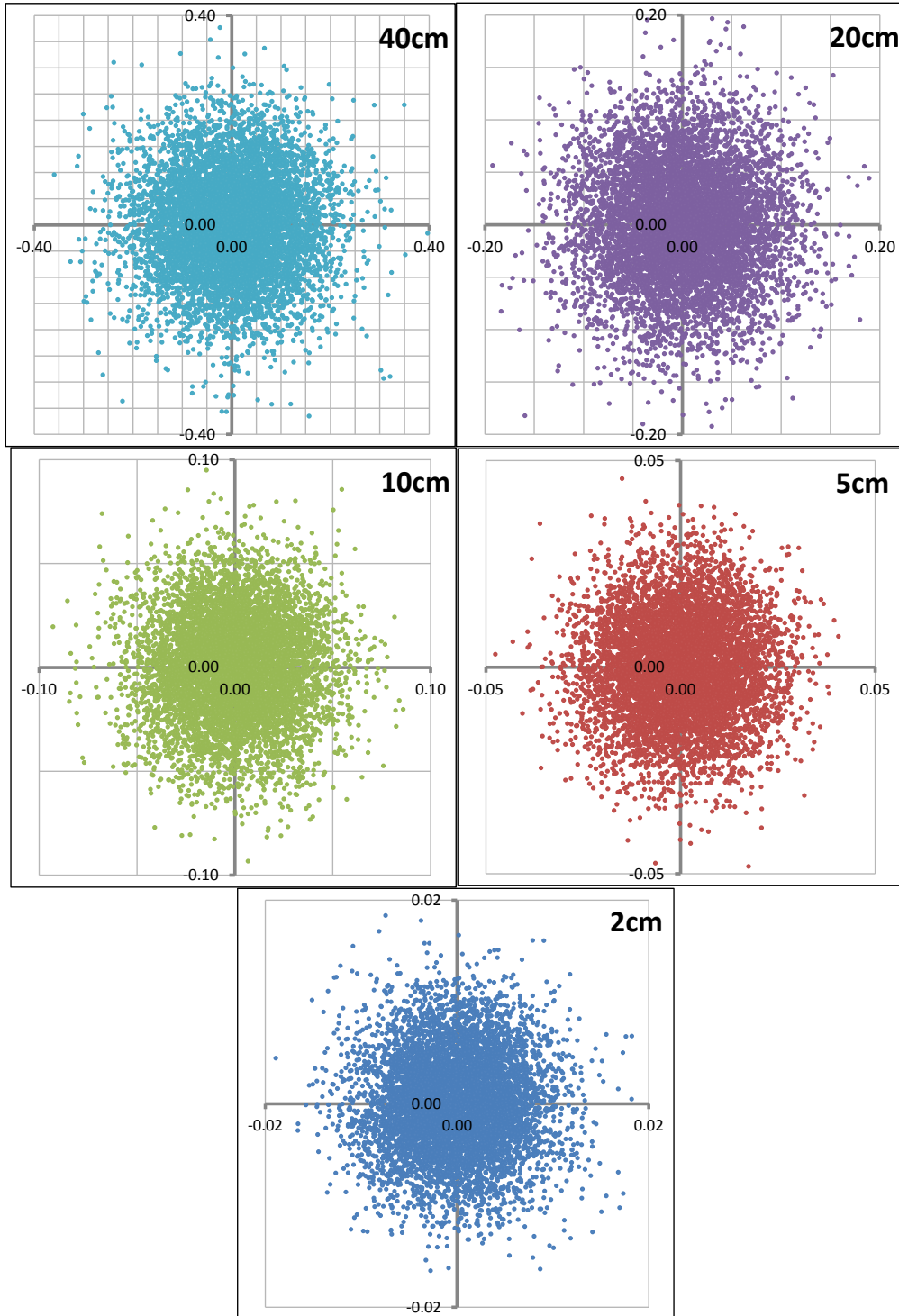
```

clear all
close all
clc
load TRACKS_18.mat;

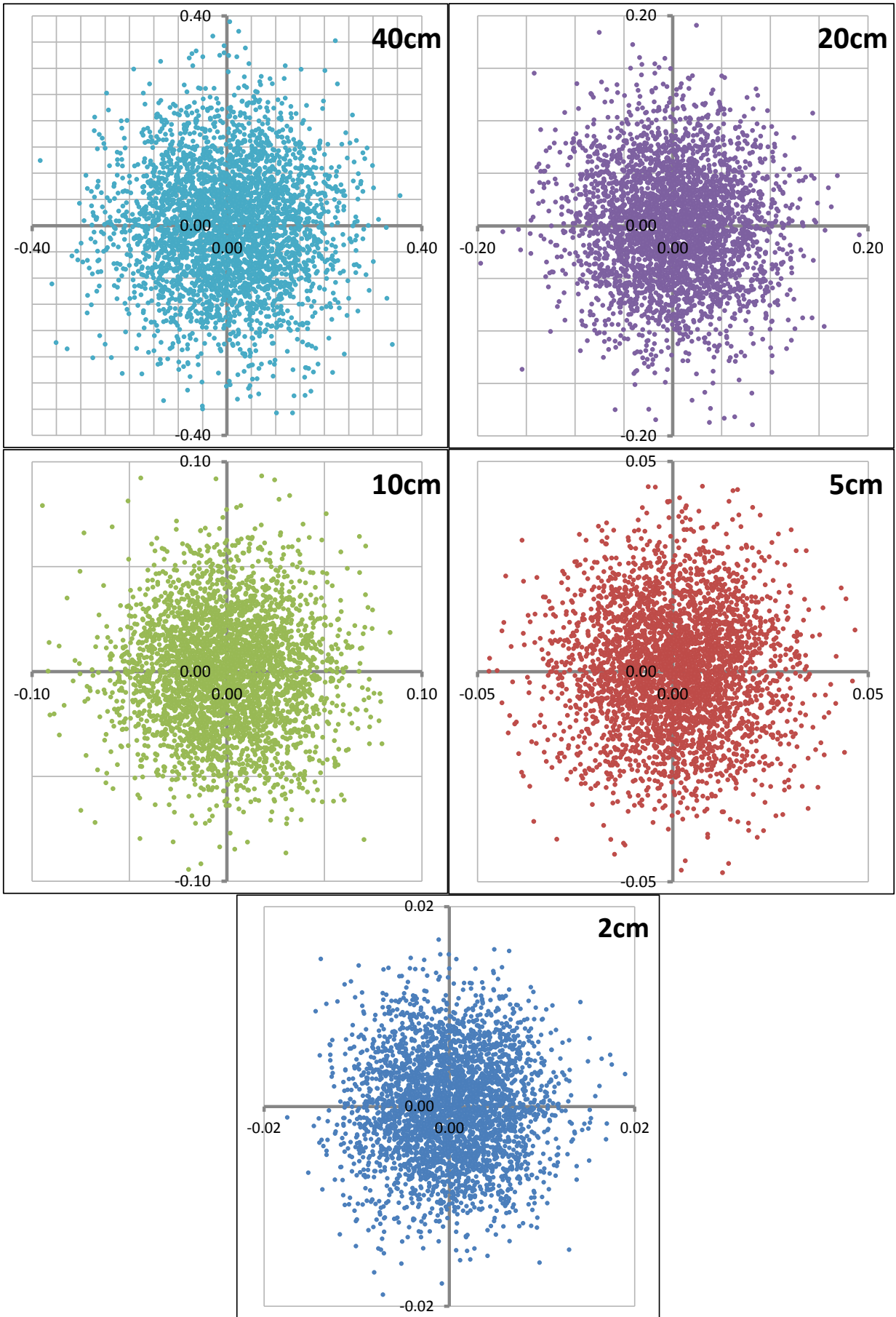
for a=1:length(TRACKLIST_SURV)
    s=char(TRACKLIST_SURV(a));
    for b=3:length(GEOMETRY) %curve research
        if strcmp(GEOMETRY(b),'Curva')==1
            s1=['X=' s '.PROJ_SEGM.El_' num2str(b) '.Est;']; eval(s1);
            s2=['Y=' s '.PROJ_SEGM.El_' num2str(b) '.Nord;']; eval(s2);
            if length(X)>3
                X=X-386887.4198;
                Y=Y-4979559.0127;
                [xc,yc,Rad]=landau(X,Y);
                Cen=[xc+386887.4198,yc+4979559.0127];
                s3=[s '.PROJ_SEGM.El_' num2str(b) '.LD.Cen=Cen;']; eval(s3);
            %center coordinate storage
                s4=[s '.PROJ_SEGM.El_' num2str(b) '.LD.Rad=Rad;']; eval(s4);
            %radius storage
            else
                s5=[s '.PROJ_SEGM.El_' num2str(b) '.LD.Error='Not enough
points';']; eval(s5); %error message: too few points
            end
            end
            clear X Y Cen Rad
        end
    end
end

```

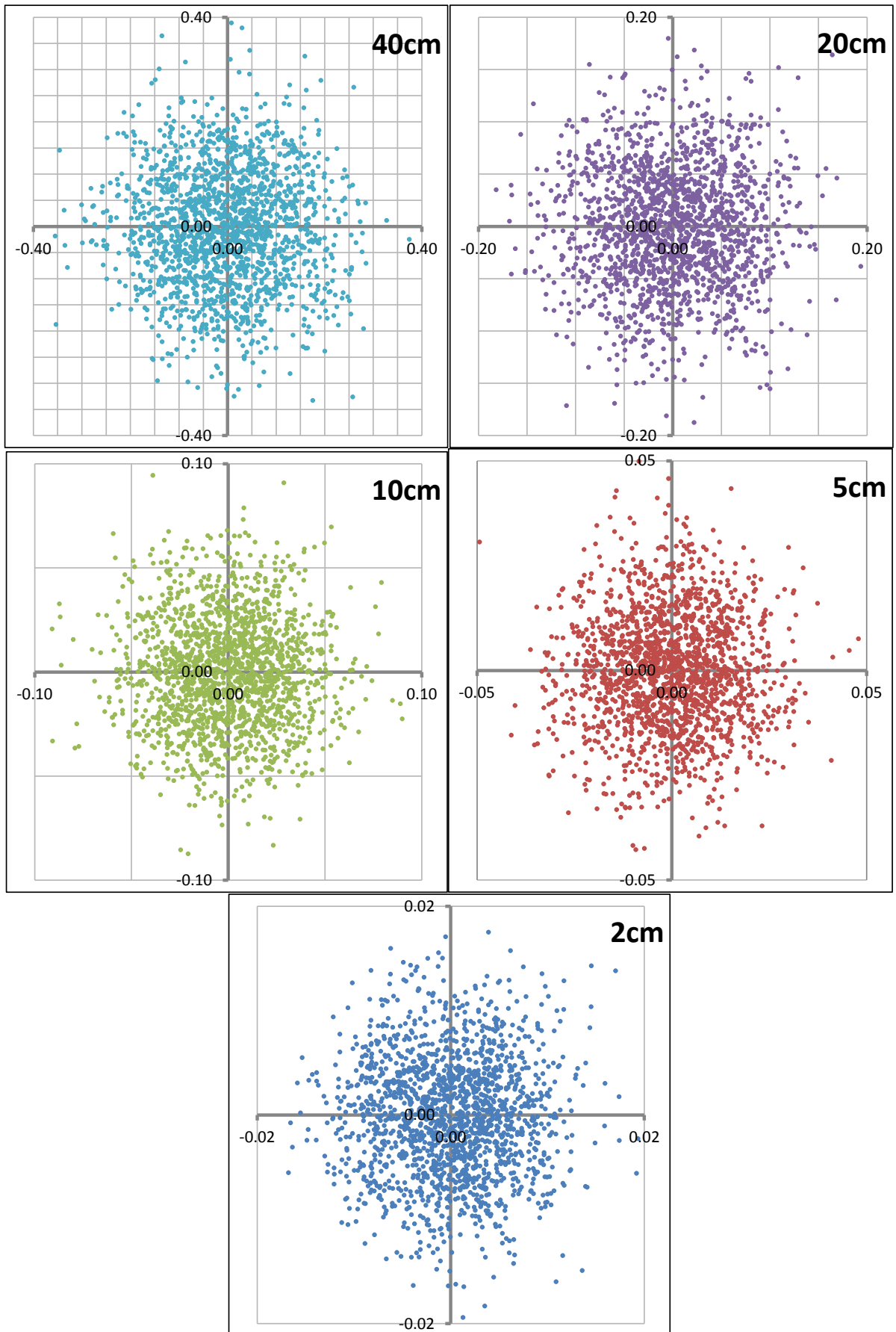
Random errors detailed view



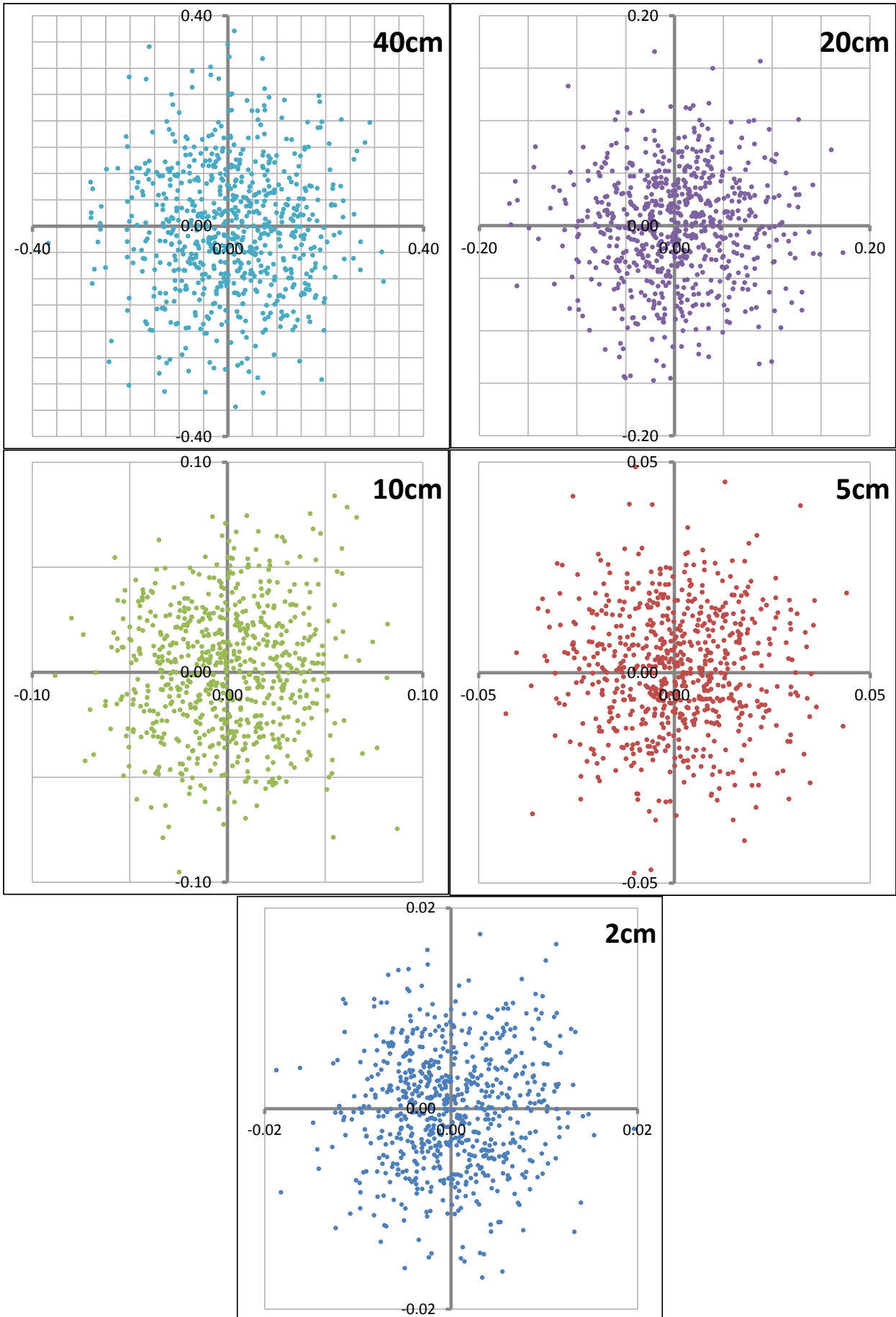
Random error distribution - 0.5m



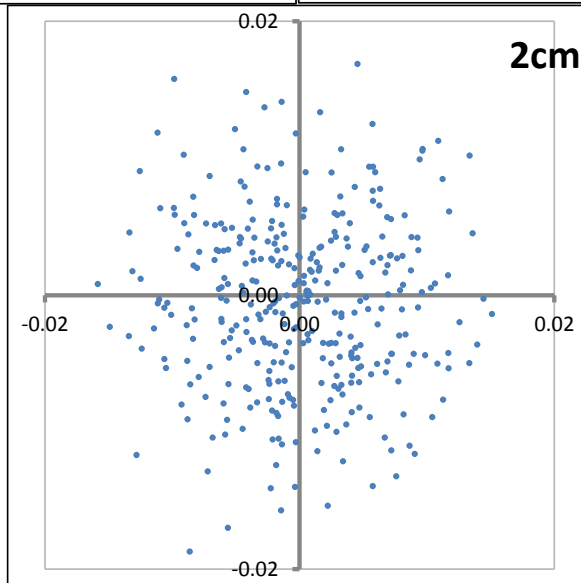
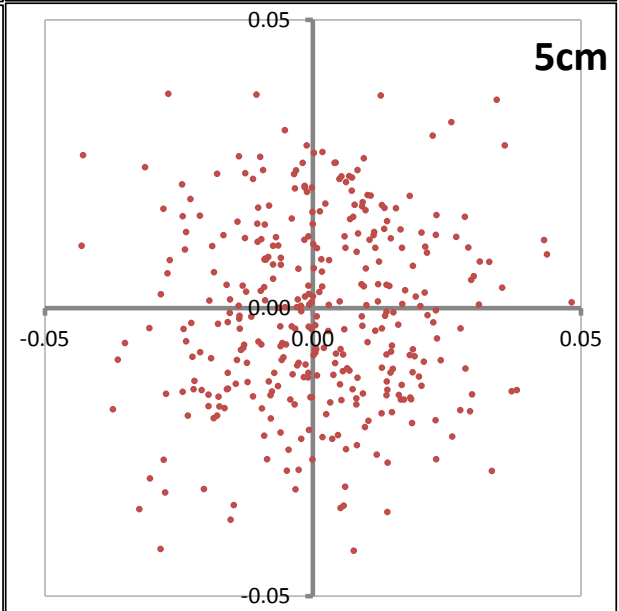
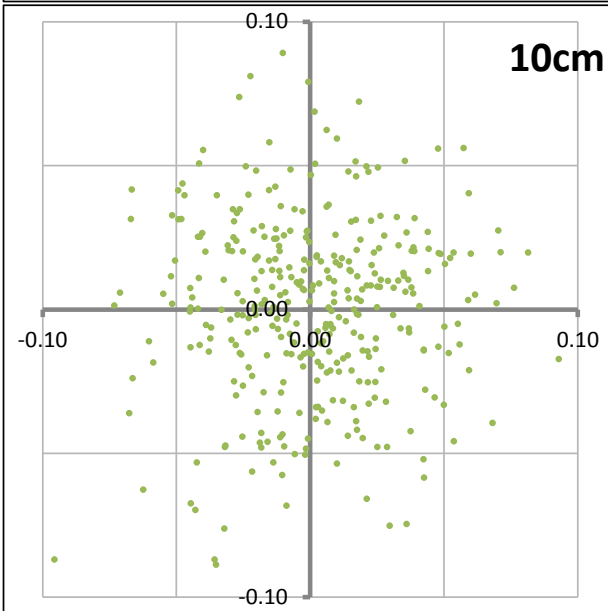
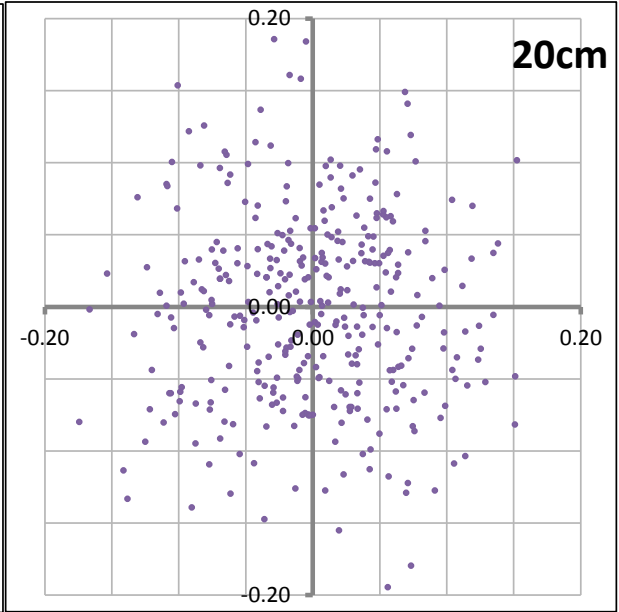
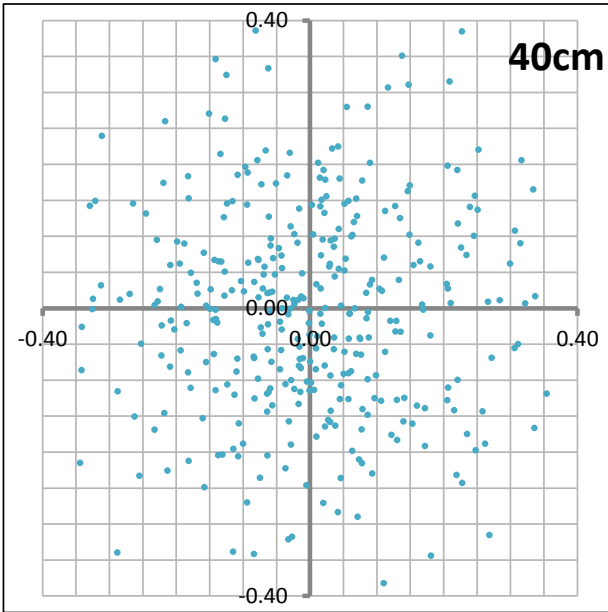
Random error distribution - 1m



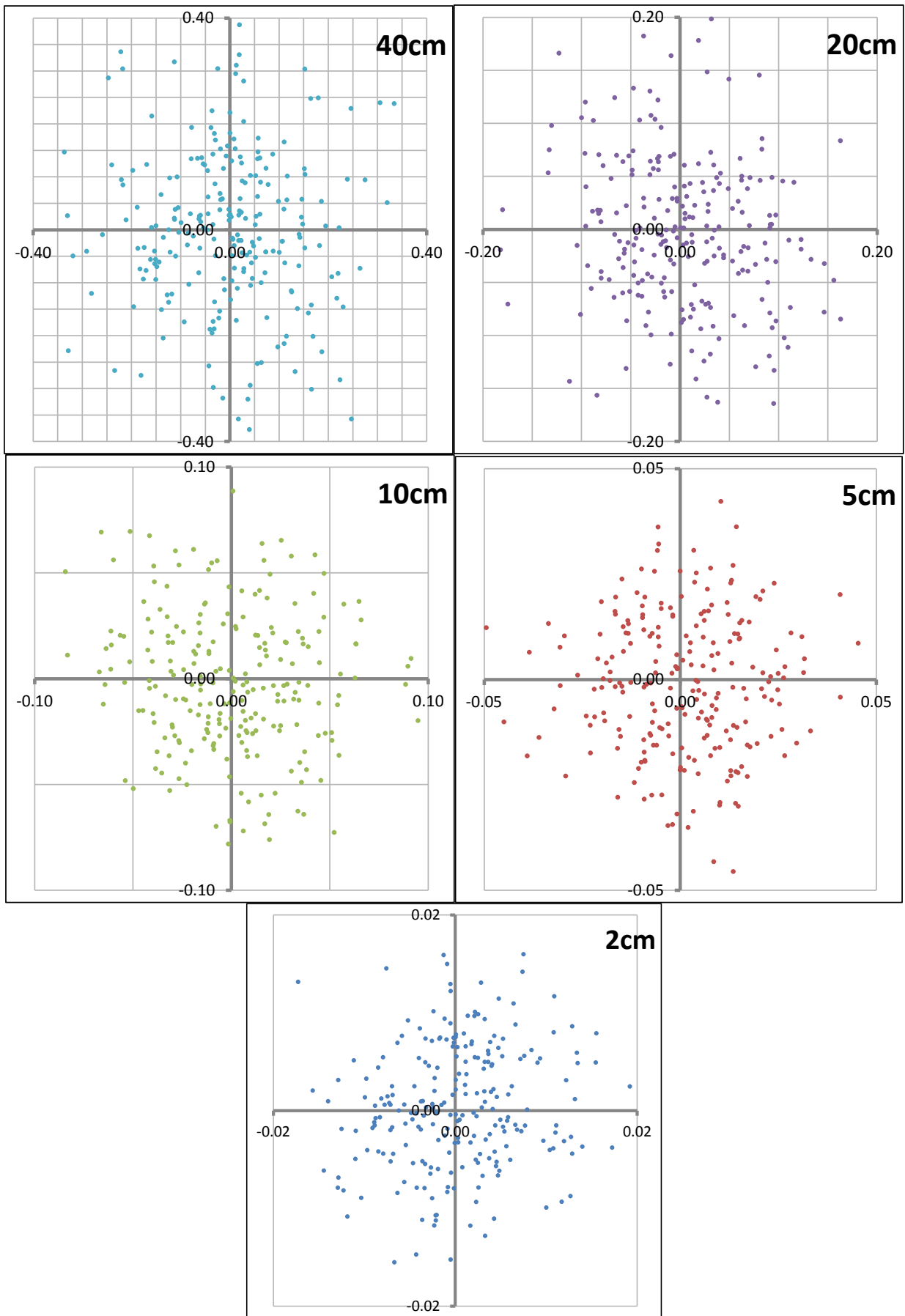
Random error distribution - 2m



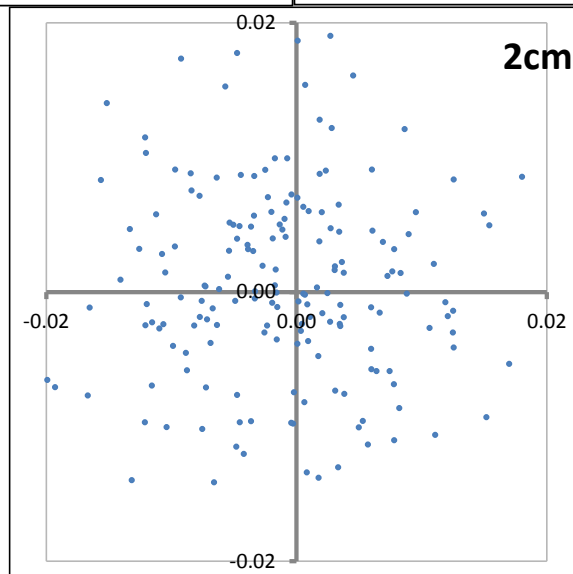
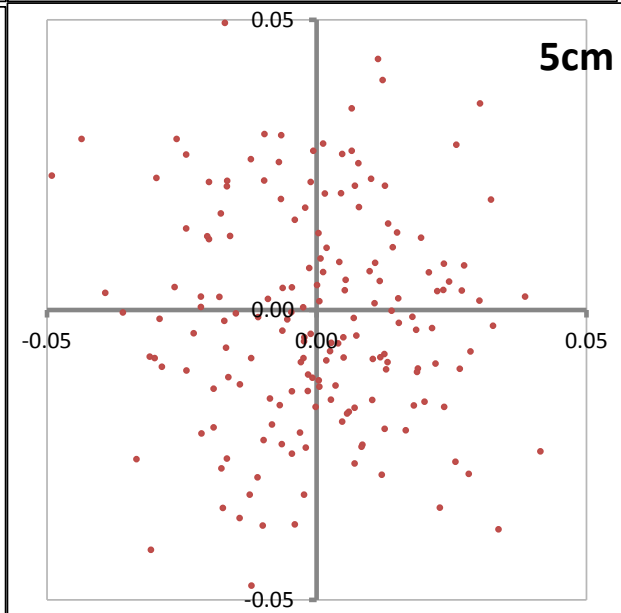
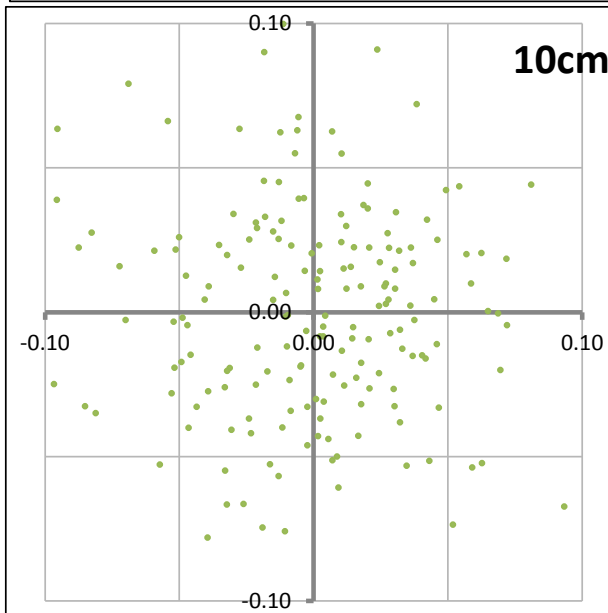
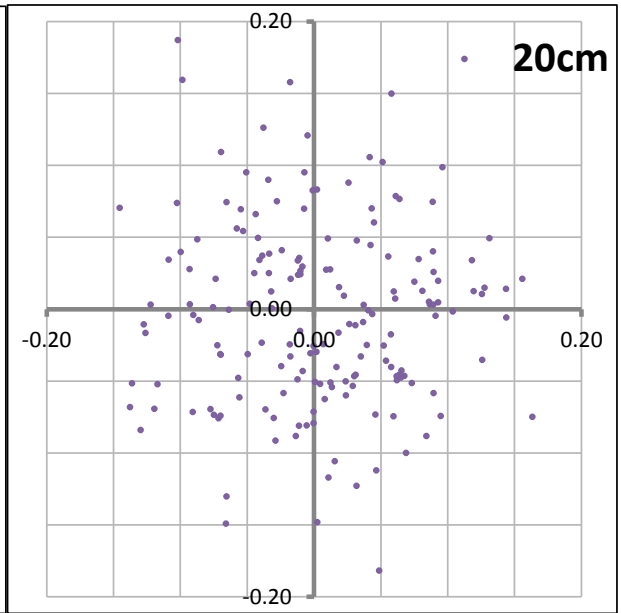
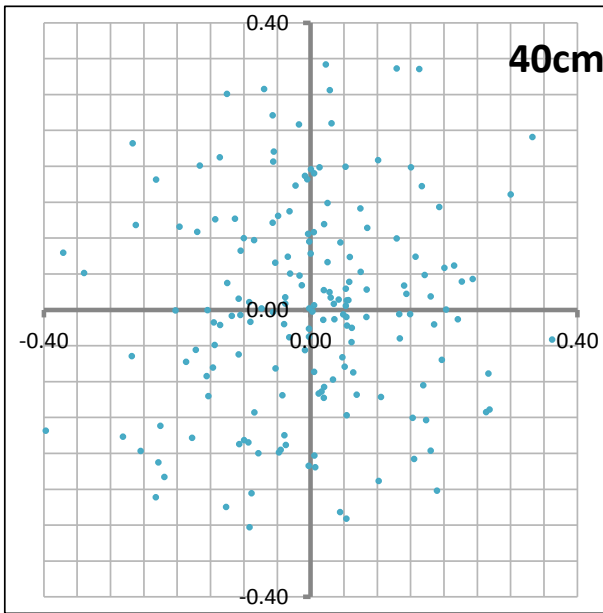
Random error distribution - 5m



Random error distribution - 10m

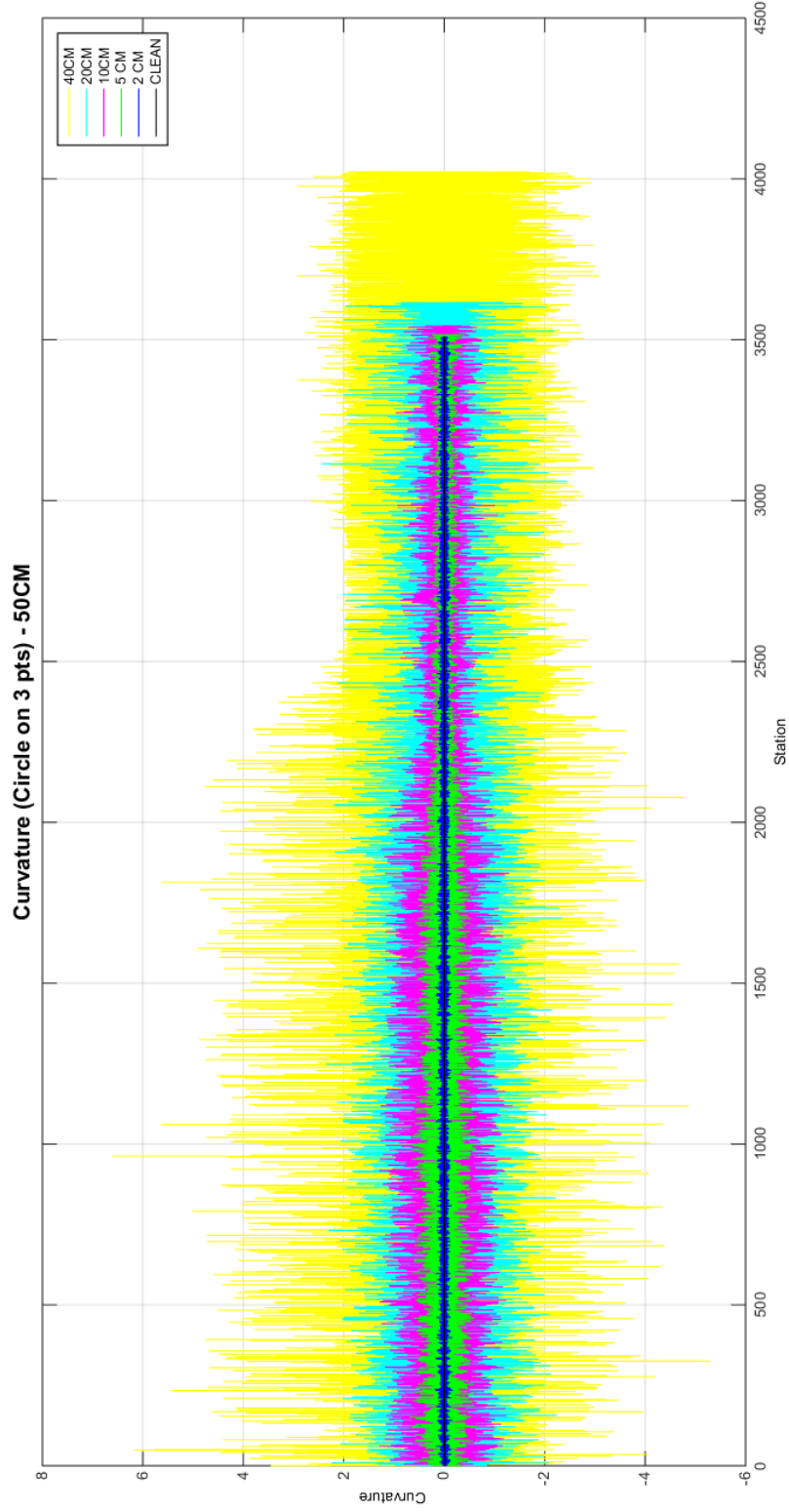


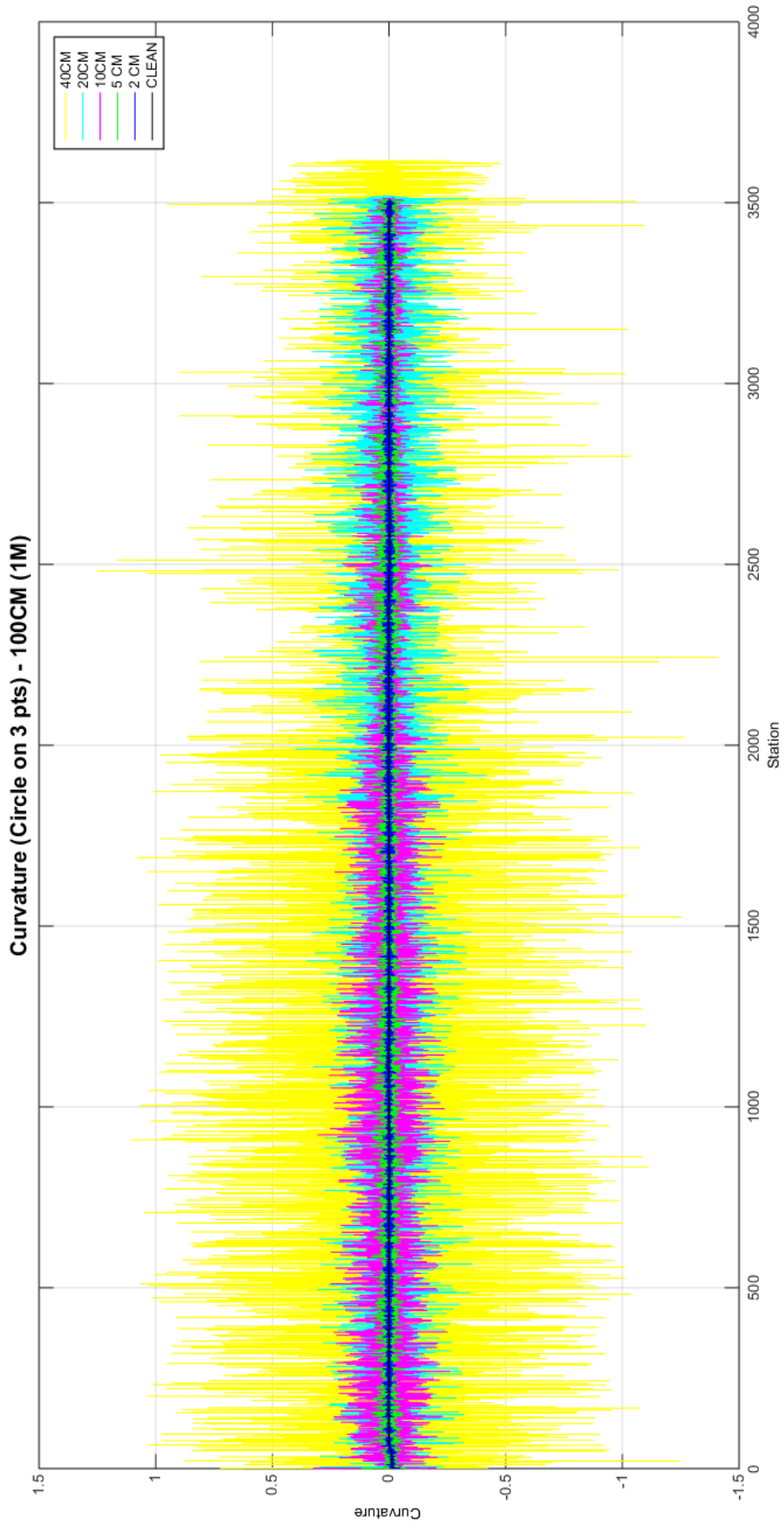
Random error distribution - 15m

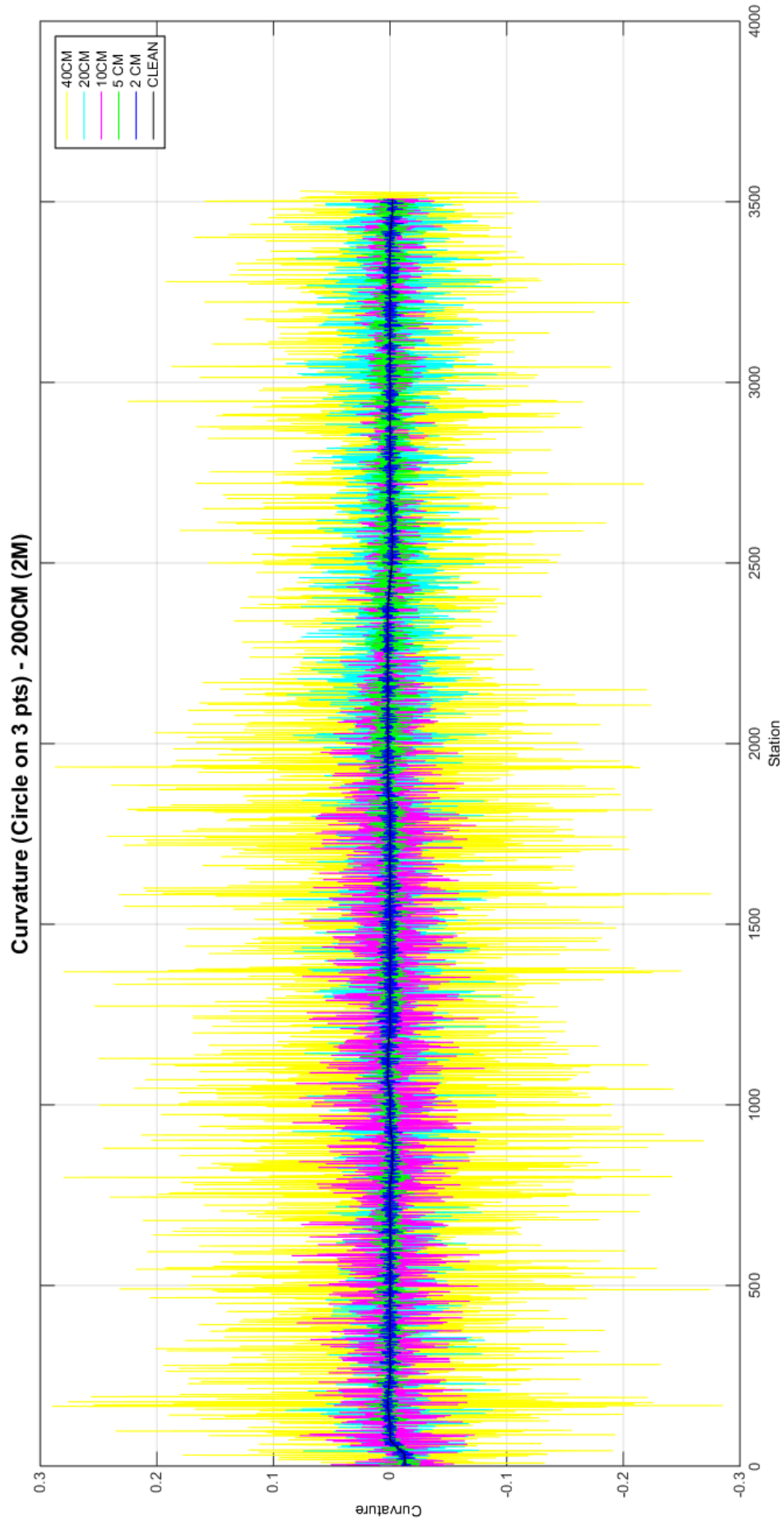


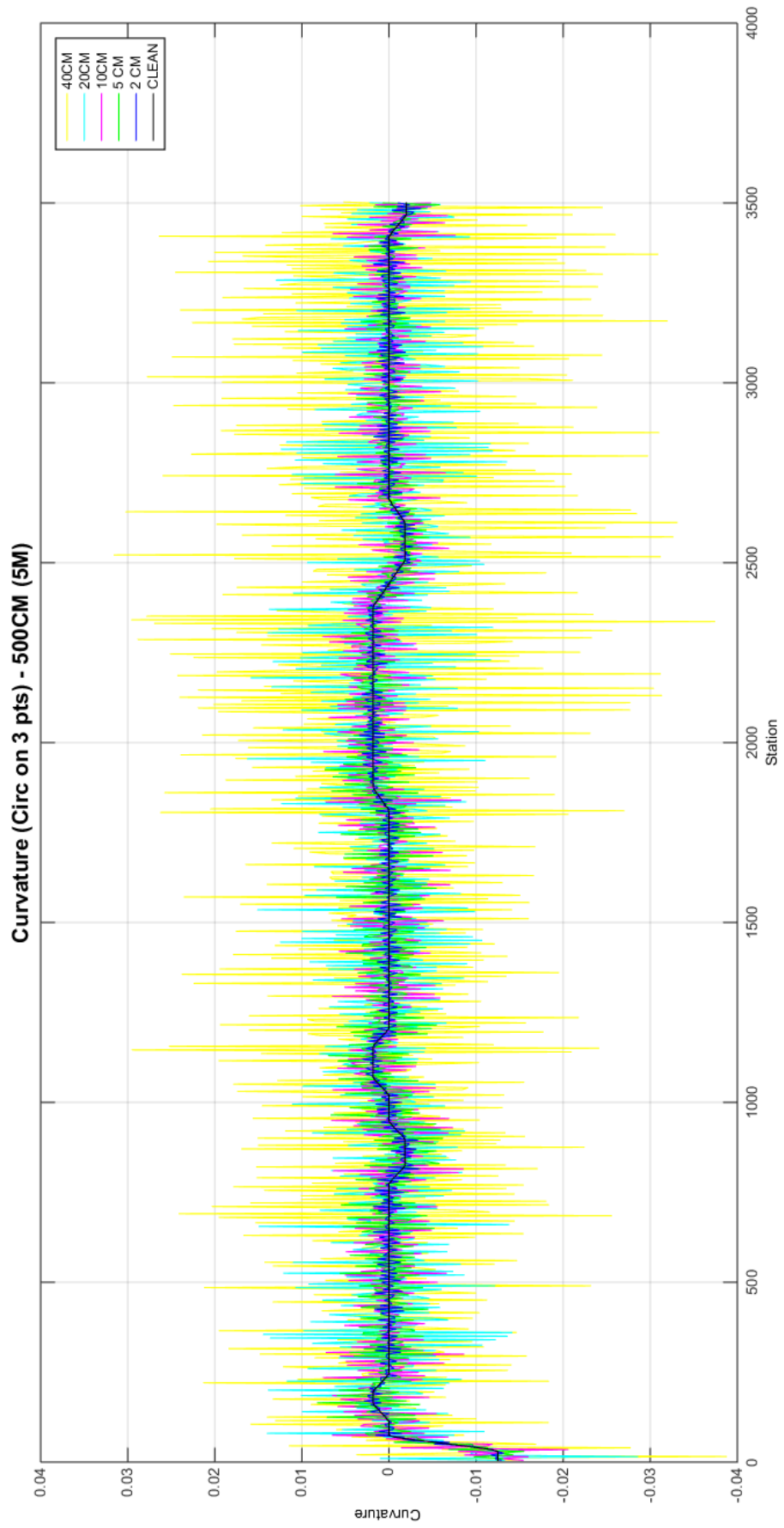
Random error distribution - 20m

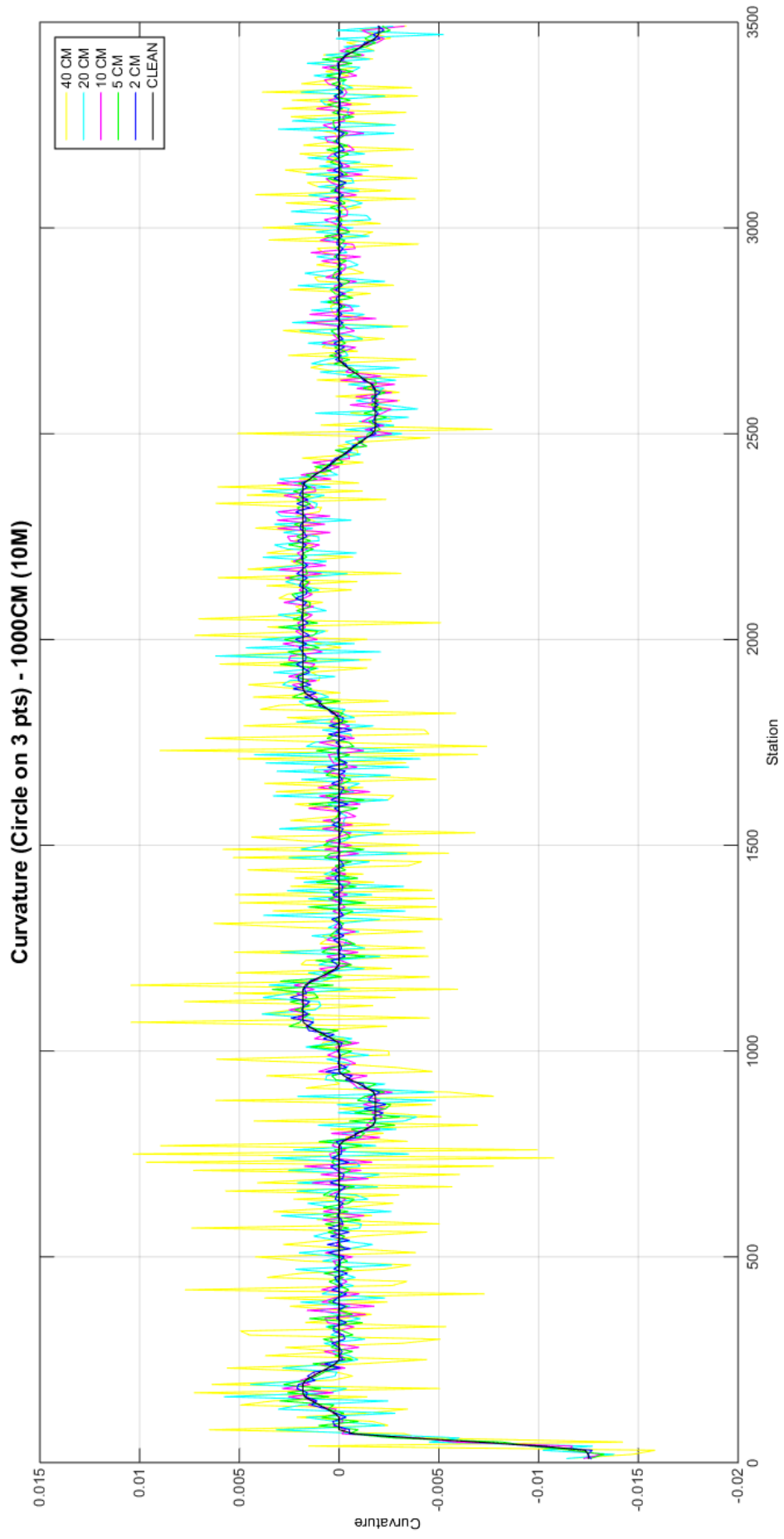
Curvature profiles: axis methodology (design based tracks)

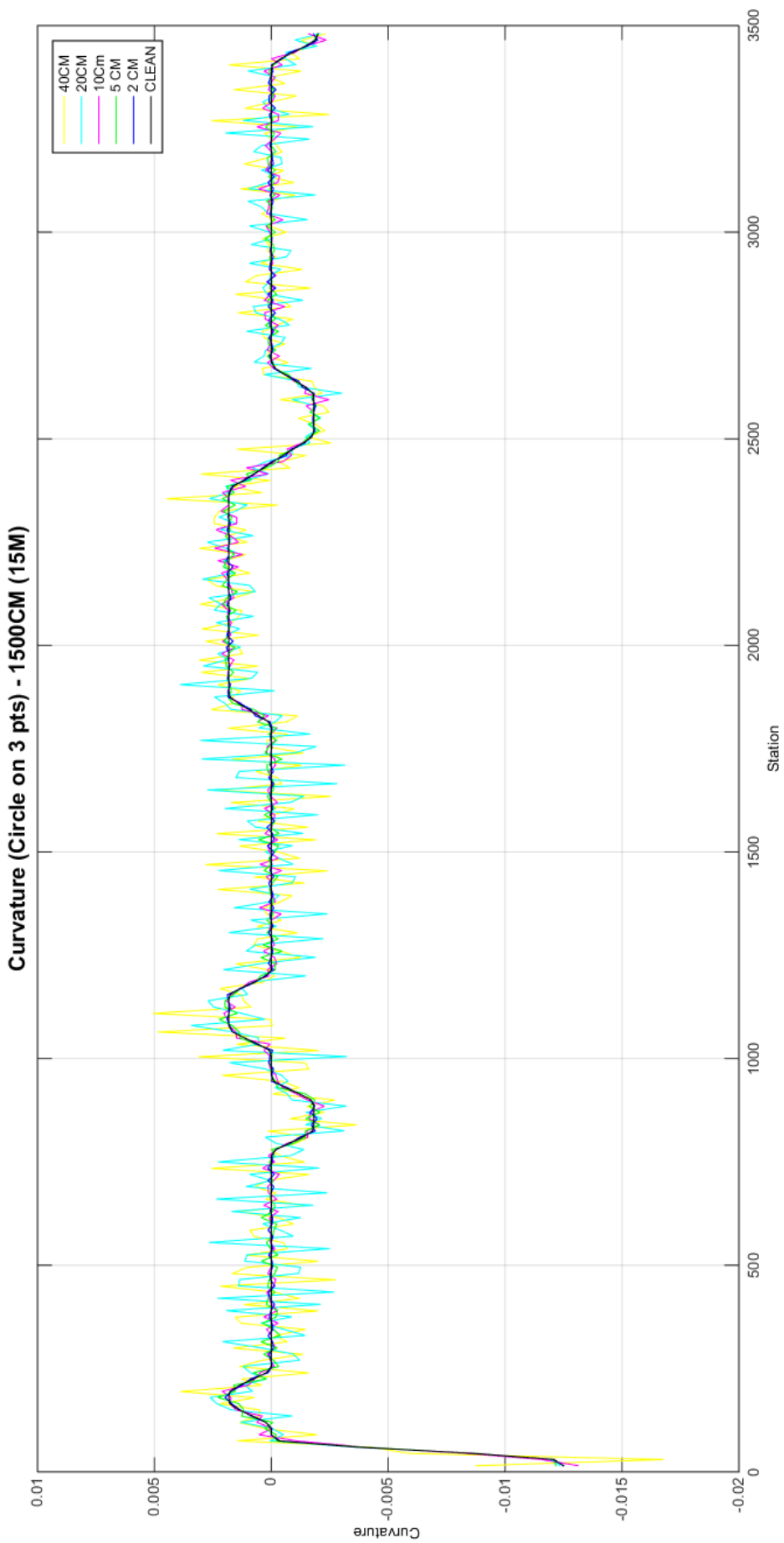


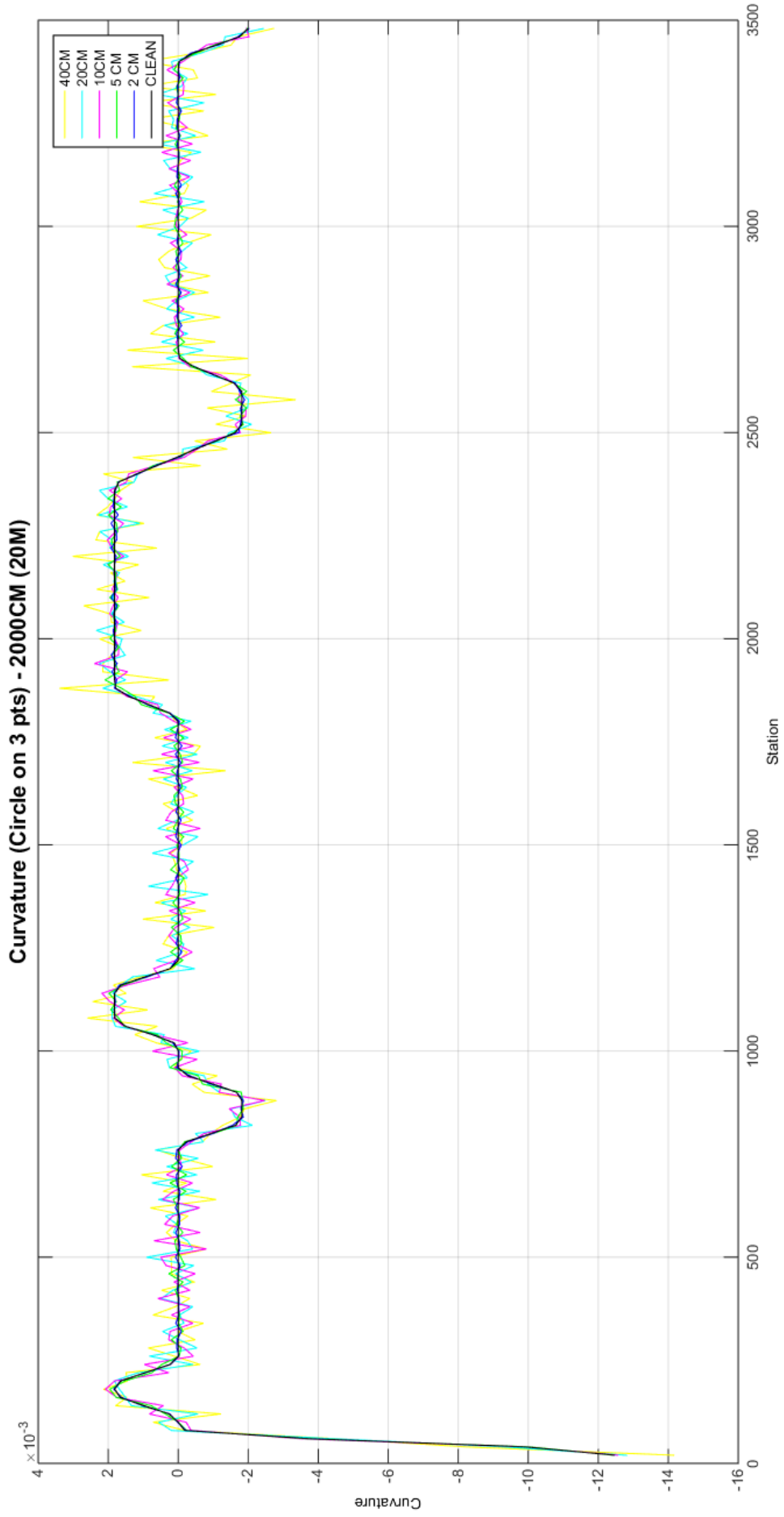




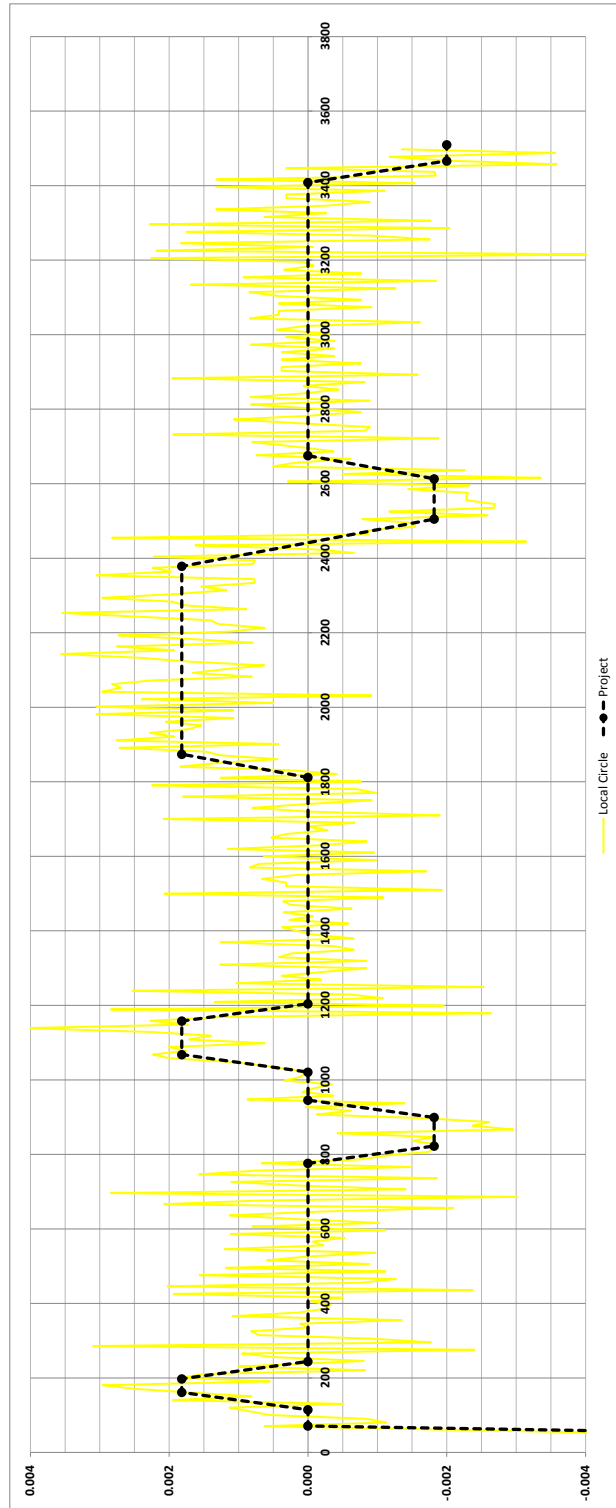




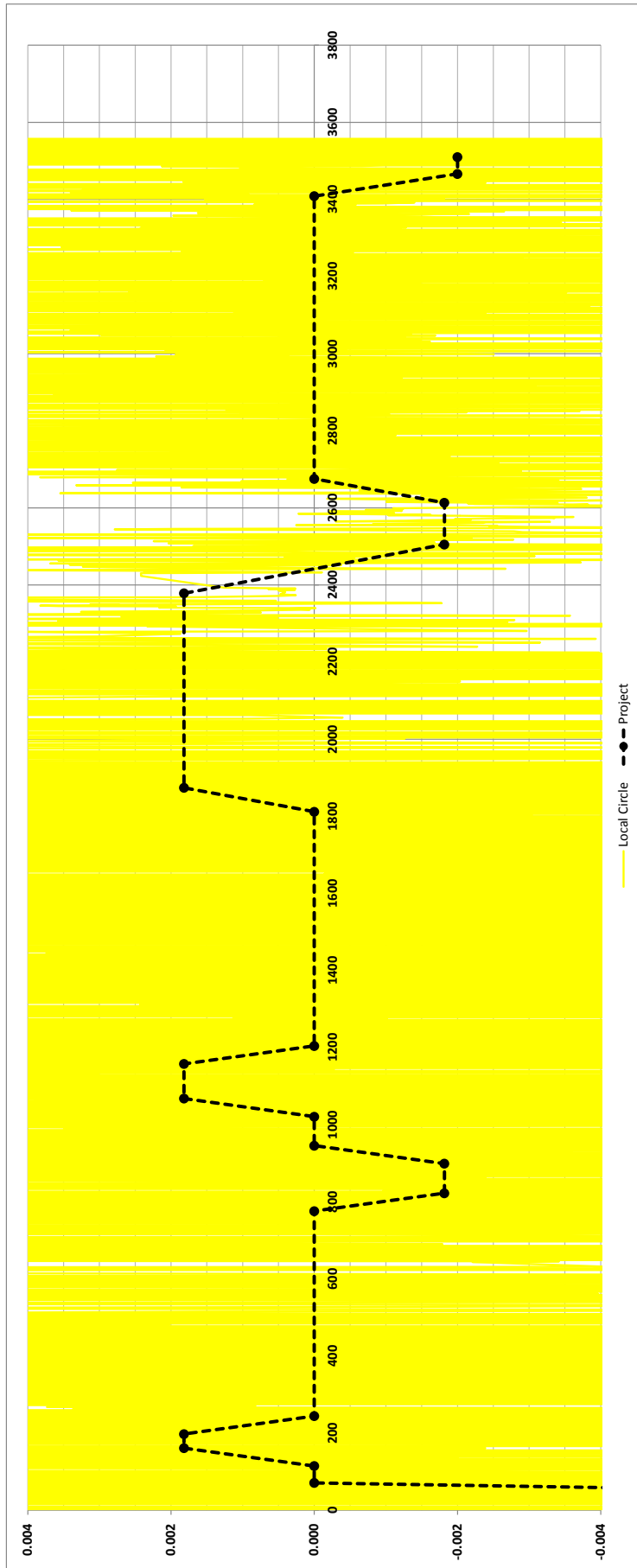




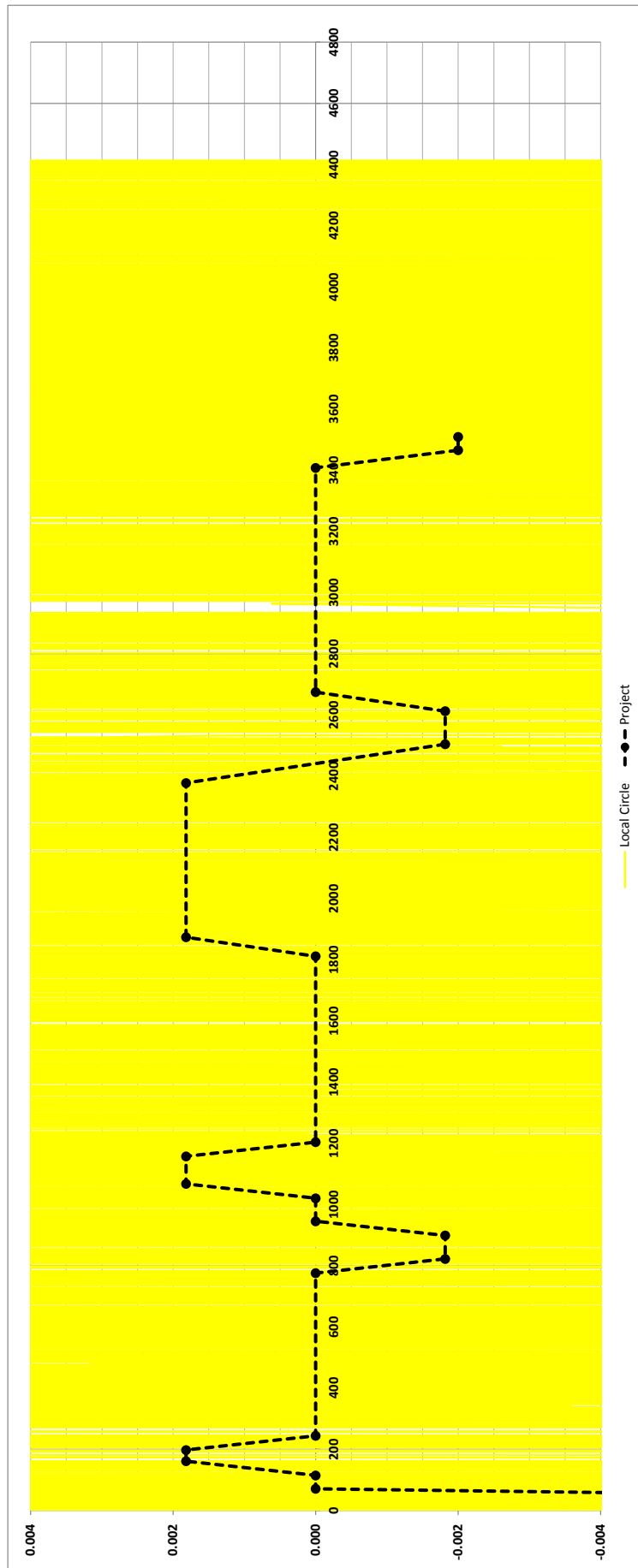
Curvature profiles: axis methodology (surveyed tracks)



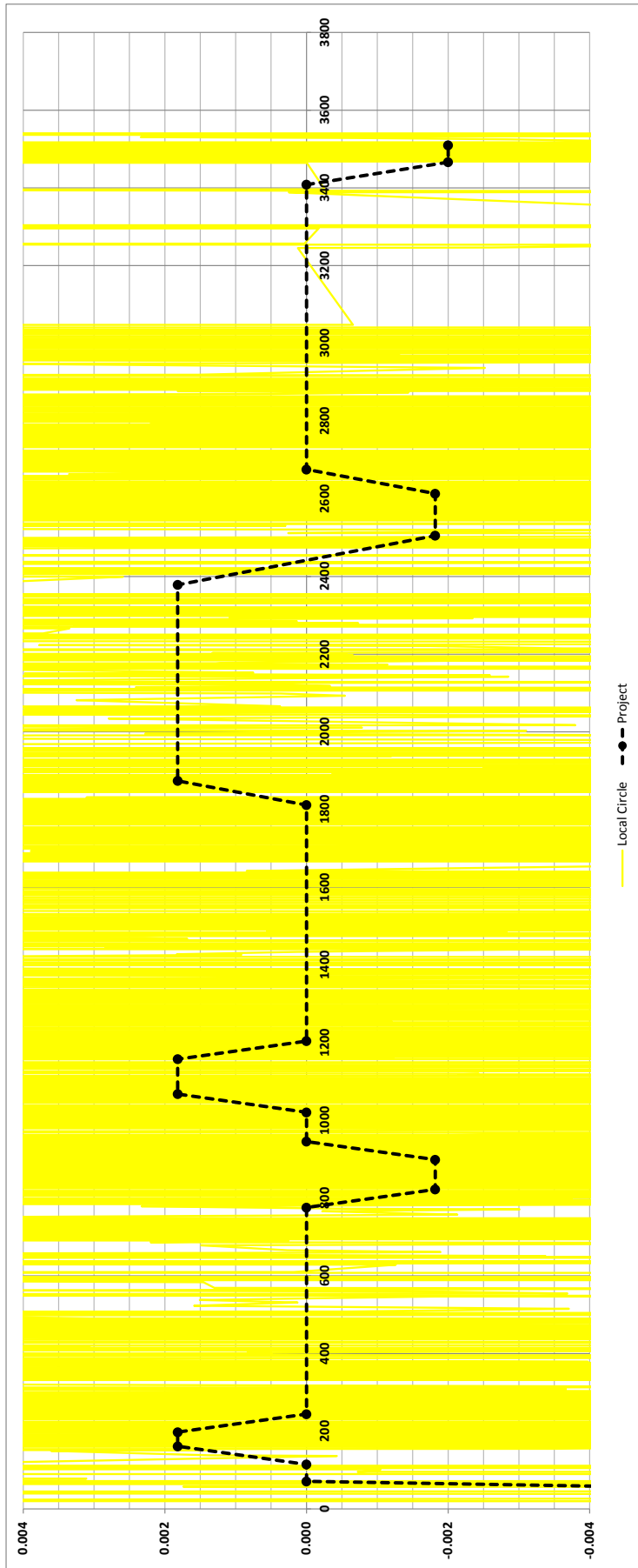
GIS Centerline



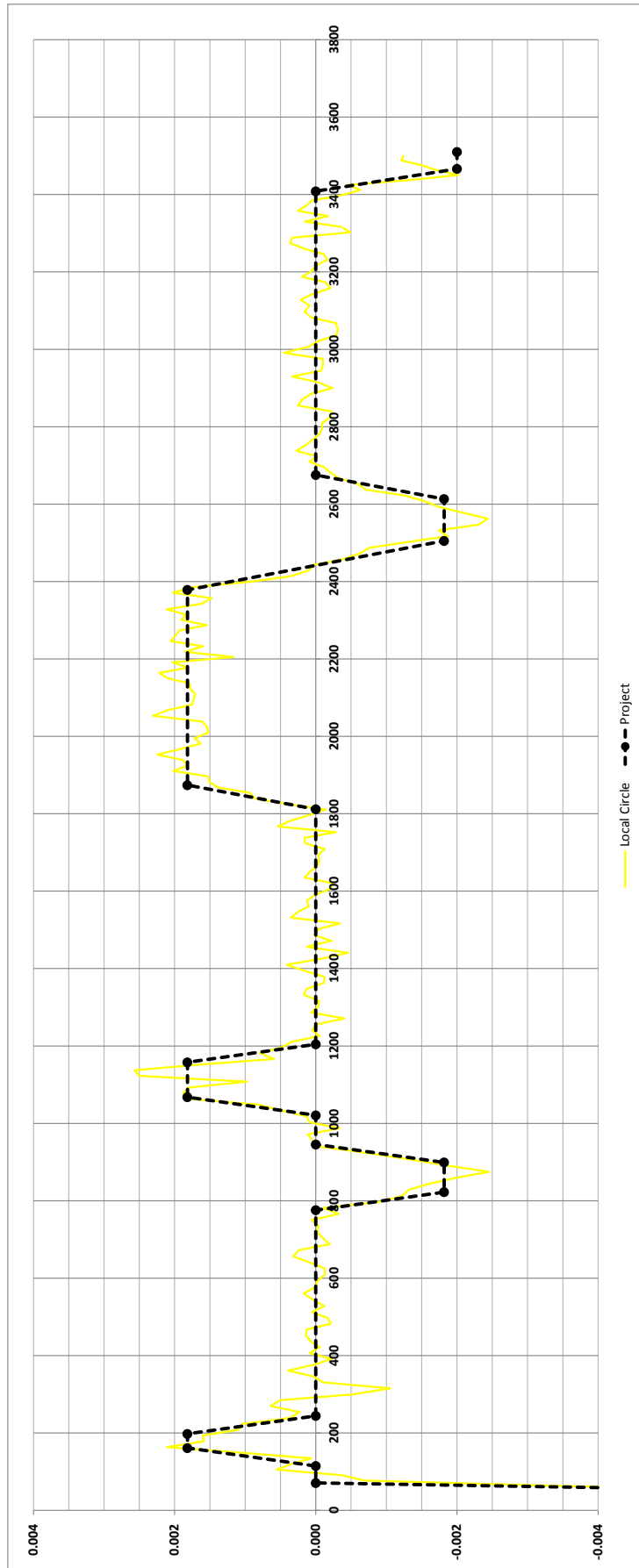
IMU Trajectory



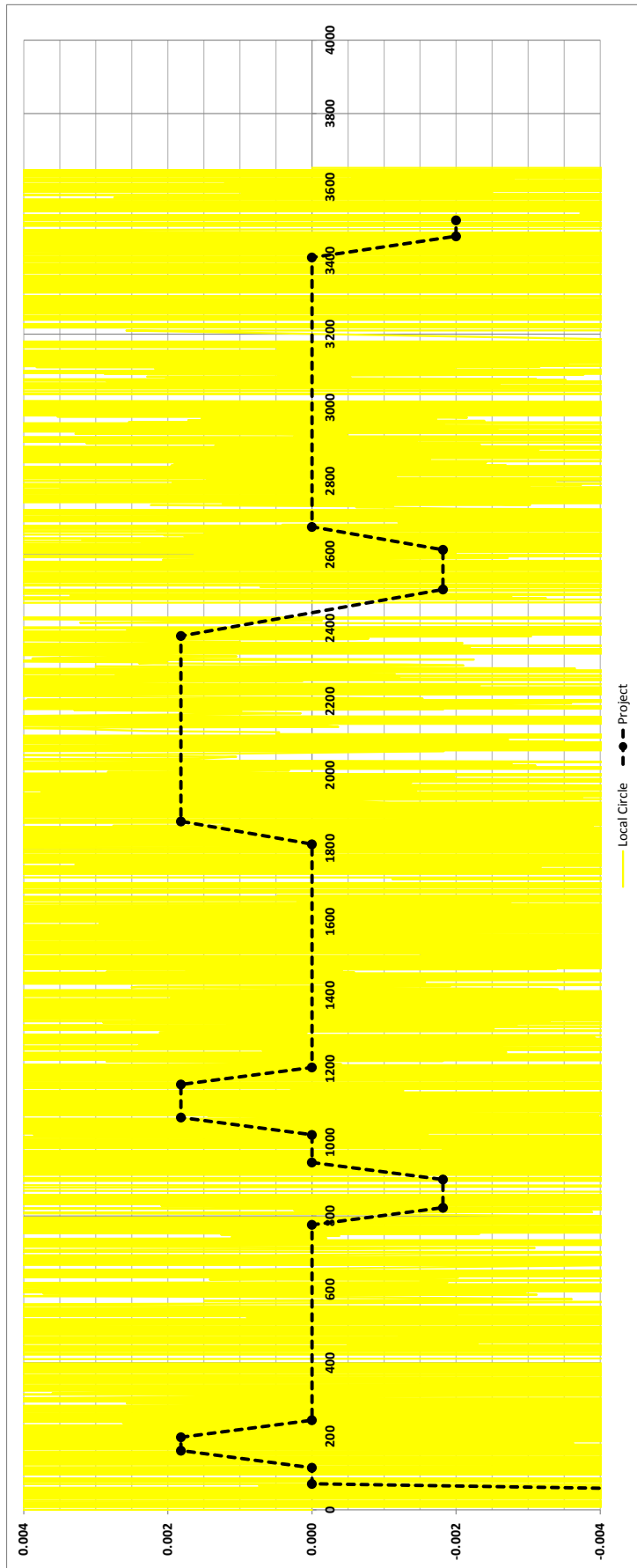
IMU Cigline



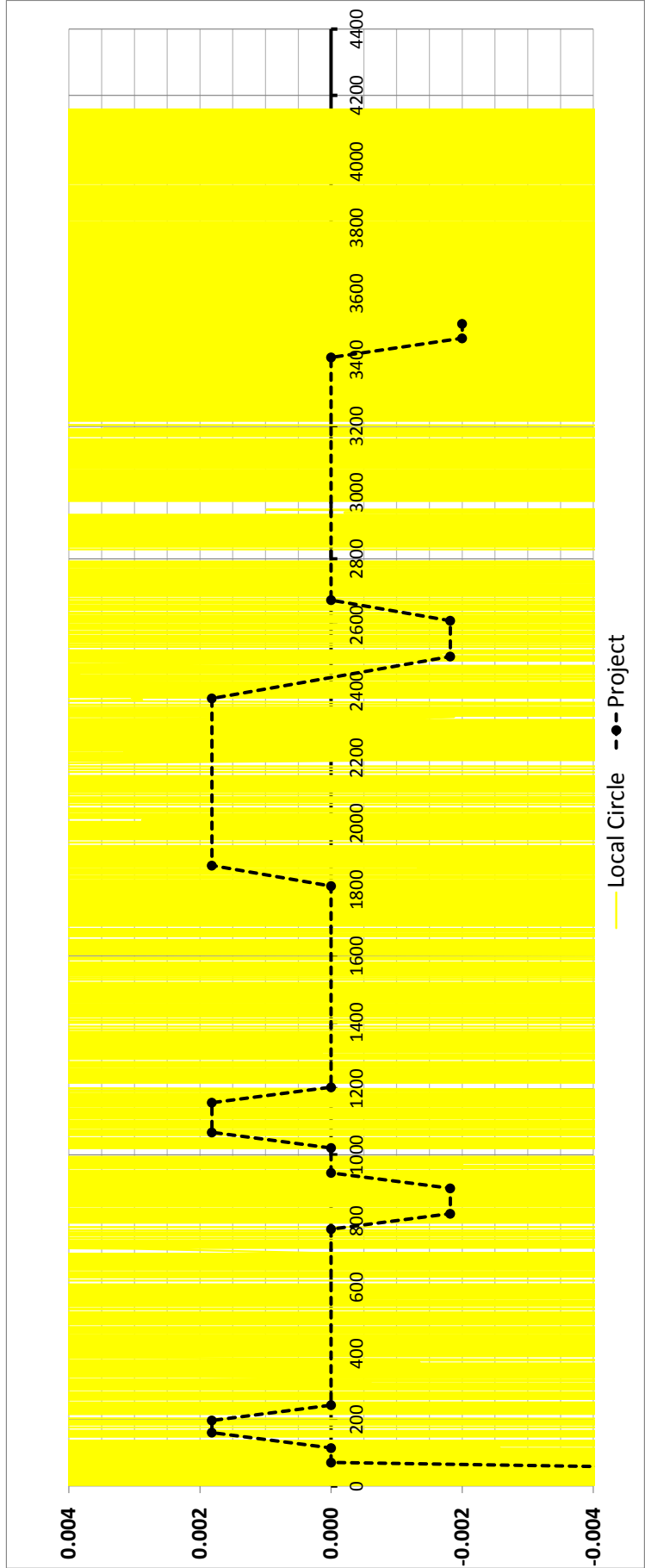
IMU Centerline



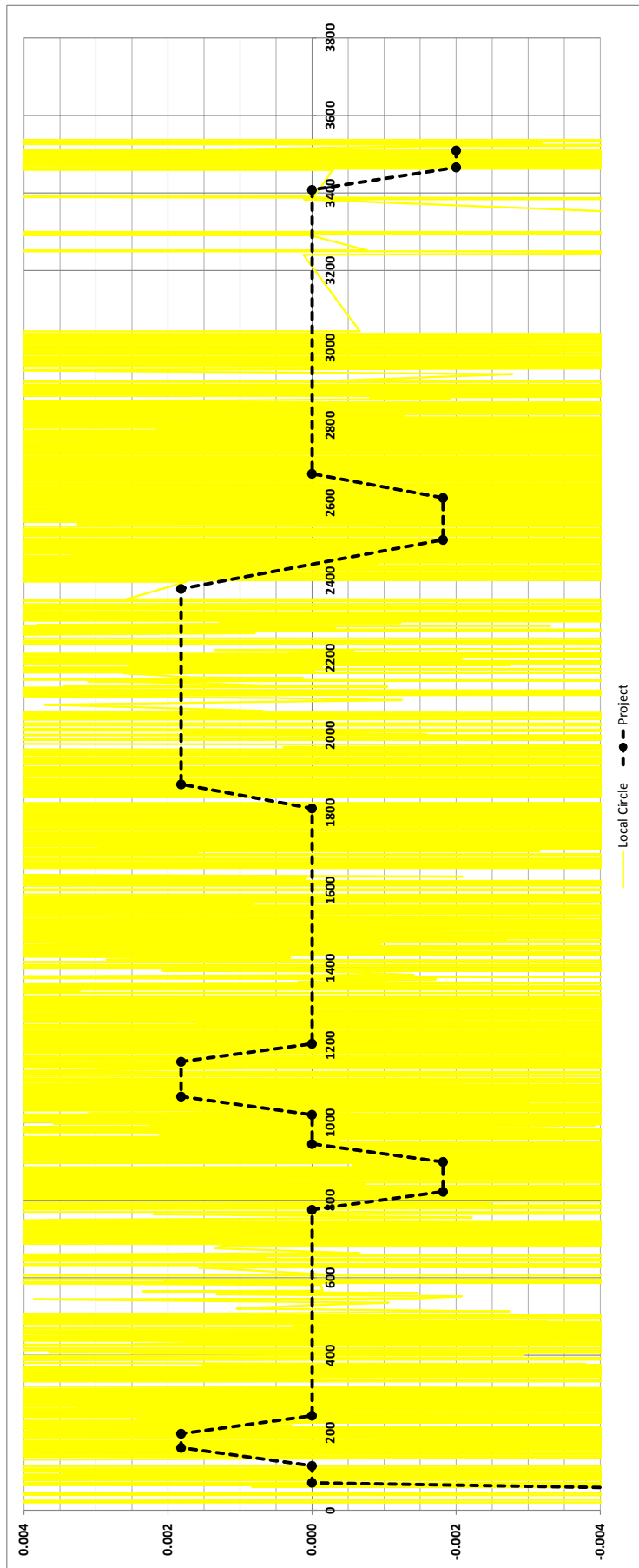
Gps Trajectory



GPS LatLine



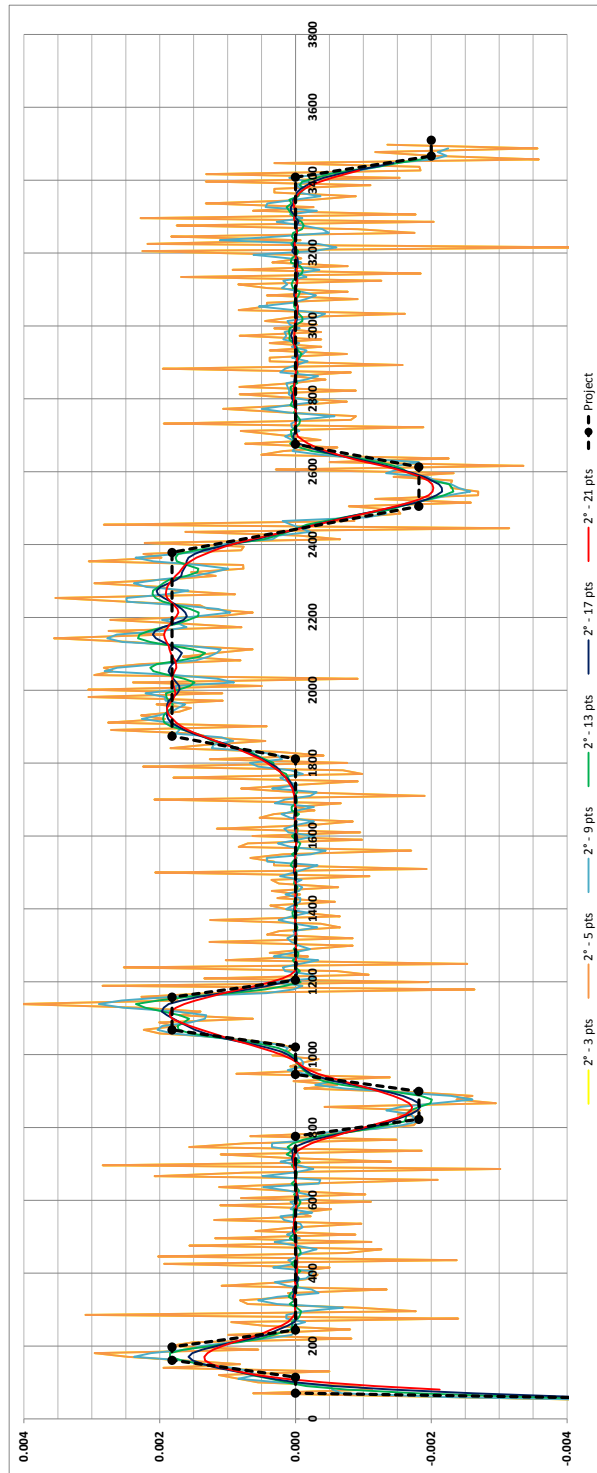
GPS CigLine



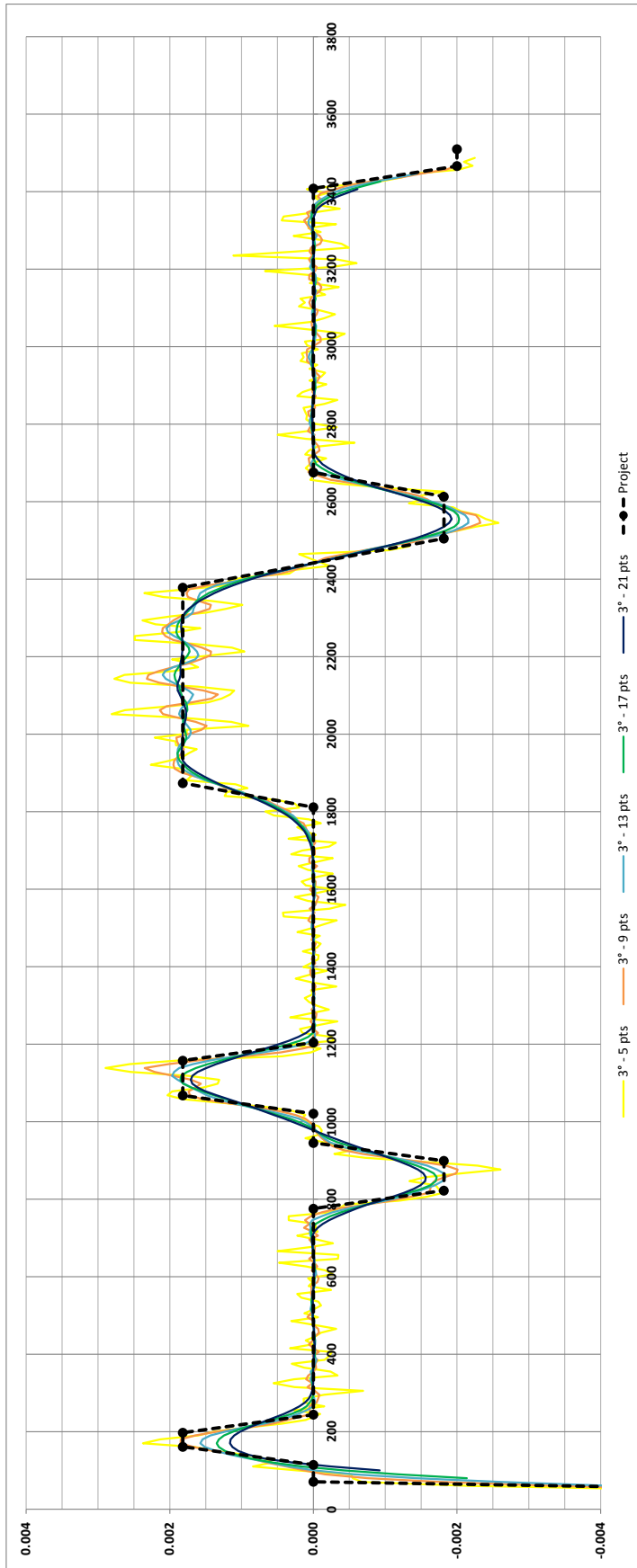
GPS Centerline



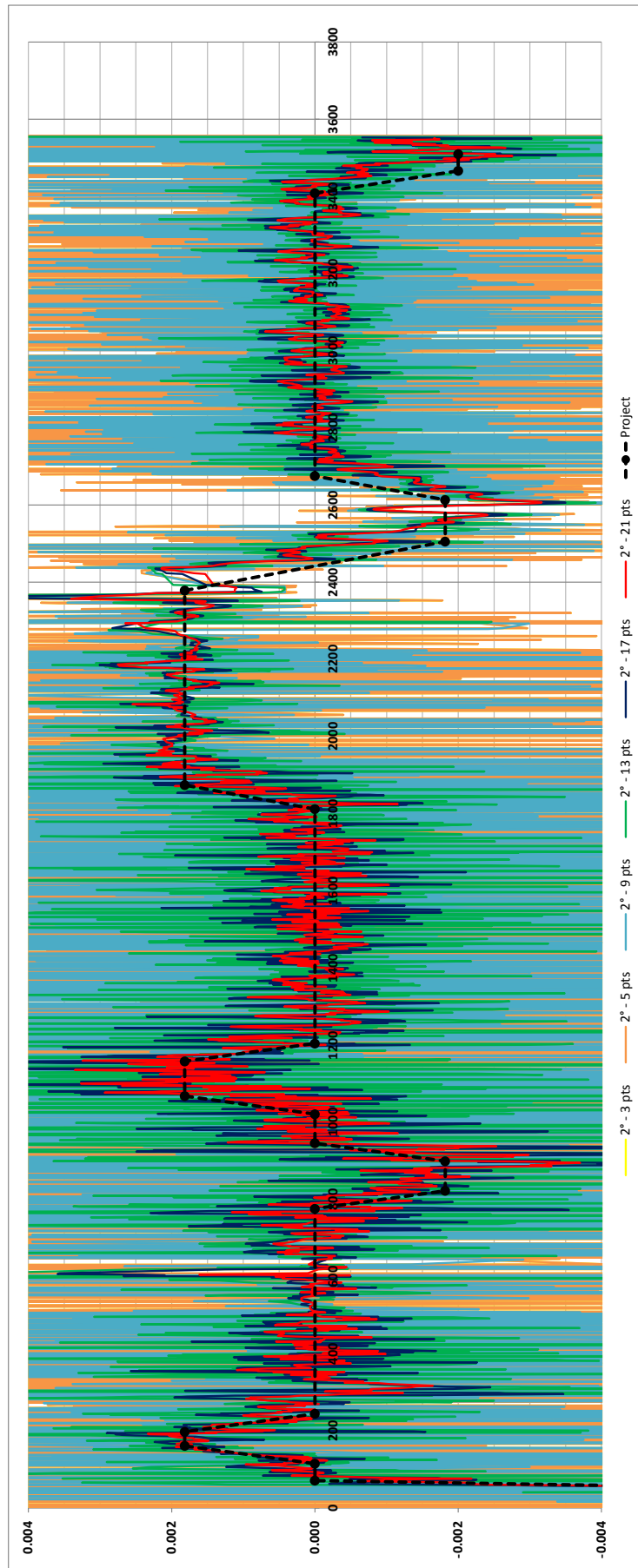
Curvature profiles: polynomial fitting (surveyed tracks)



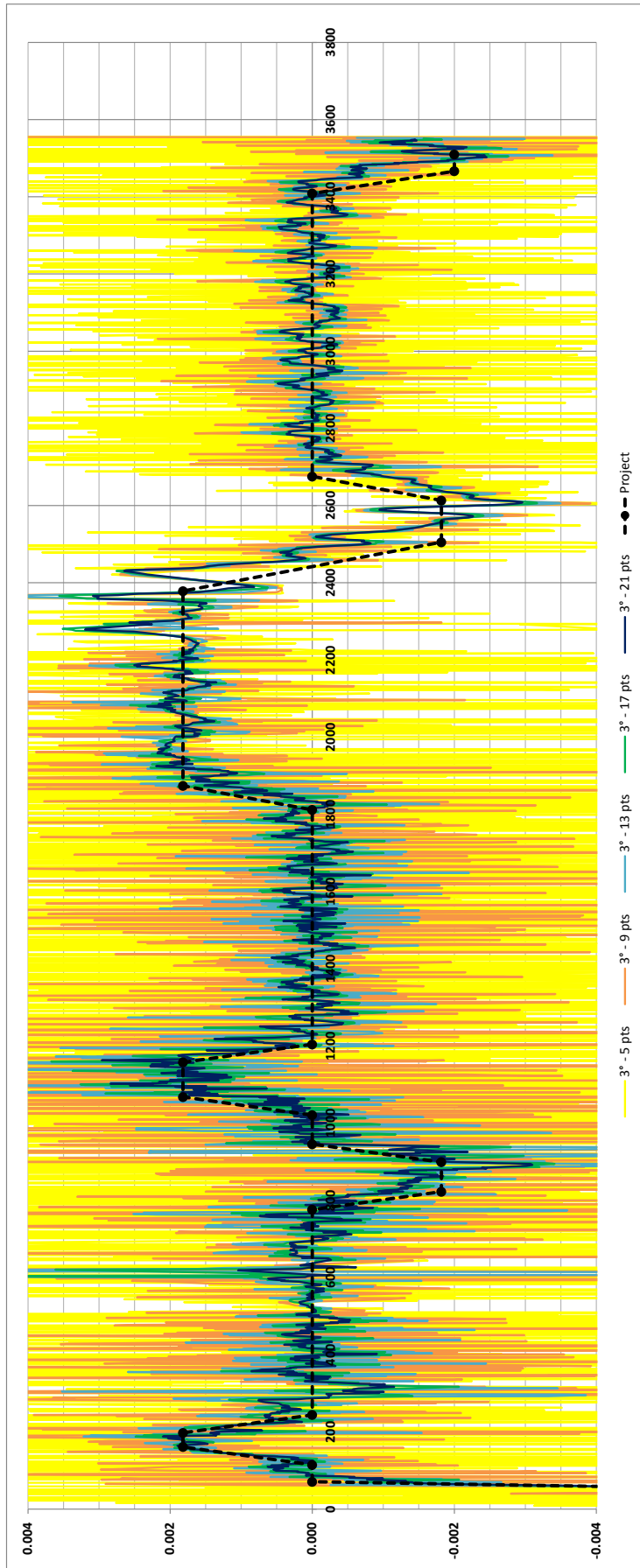
GIS Centerline - 2nd degree



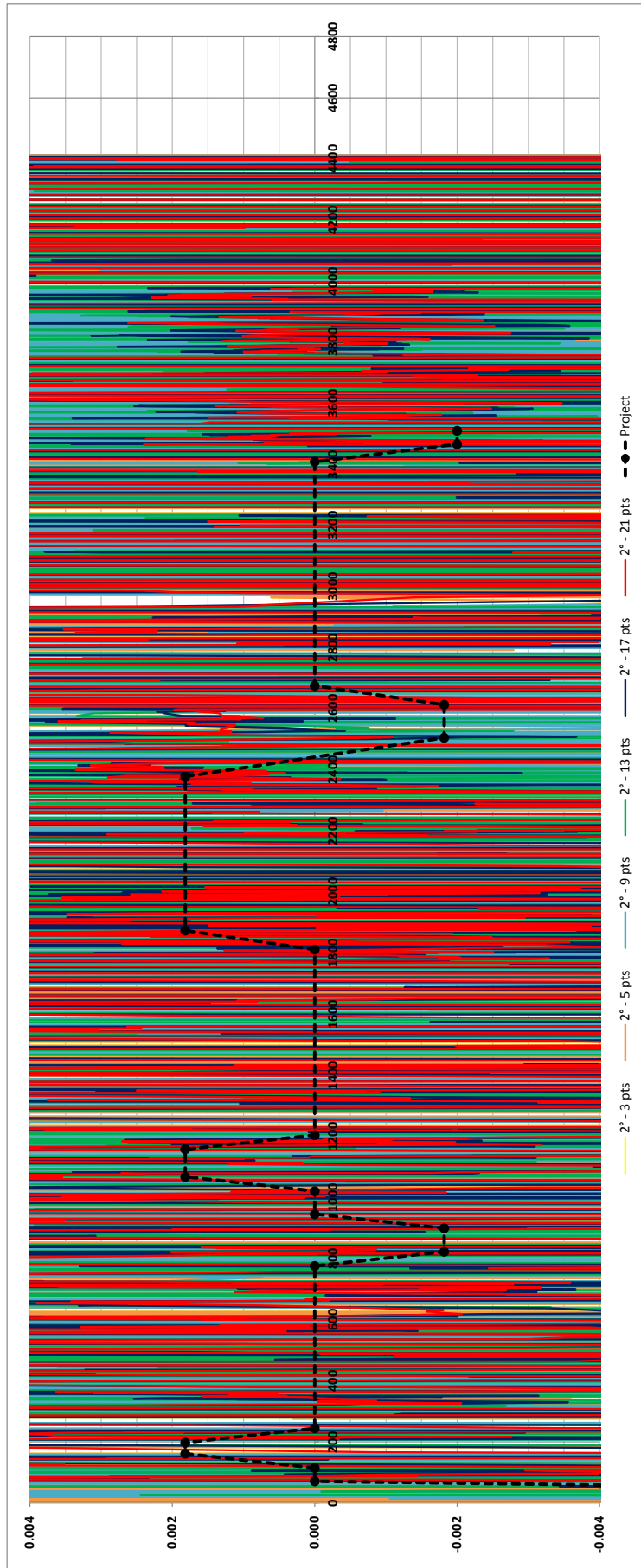
GIS Centerline - 3rd degree



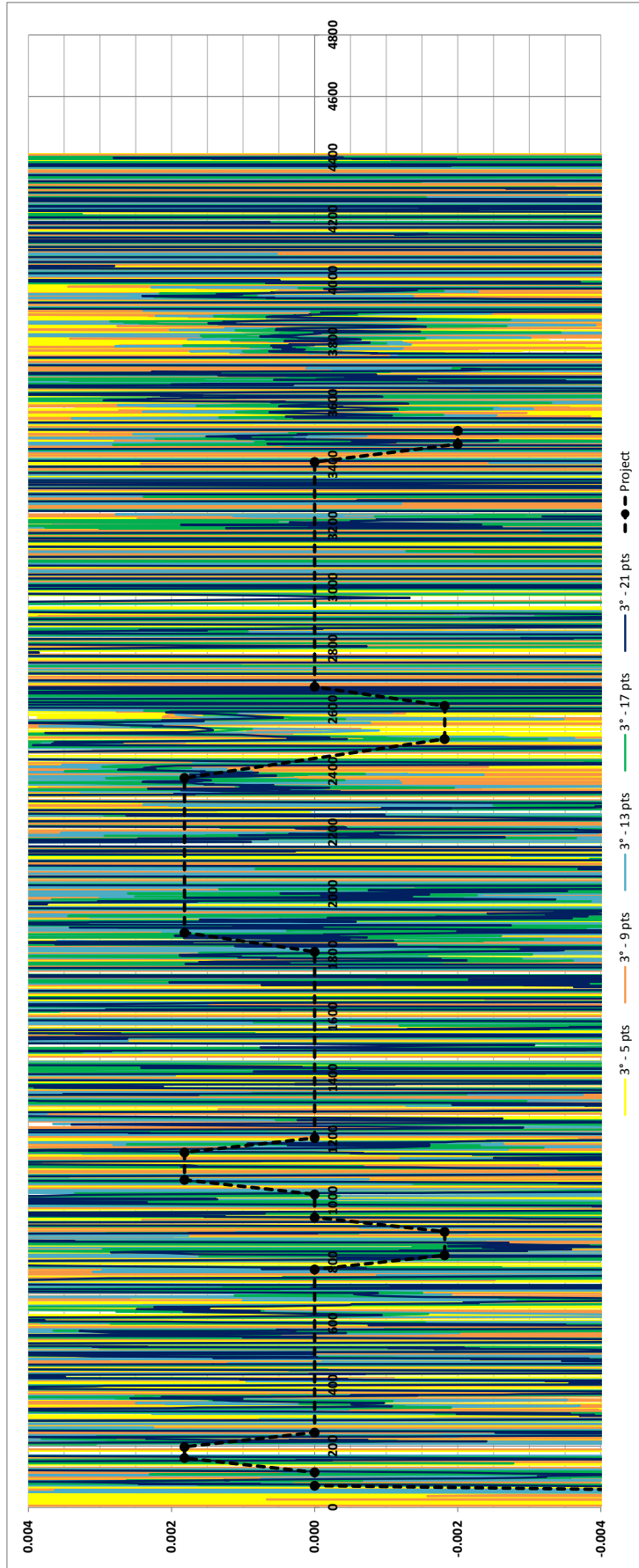
IMU Trajectory - 2nd degree



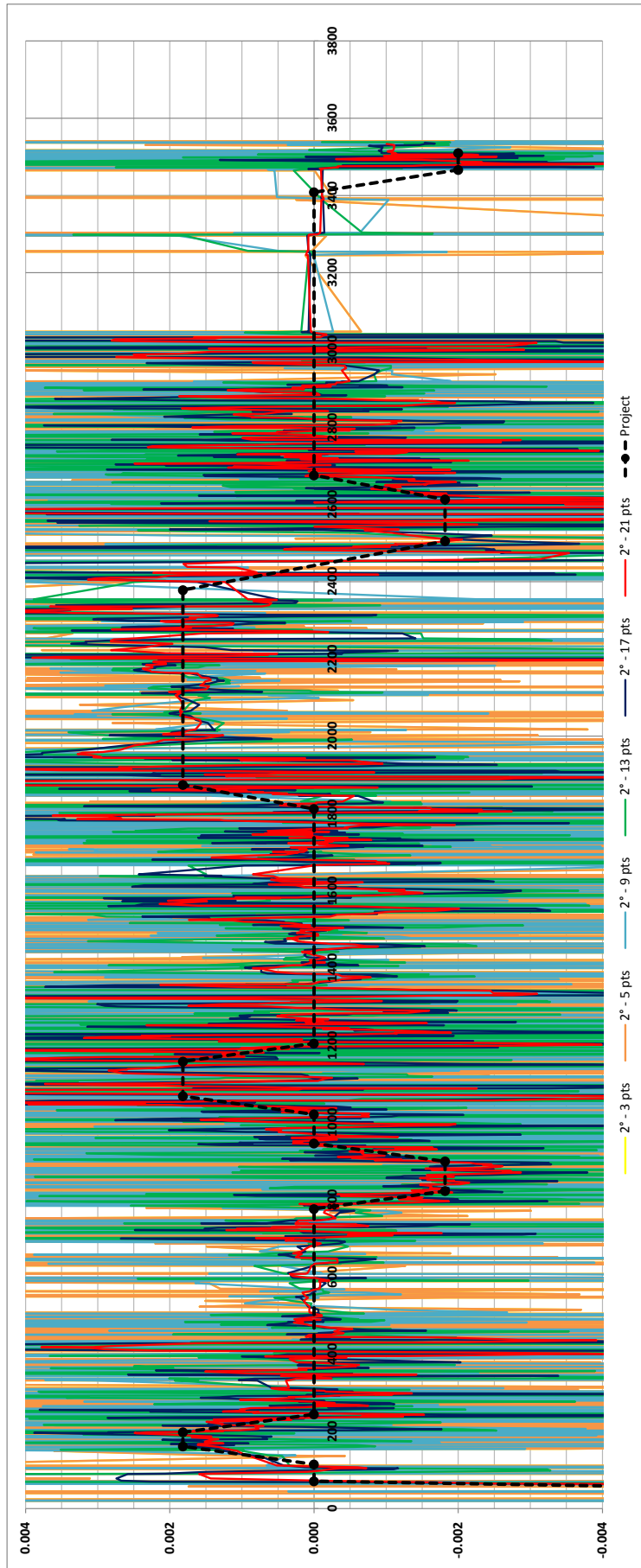
IMU Trajectory - 3rd degree



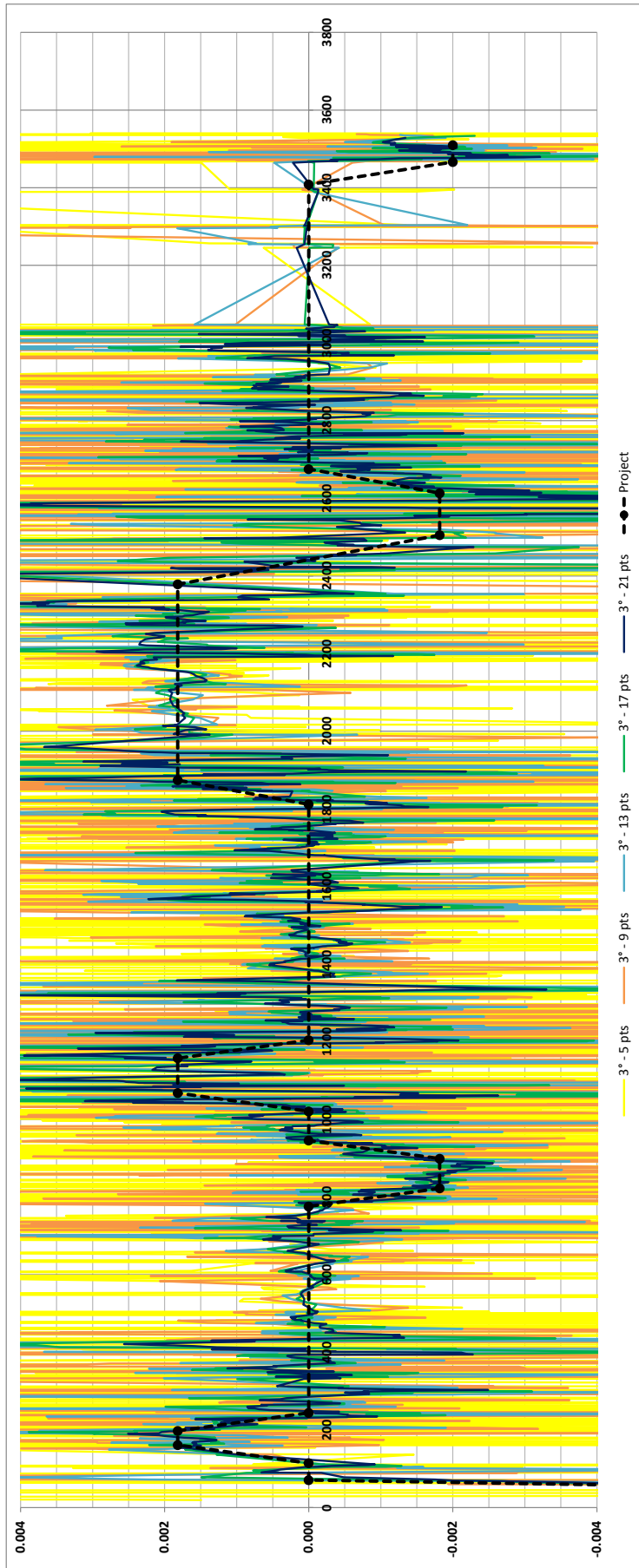
IMU Cigline - 2nd degree



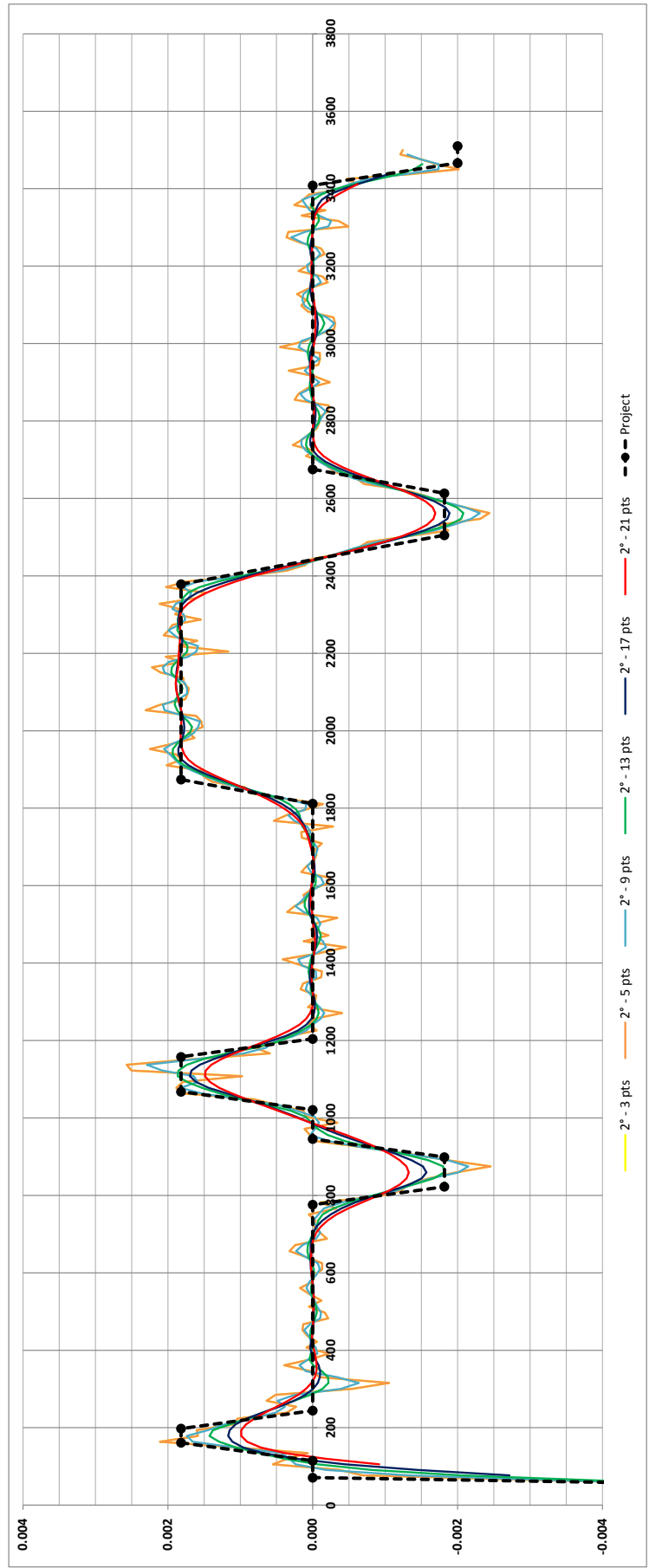
IMU Cigline - 3rd degree



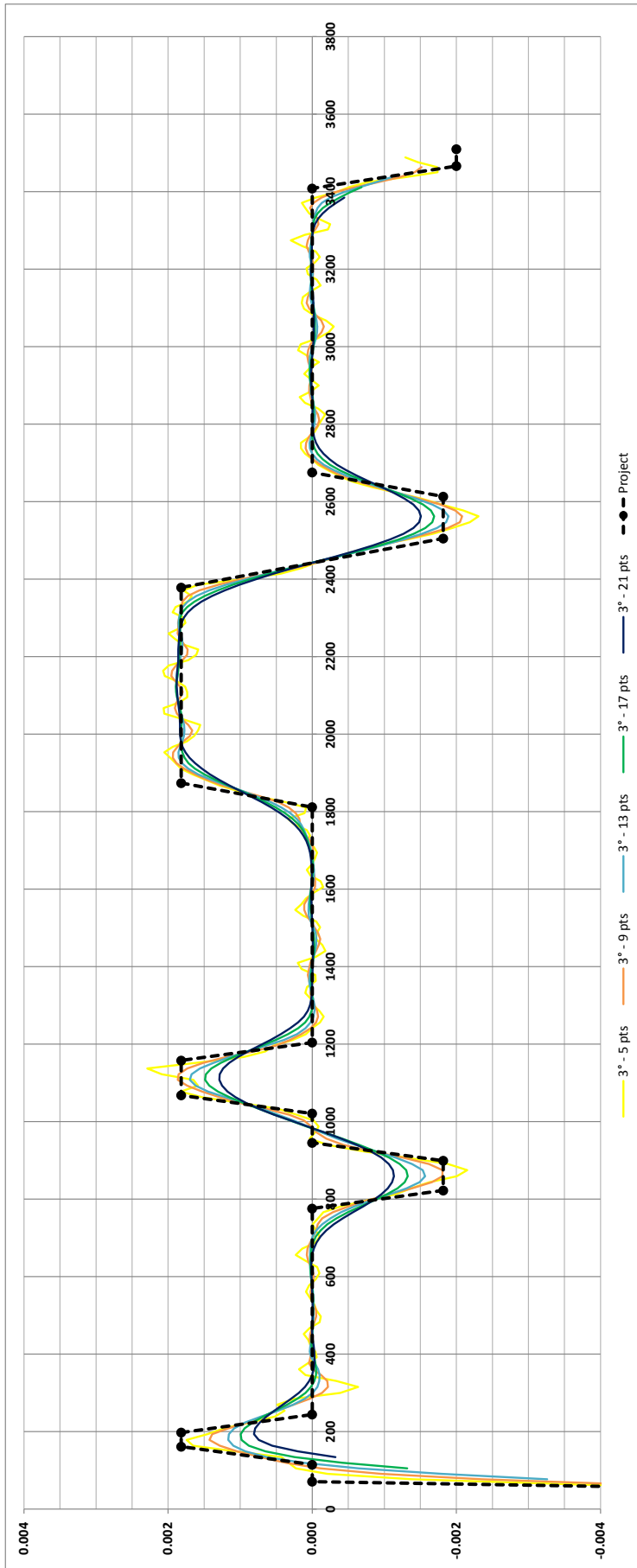
IMU Centerline - 2nd degree



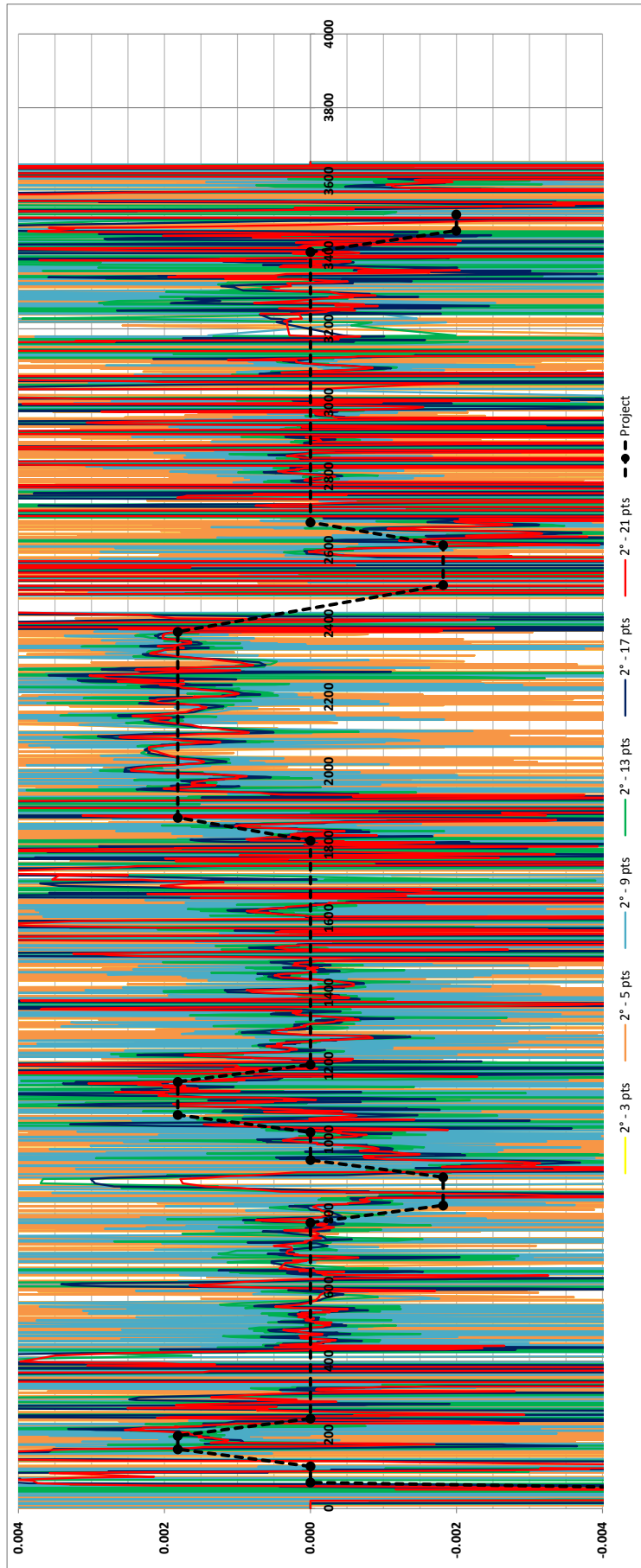
IMU Centerline - 3rd degree



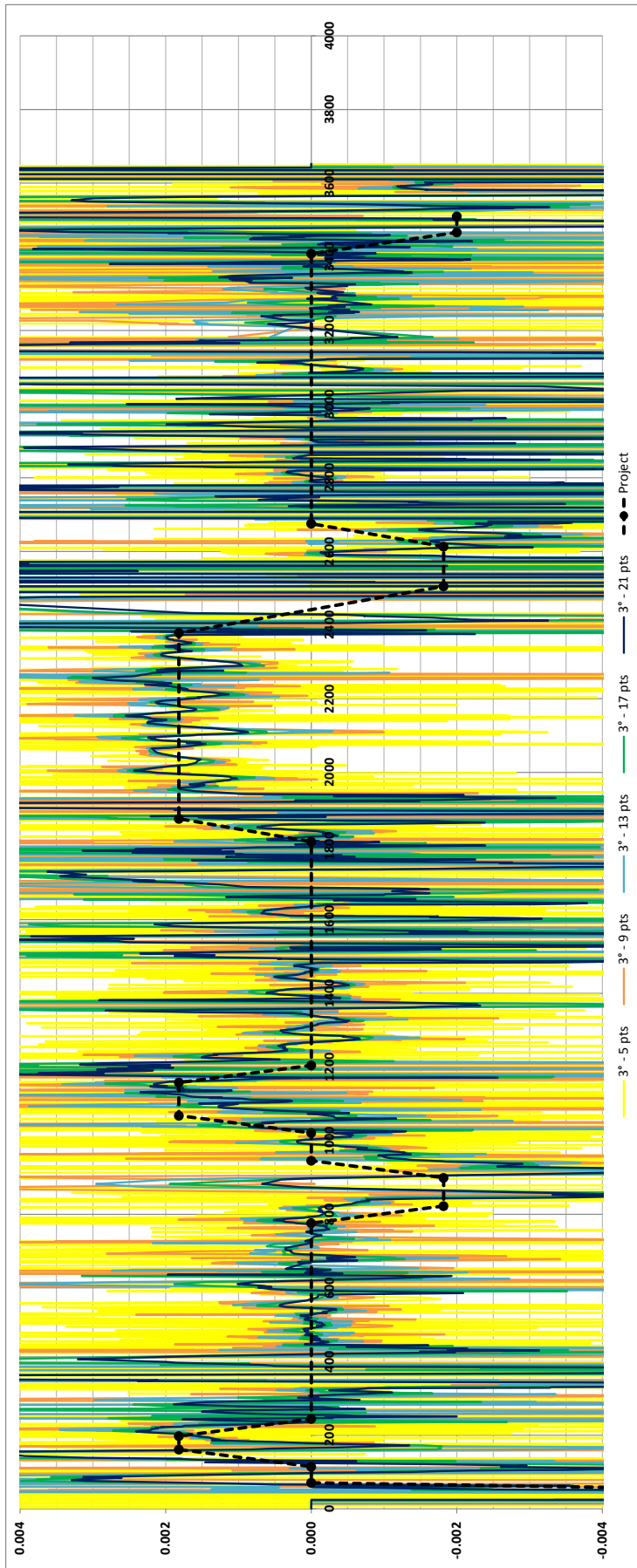
GPS Trajectory - 2nd degree



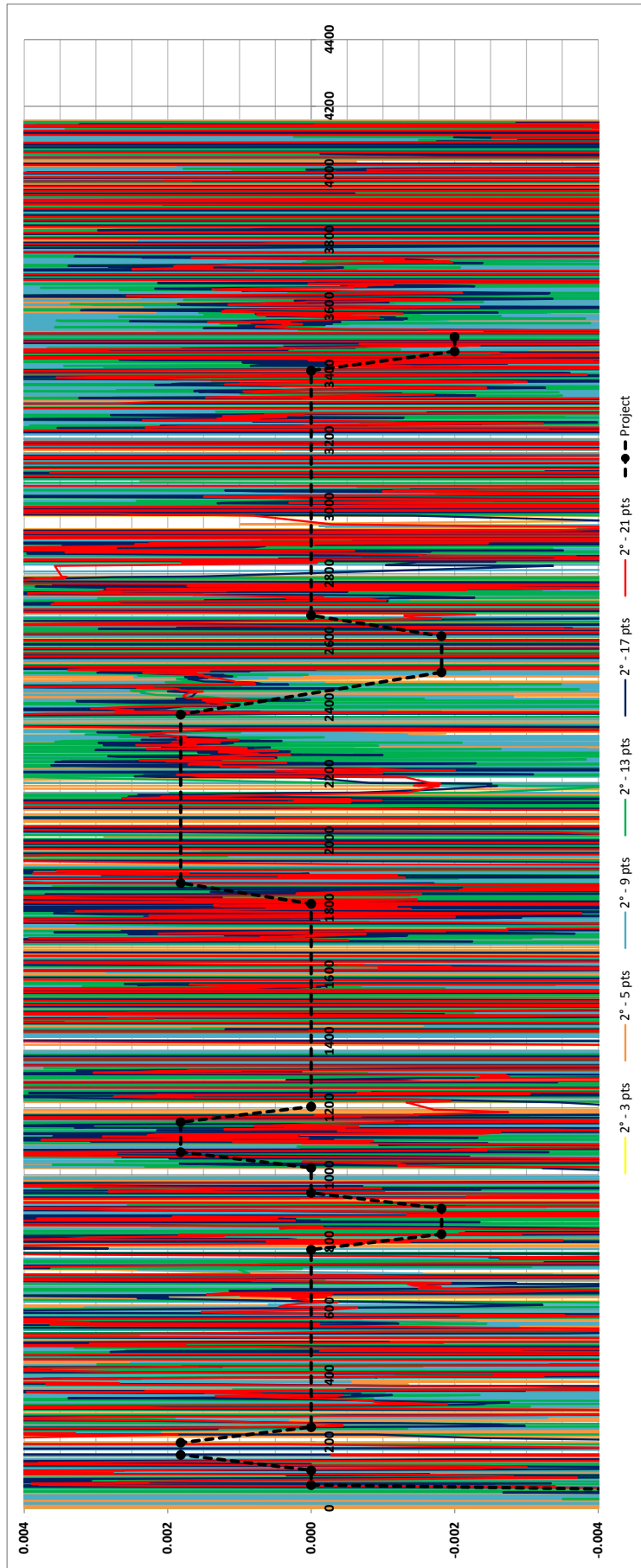
GPS Trajectory - 3rd degree



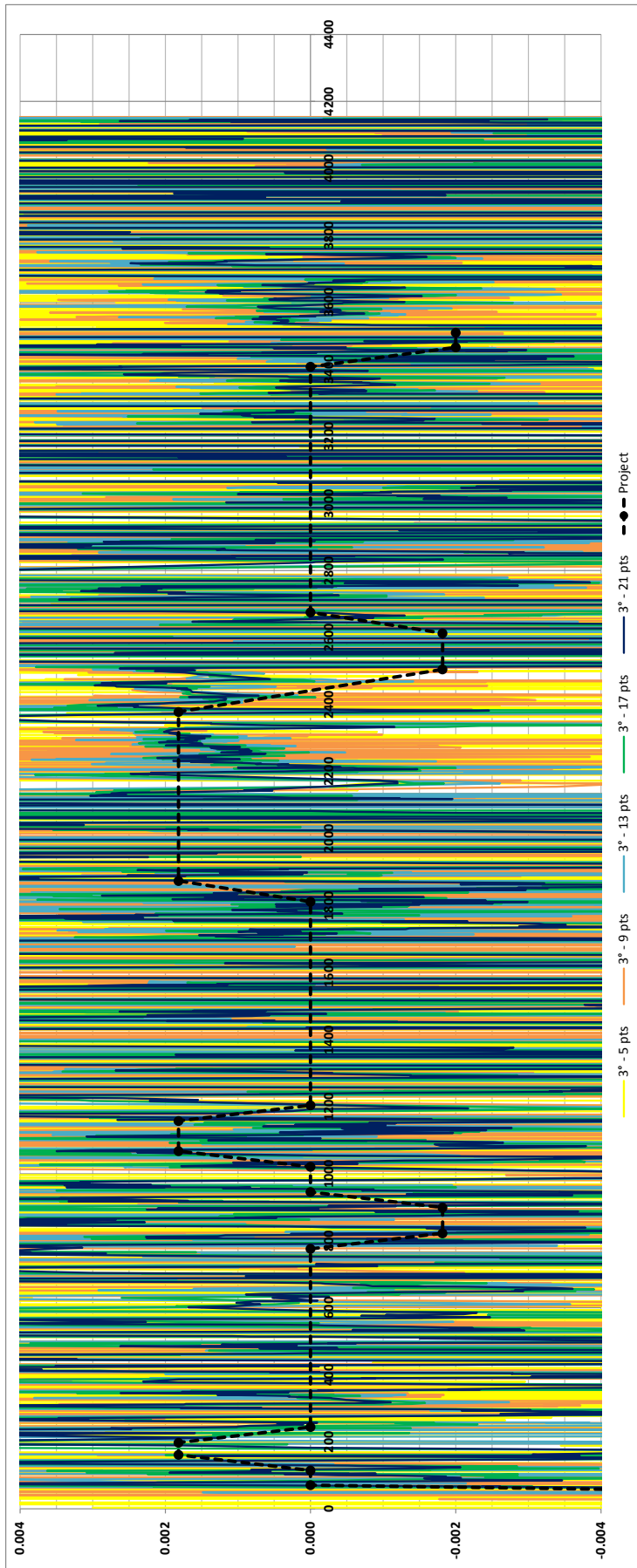
GPS Latline - 2nd degree



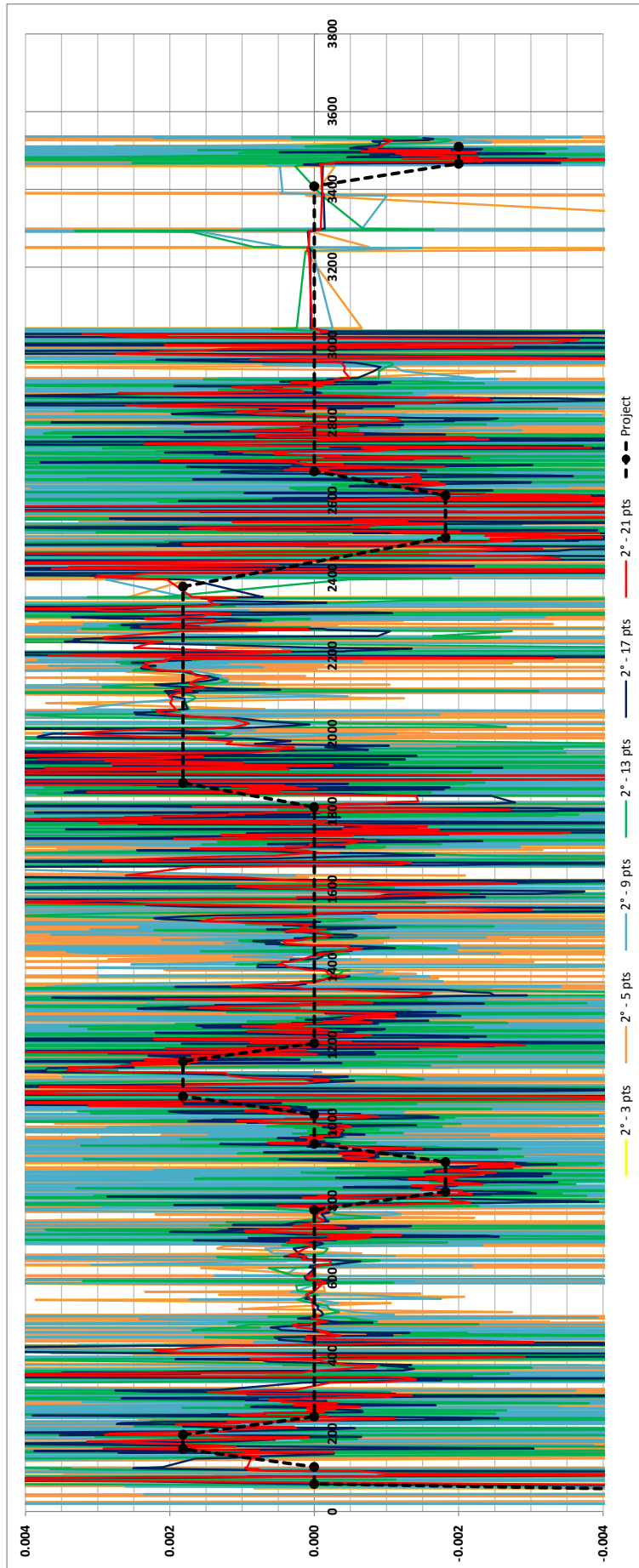
GPS Latline - 3rd degree



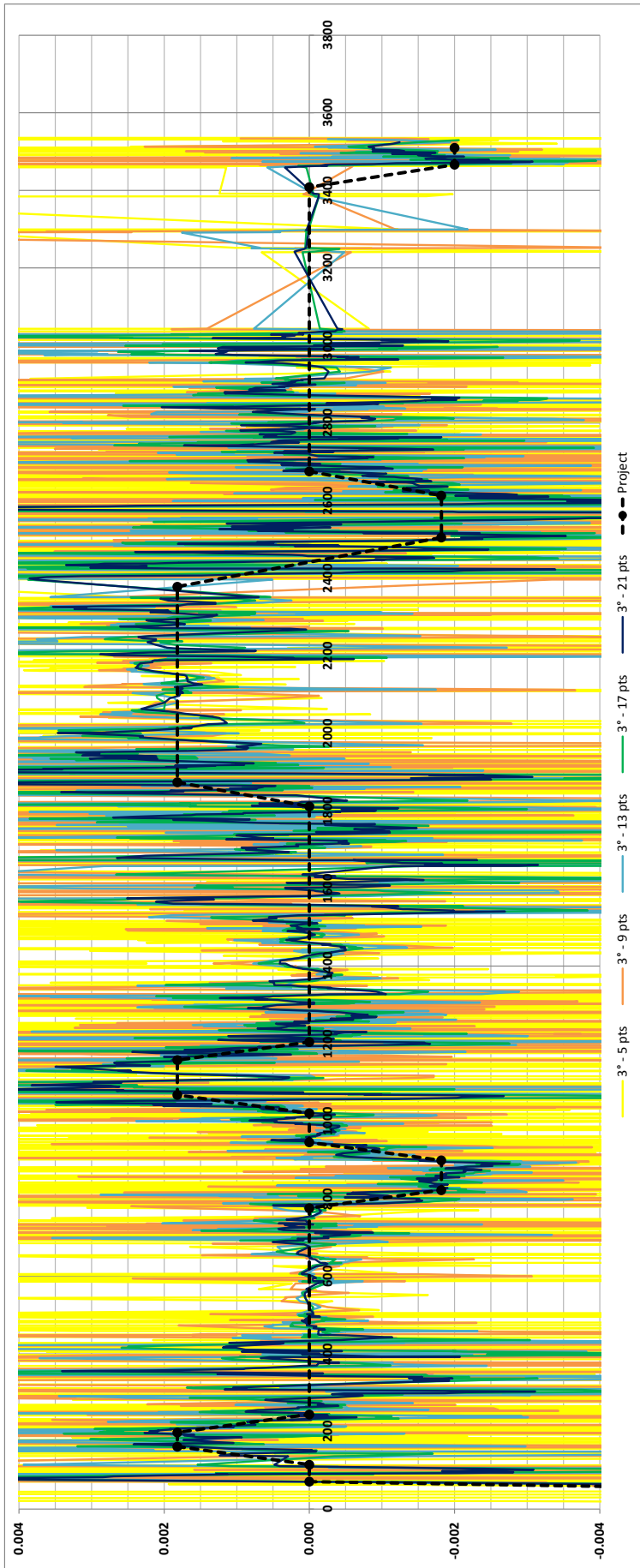
GPS Cigline - 2nd degree



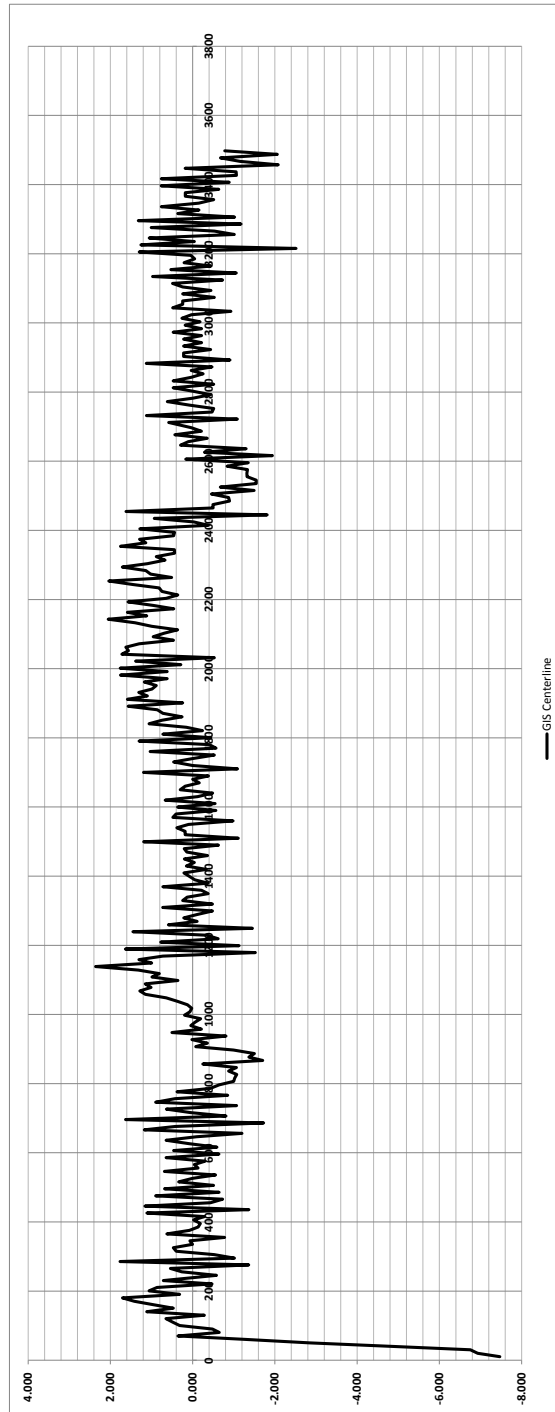
GPS Cigline - 3rd degree



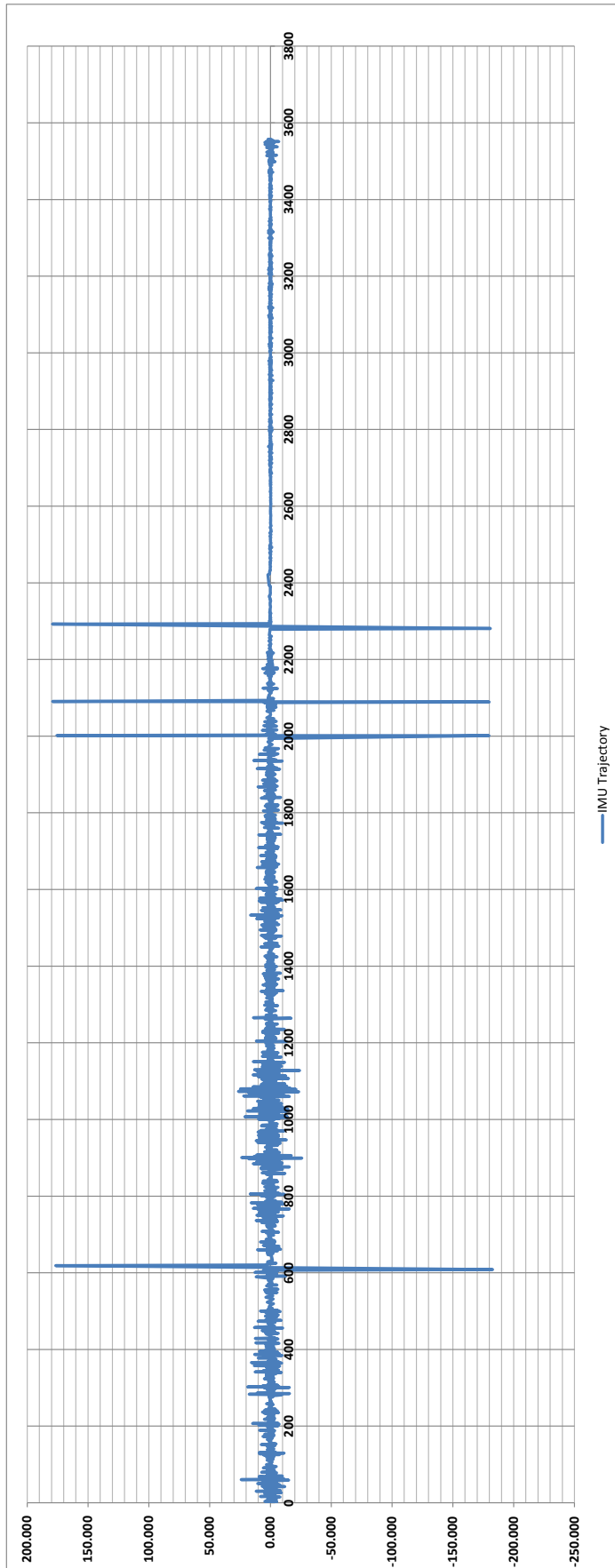
GPS Centerline - 2nd degree



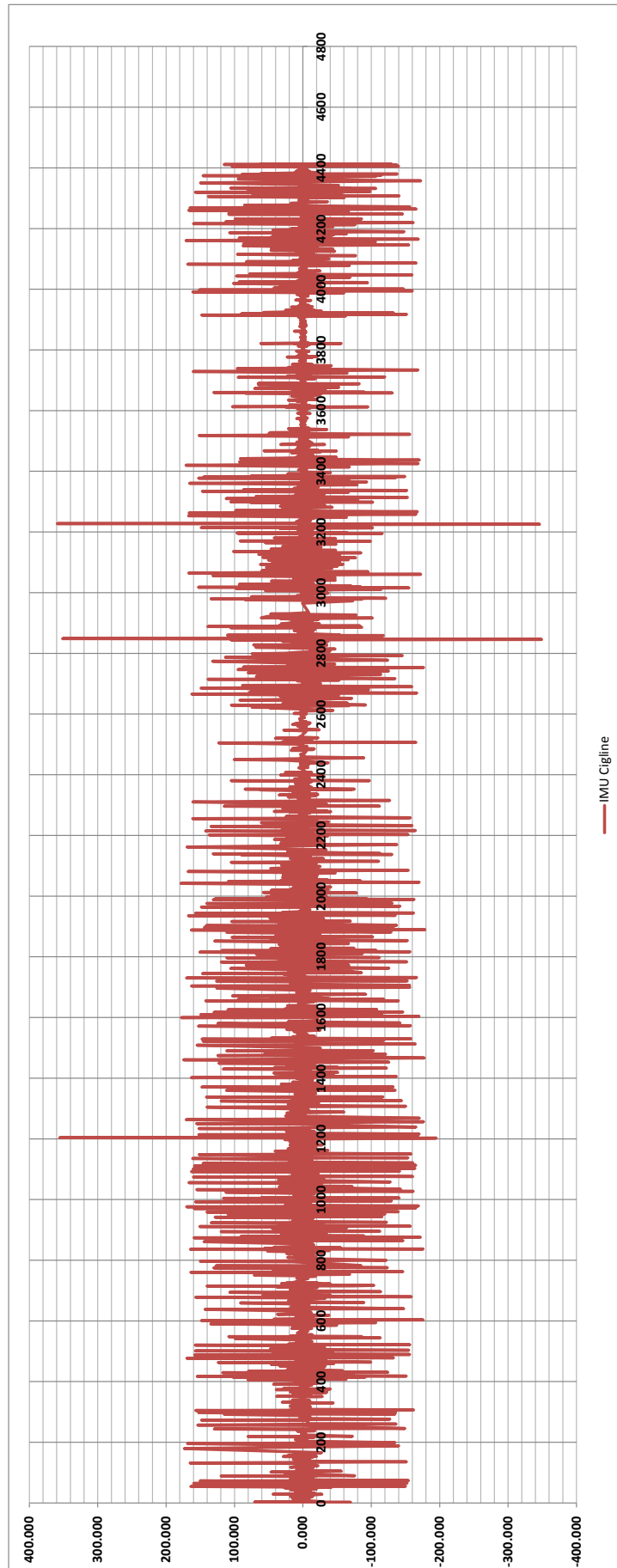
GPS Centerline - 3rd degree



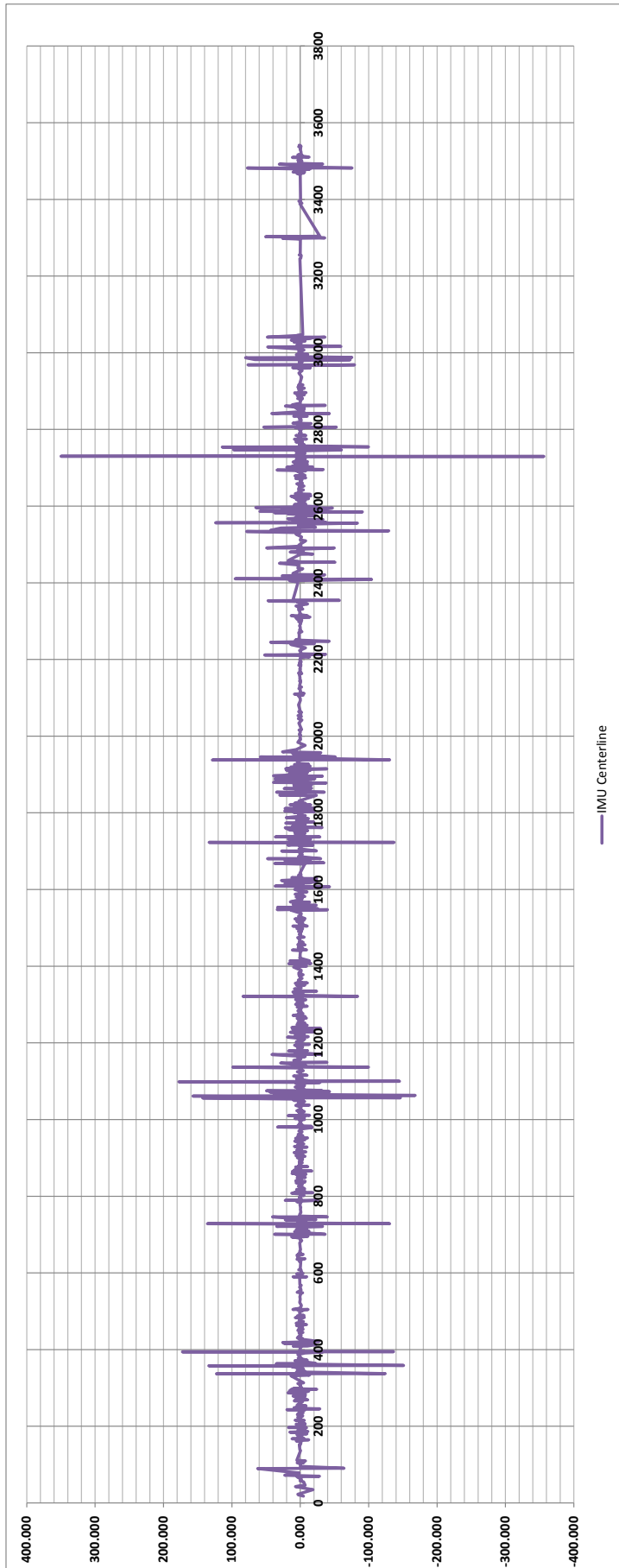
GIS Centerline



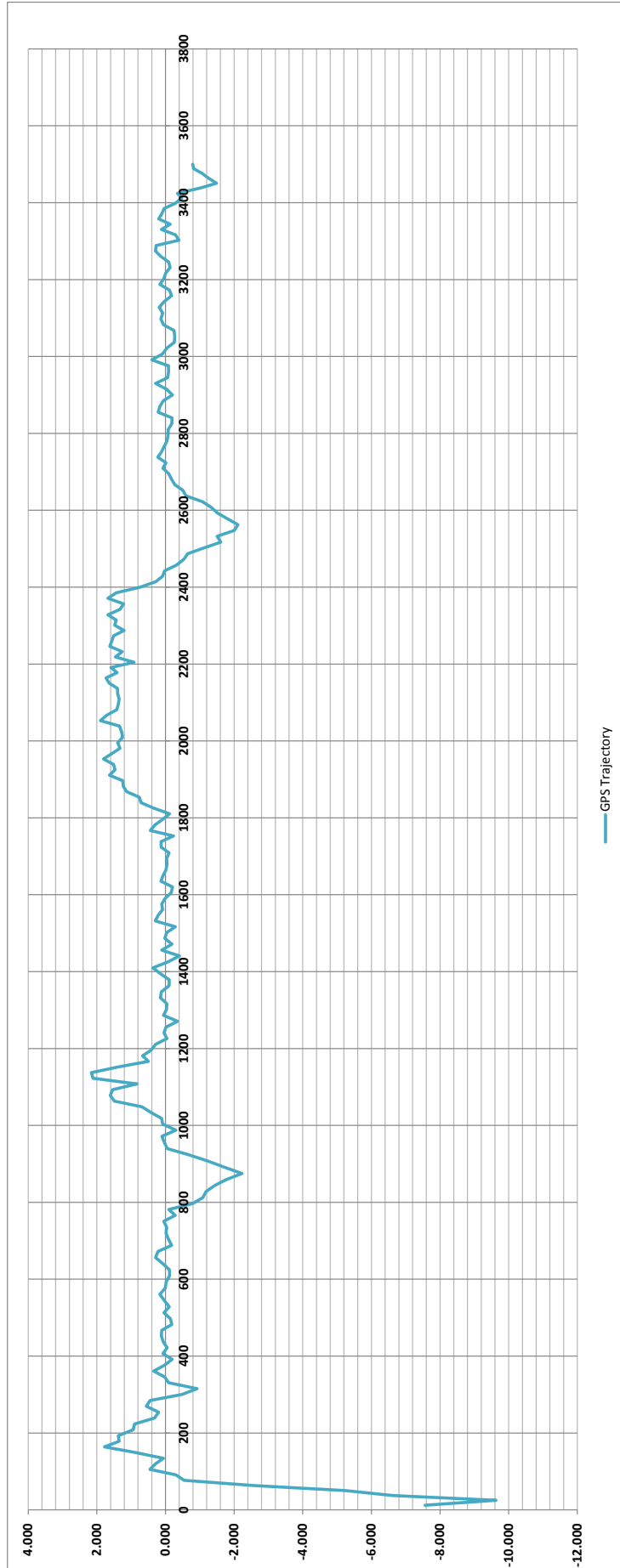
IMU Trajectory



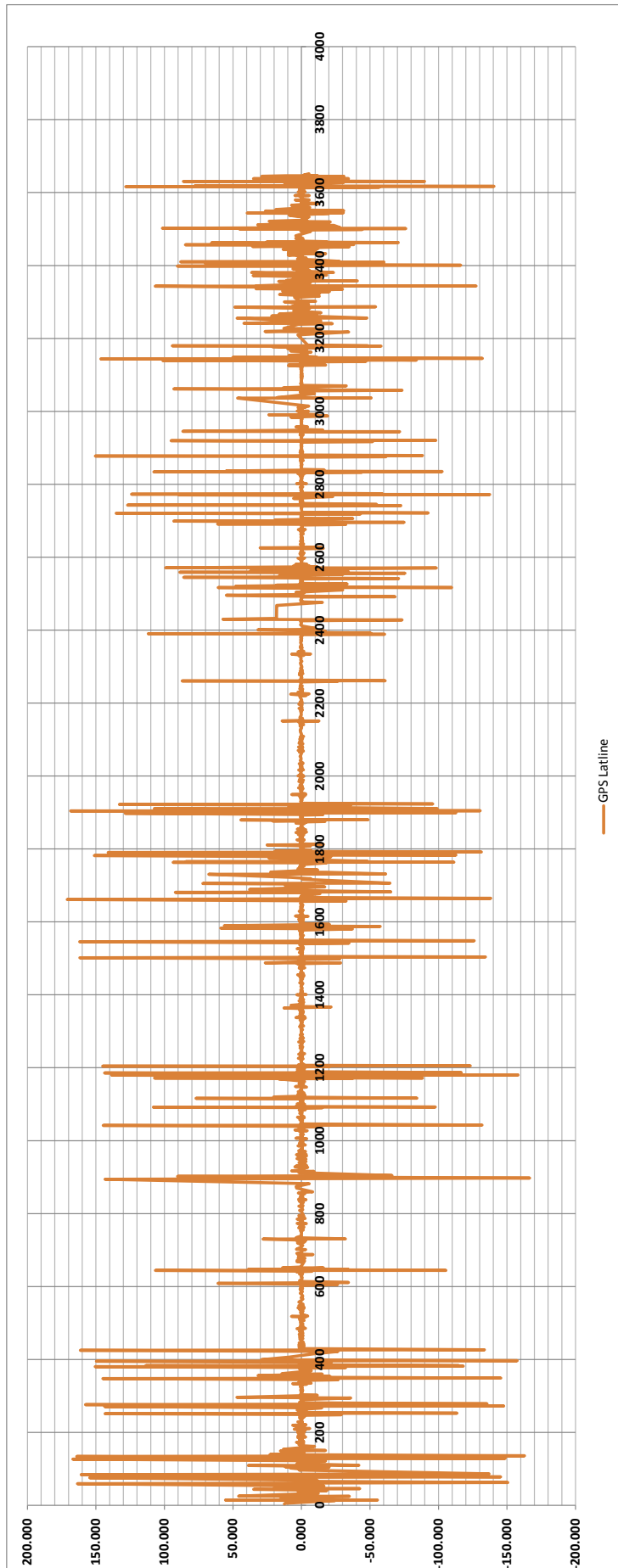
IMU Cigline



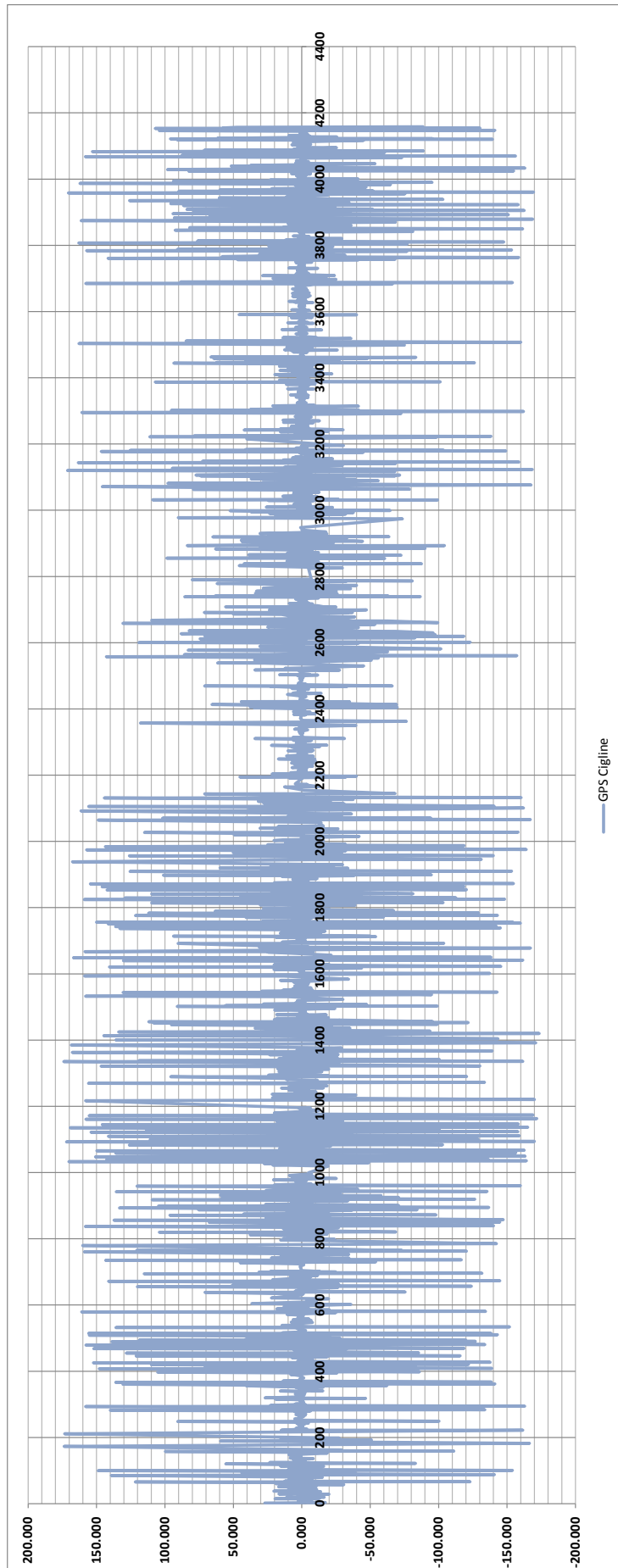
IMU Centerline



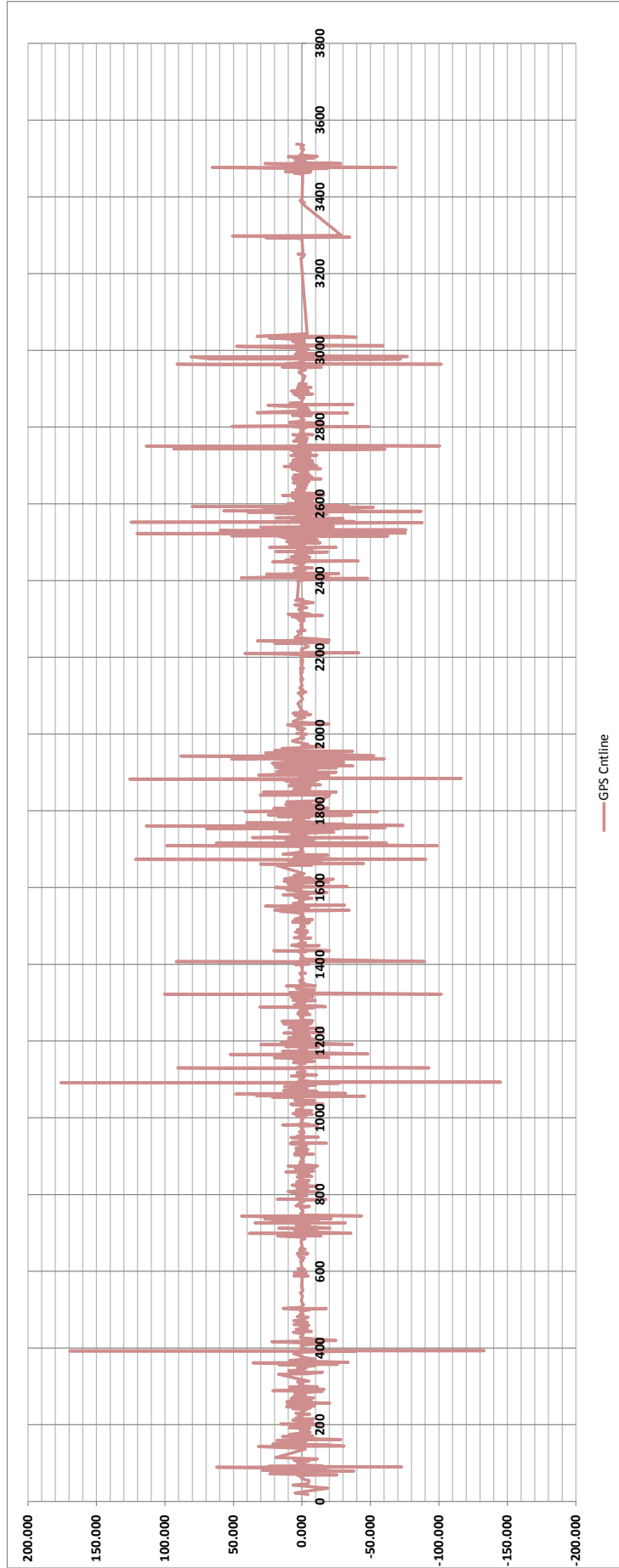
GPS Trajectory



GPS Latline

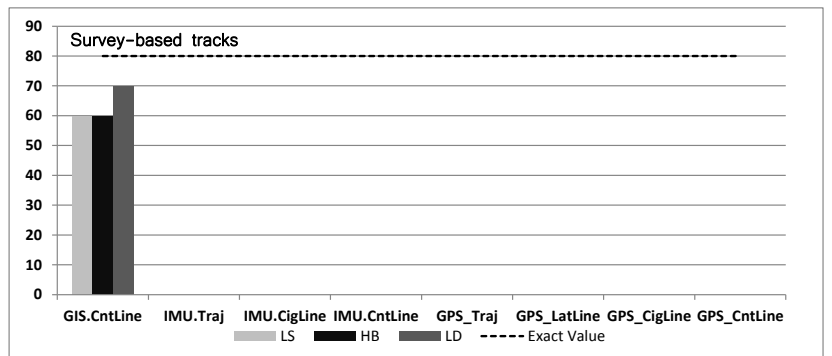
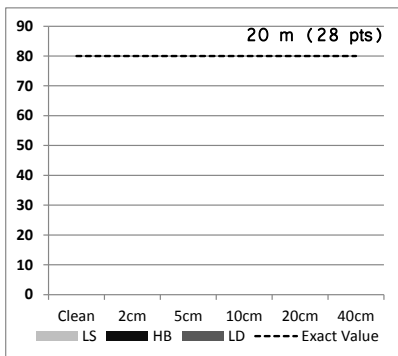
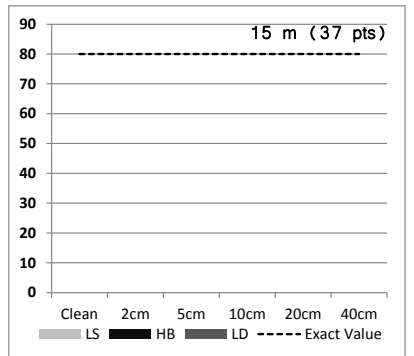
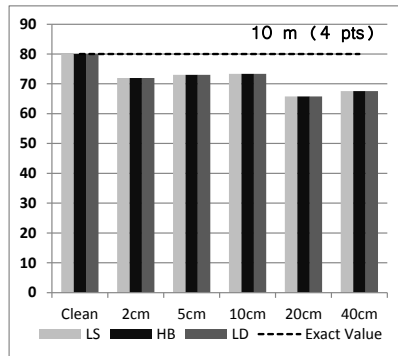
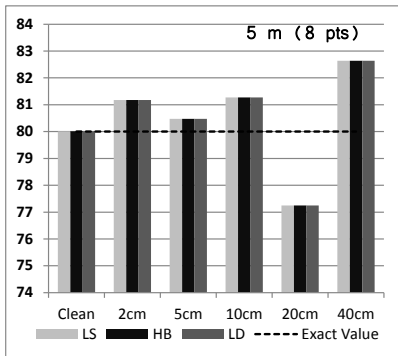
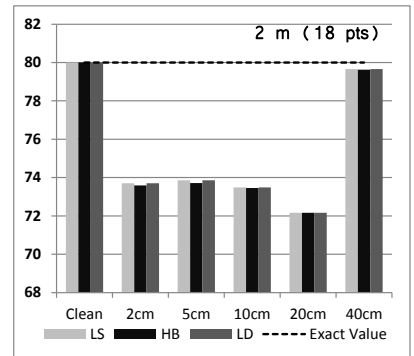
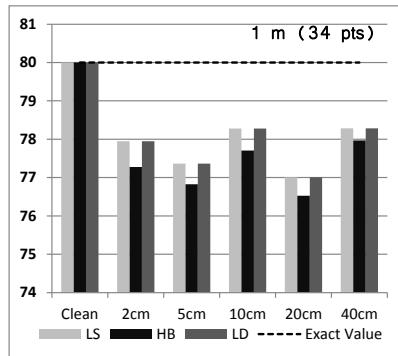
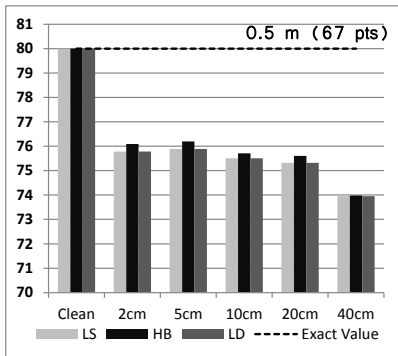


GPS Cigline

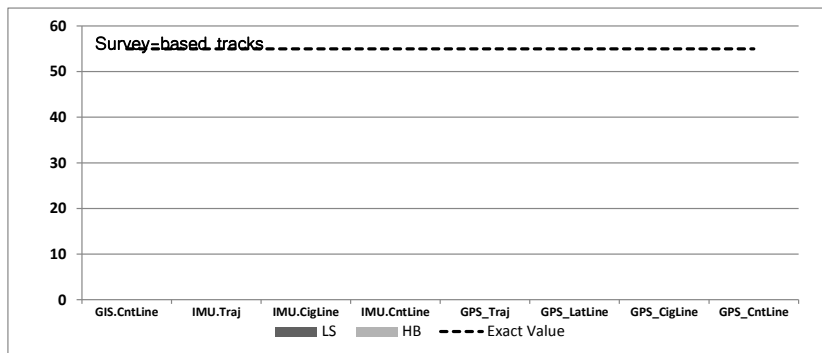
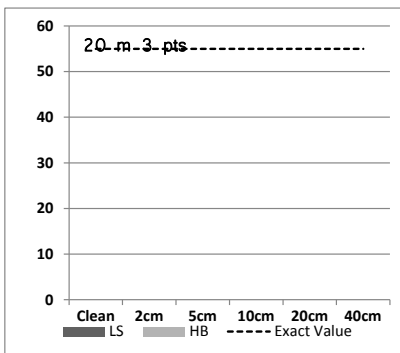
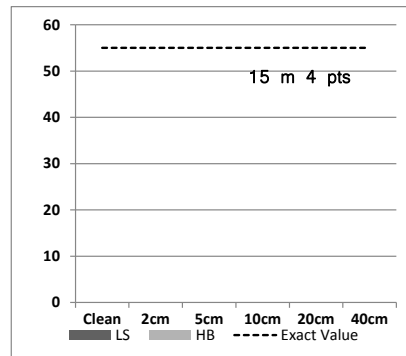
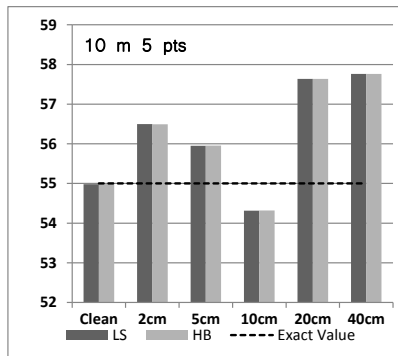
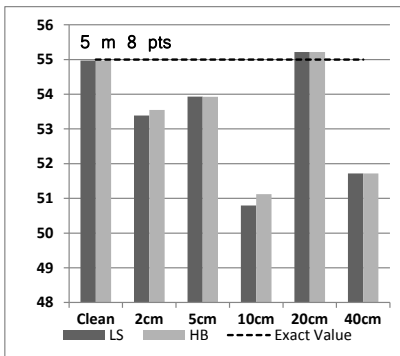
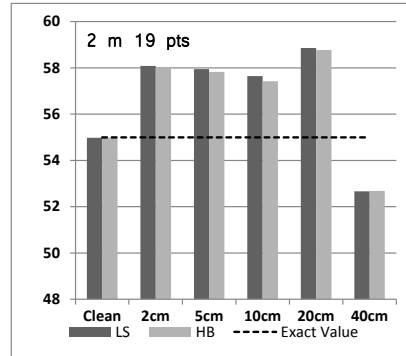
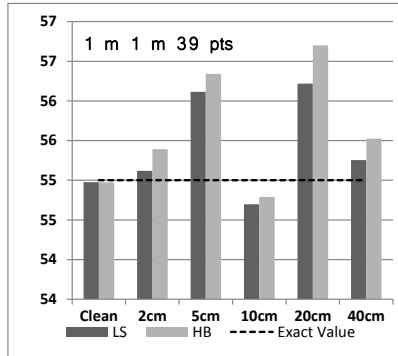
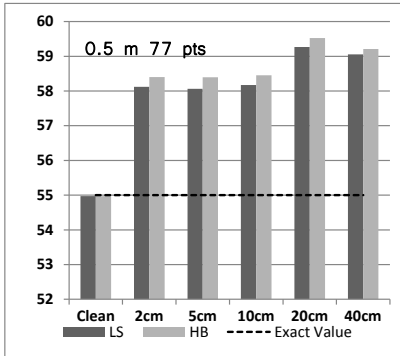


GPS Centerline

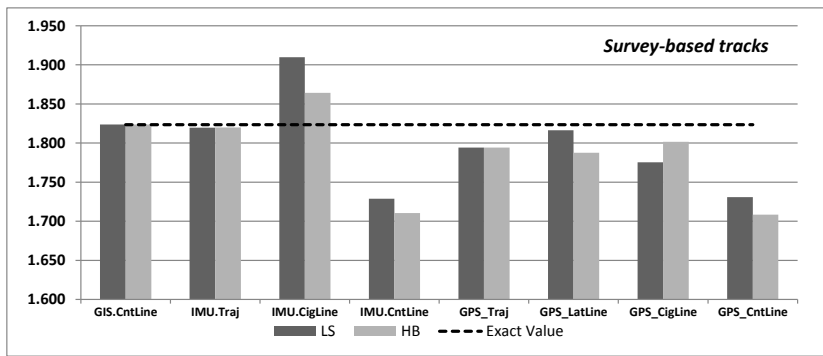
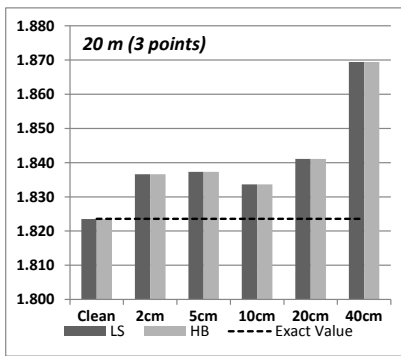
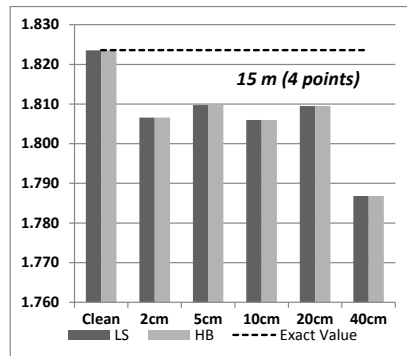
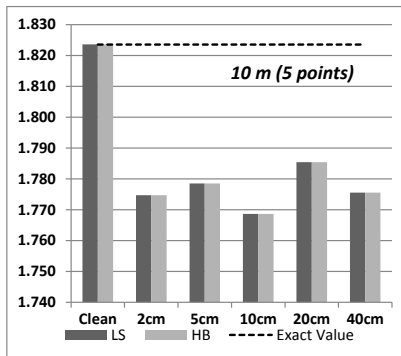
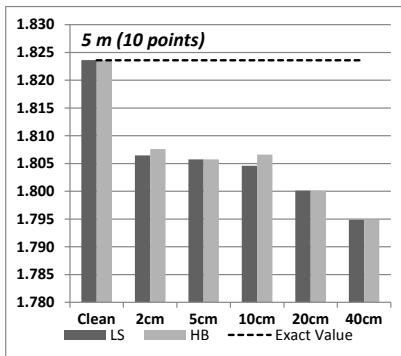
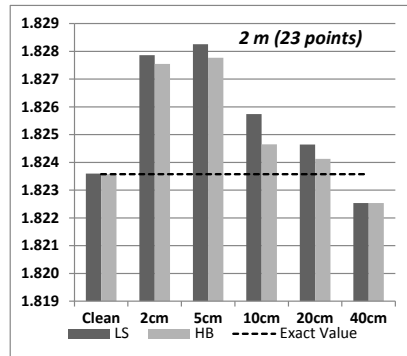
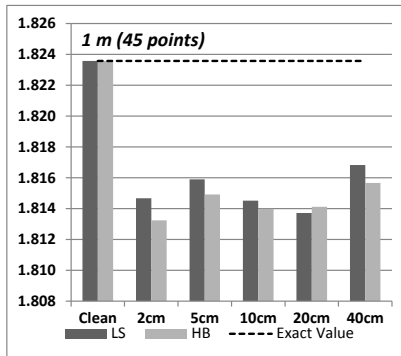
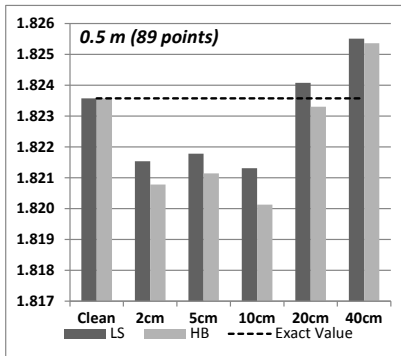
AL3 Fitting results



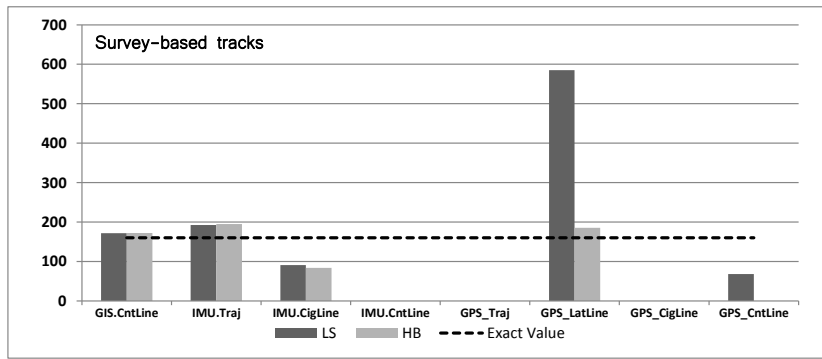
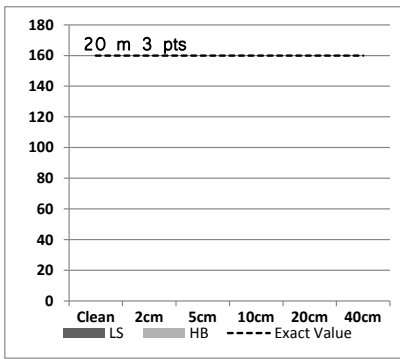
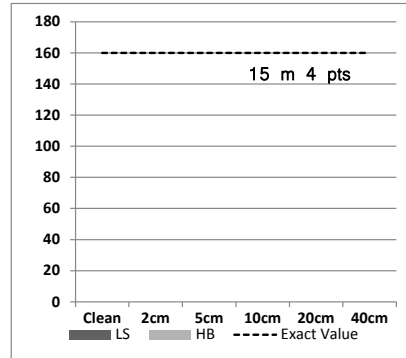
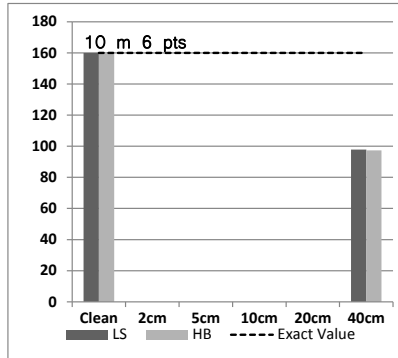
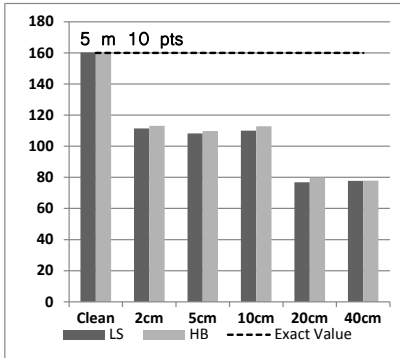
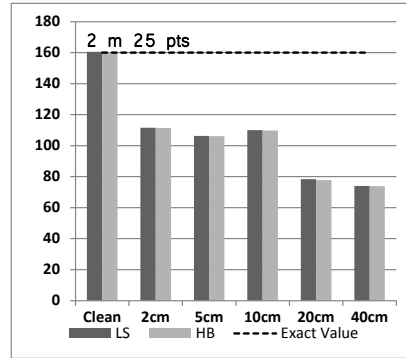
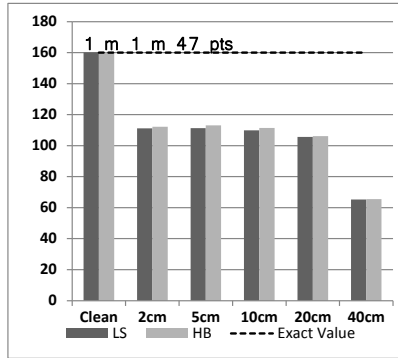
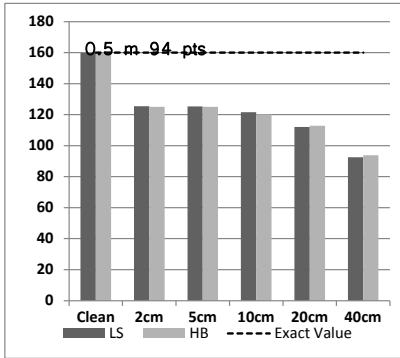
Circular curve #1



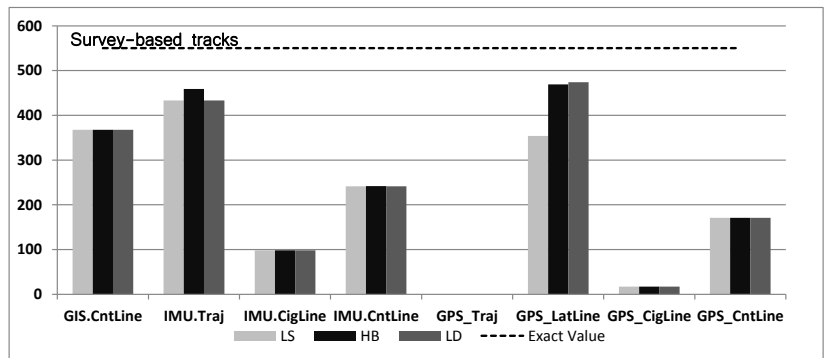
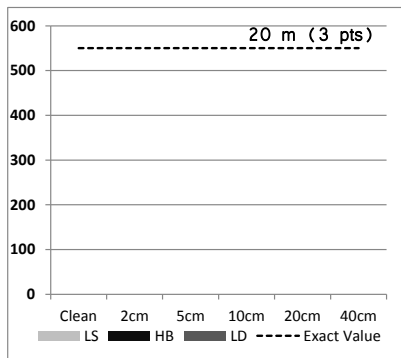
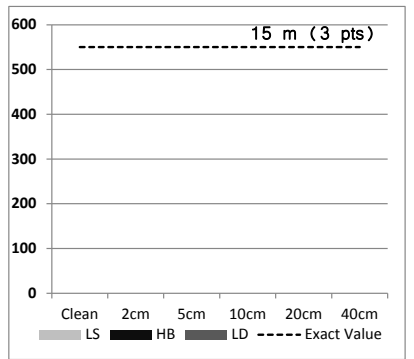
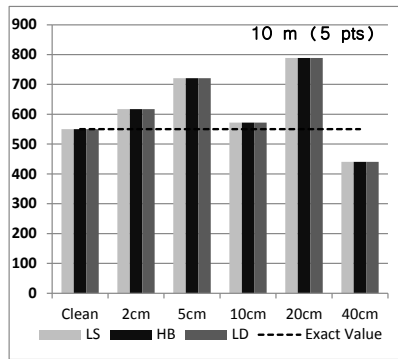
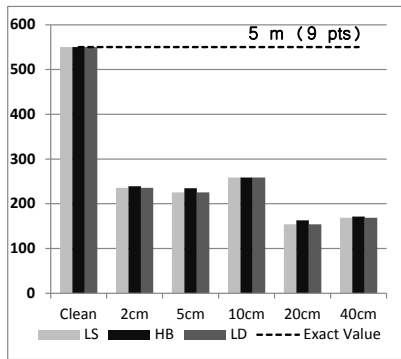
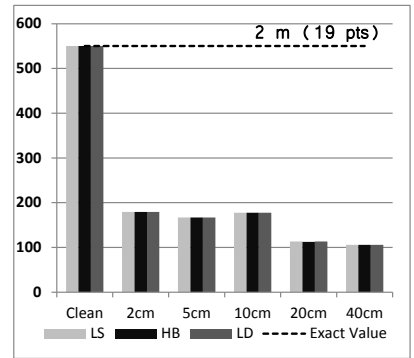
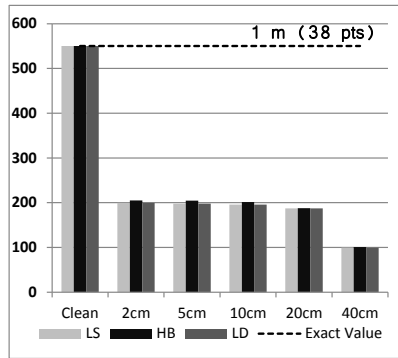
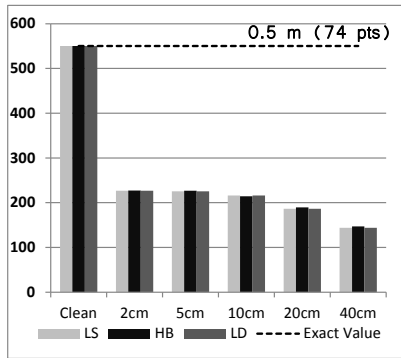
Clothoid #2



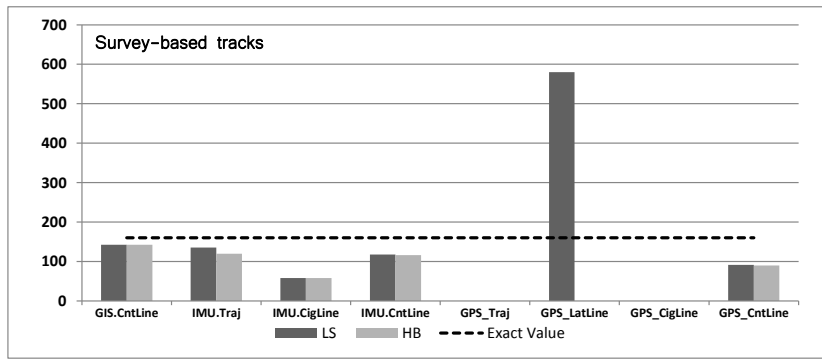
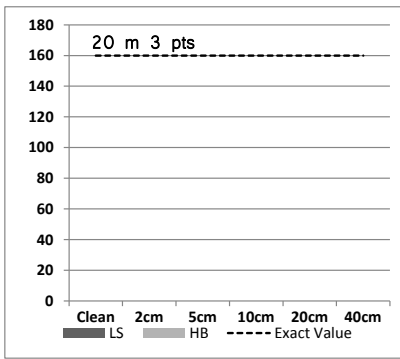
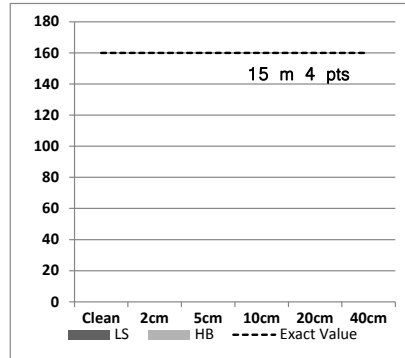
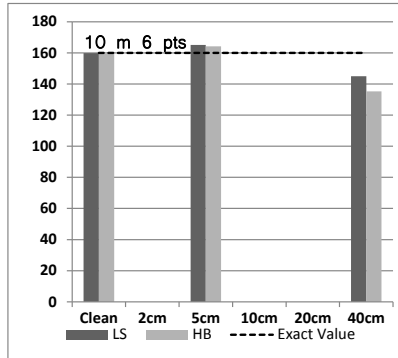
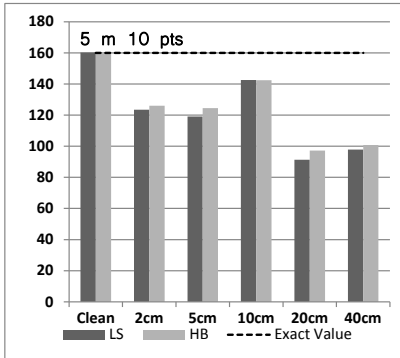
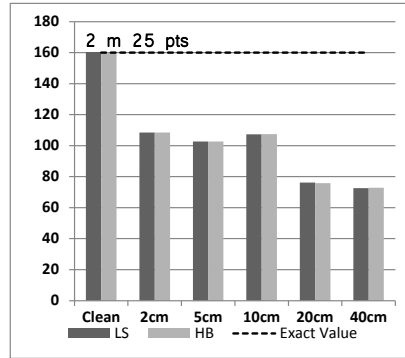
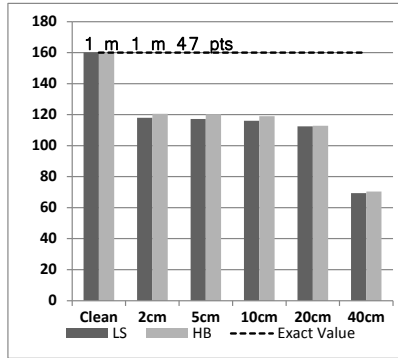
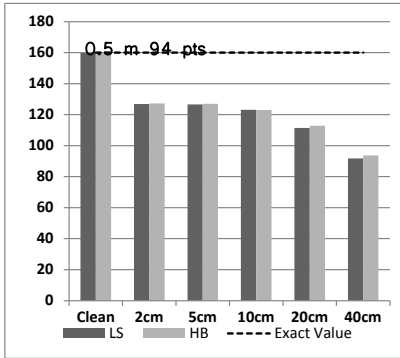
Tangent #3



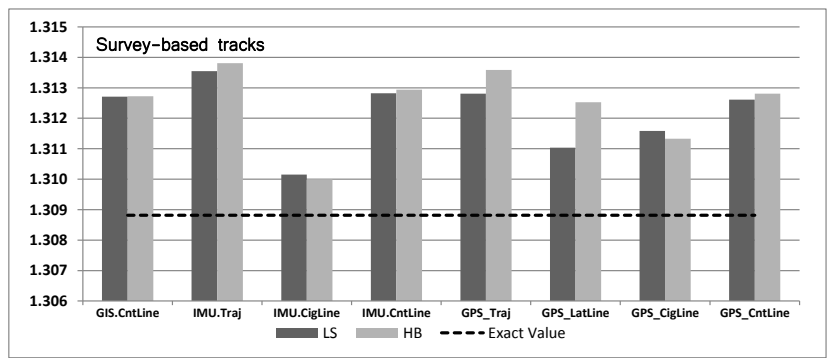
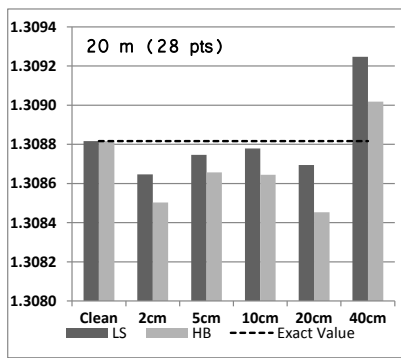
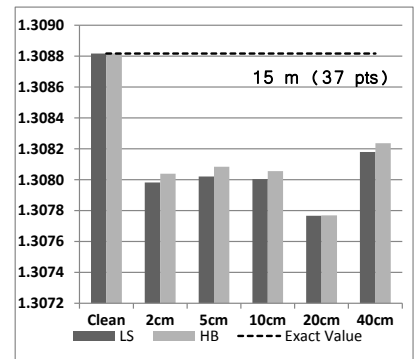
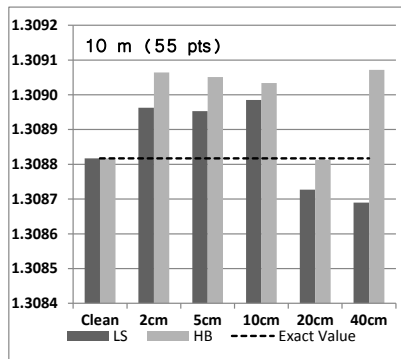
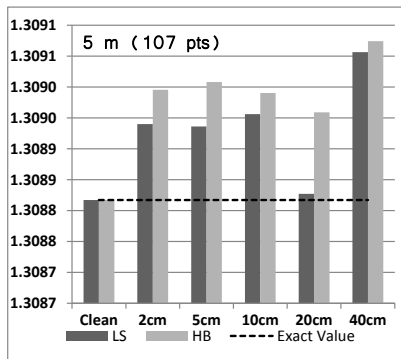
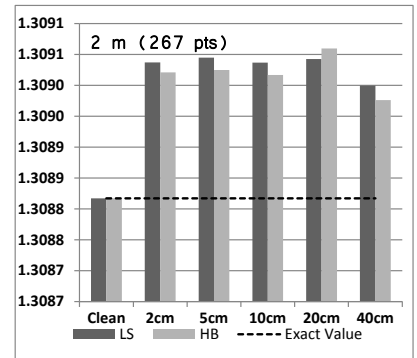
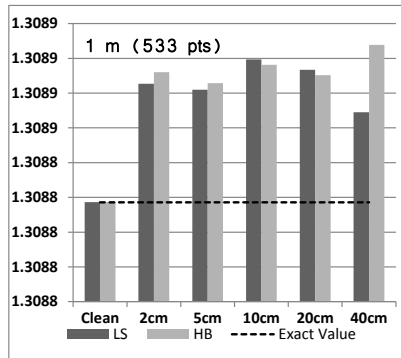
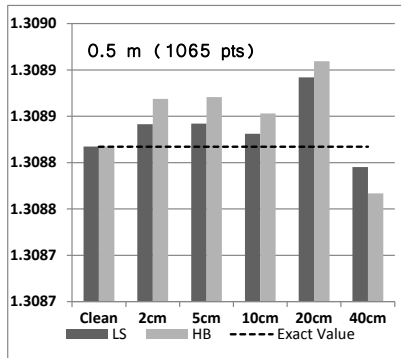
Clothoid #4



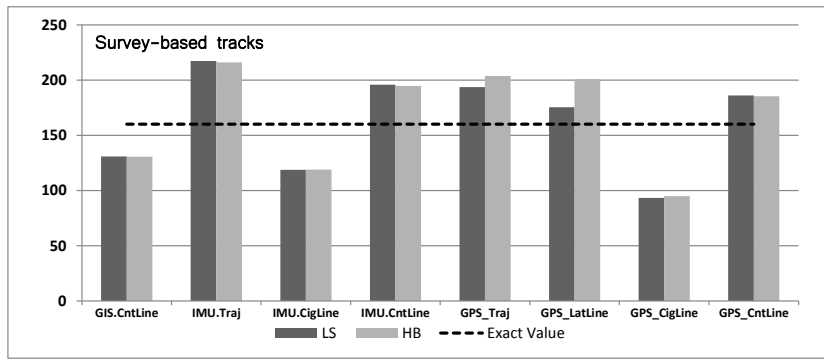
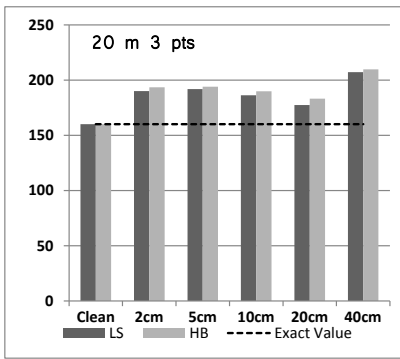
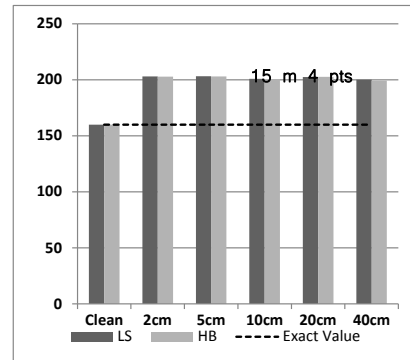
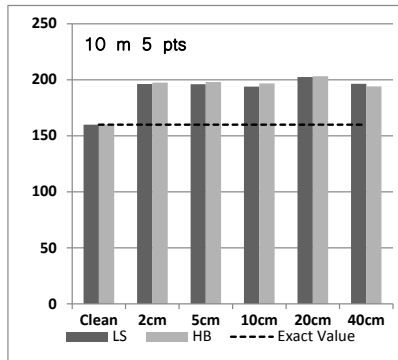
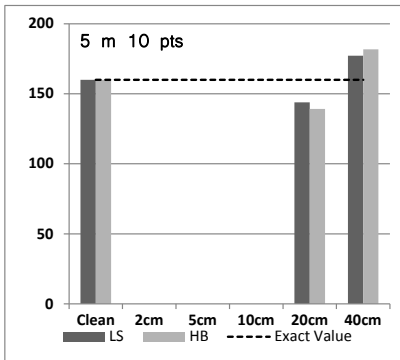
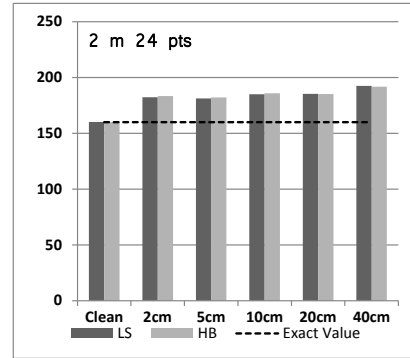
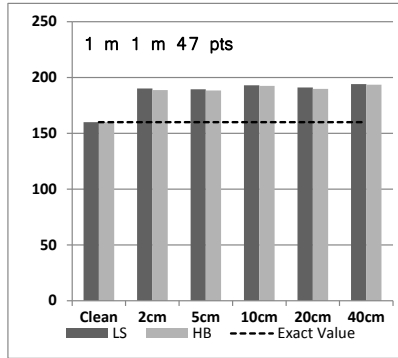
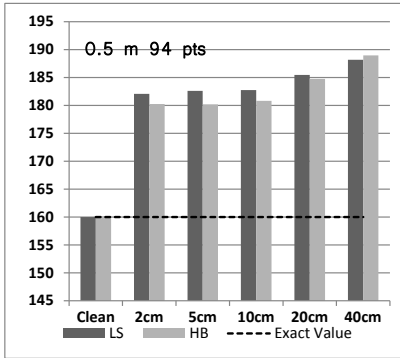
Circular Curve #5



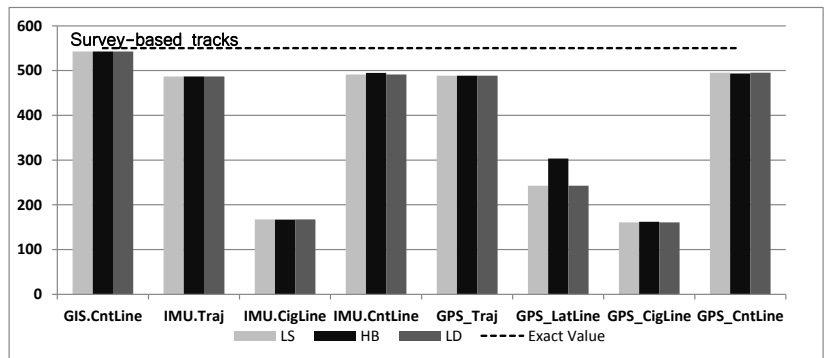
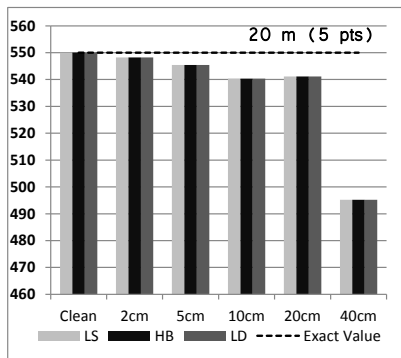
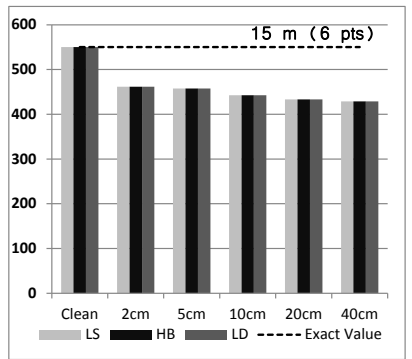
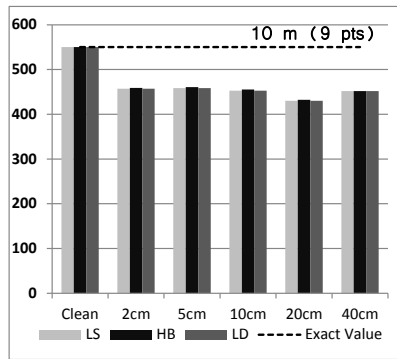
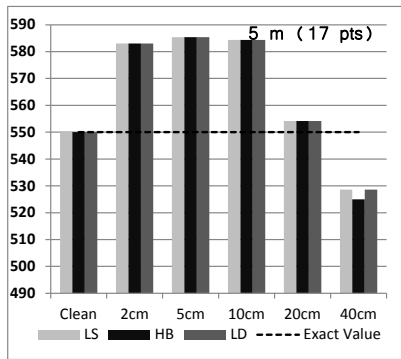
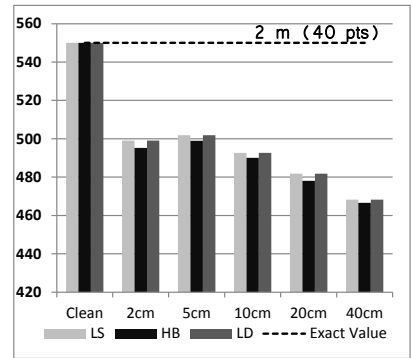
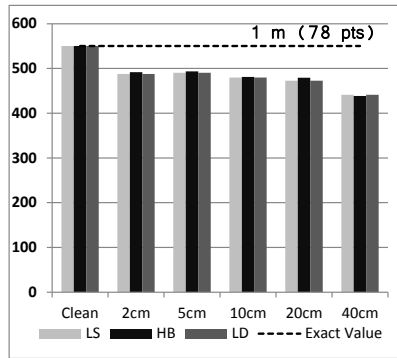
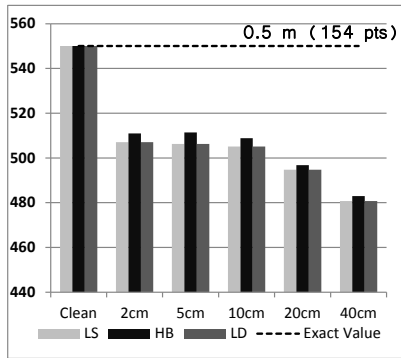
Clothoid #6



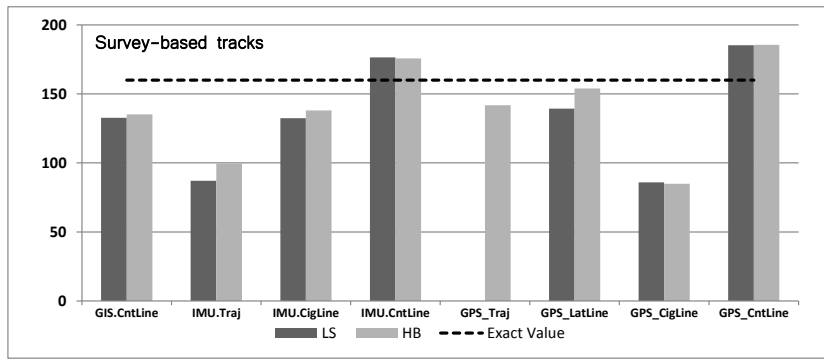
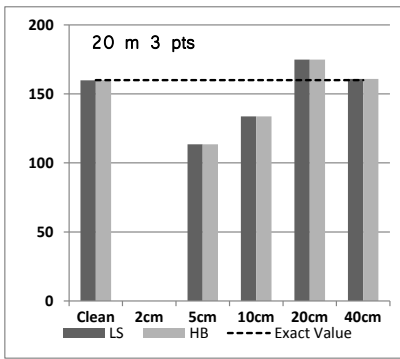
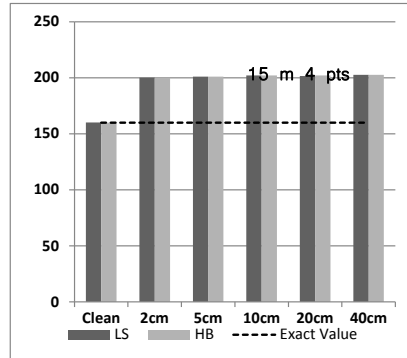
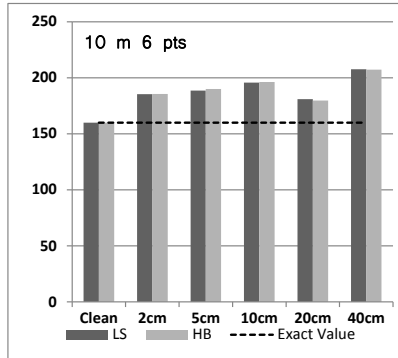
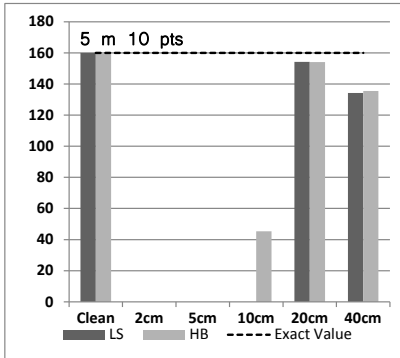
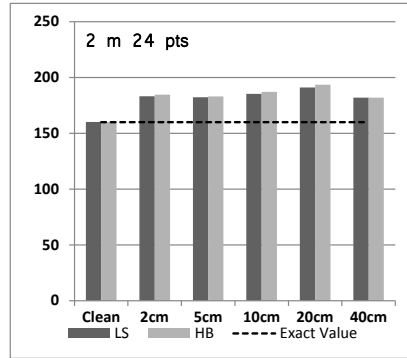
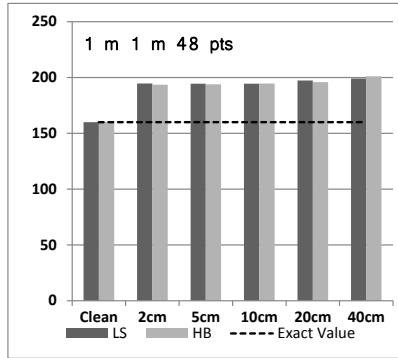
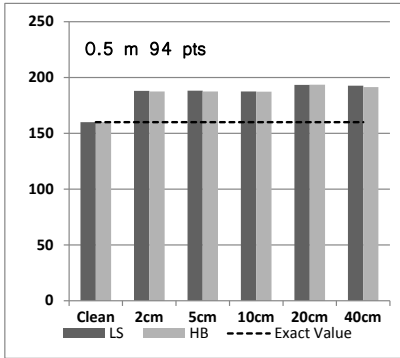
Tangent #7



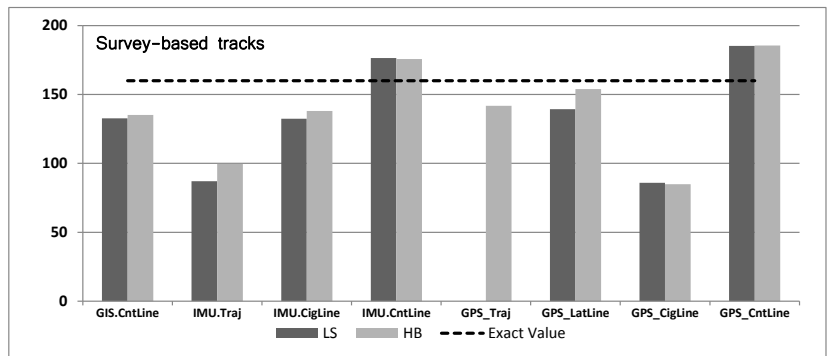
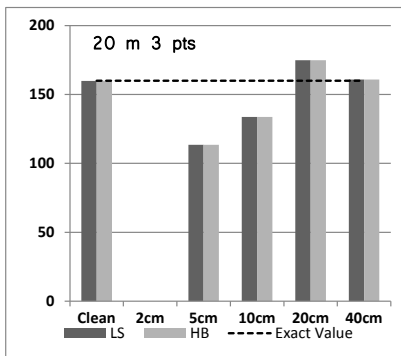
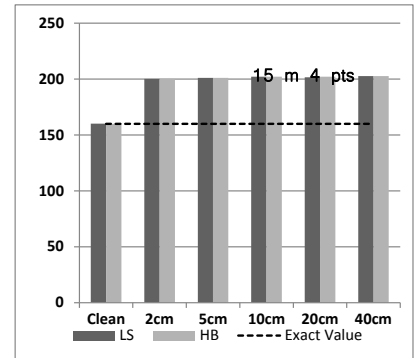
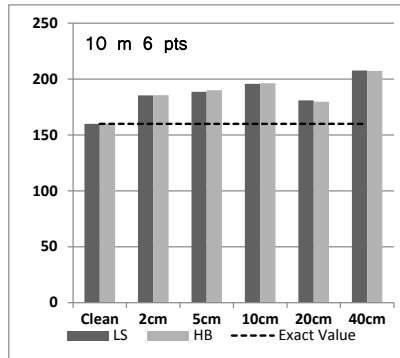
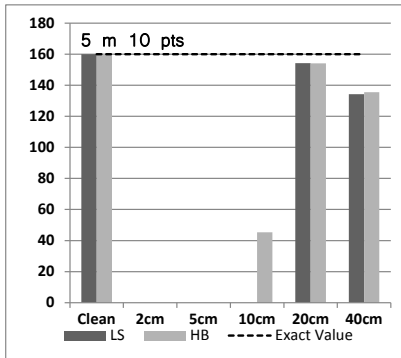
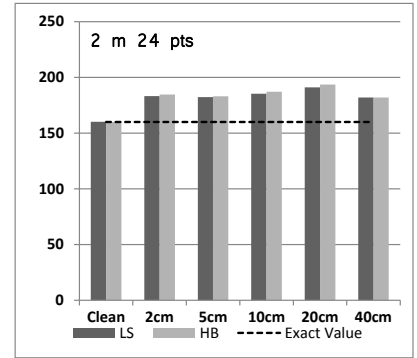
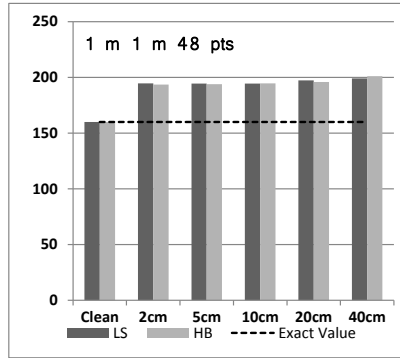
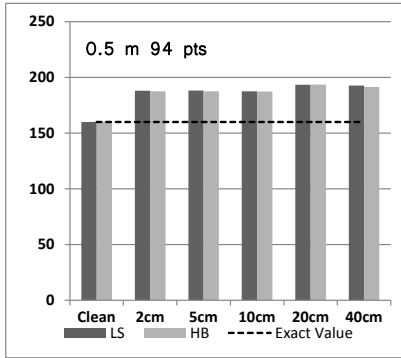
Clothoid #8



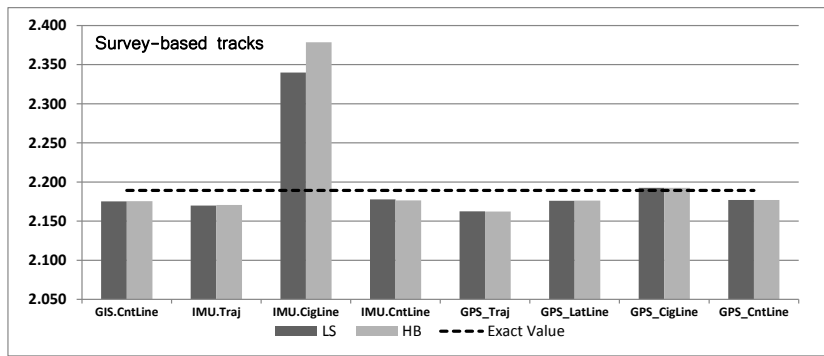
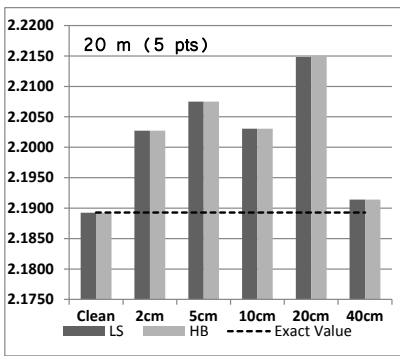
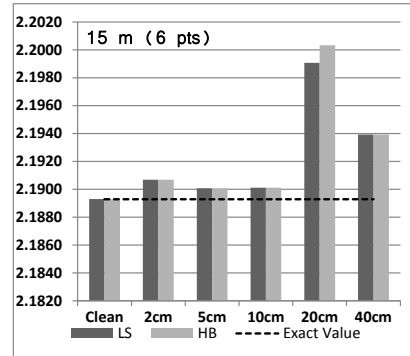
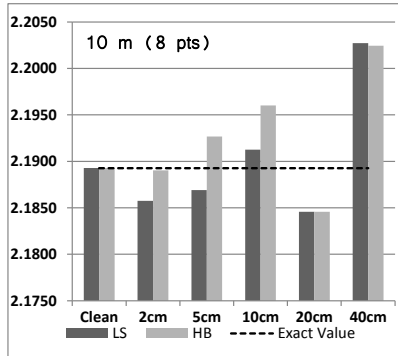
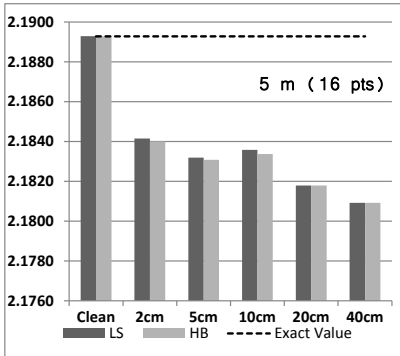
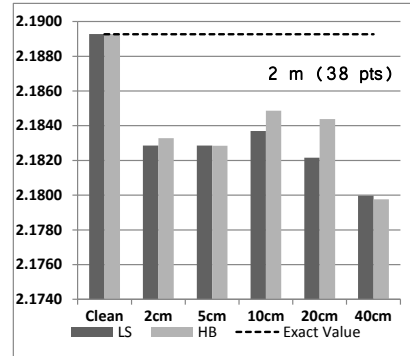
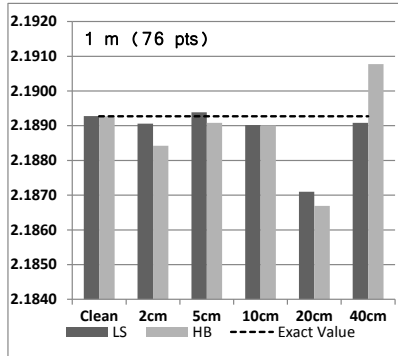
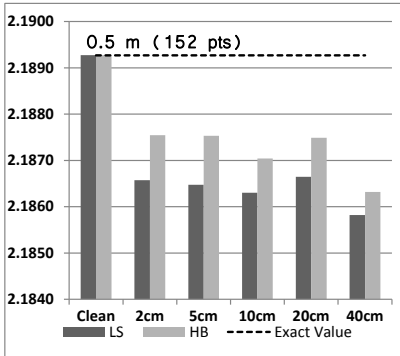
Circular Curve #9



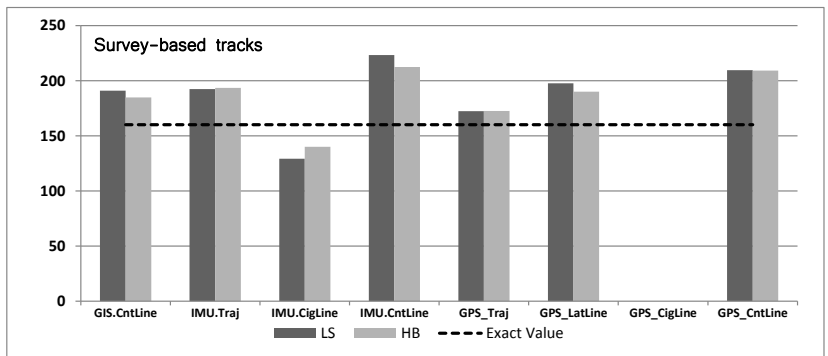
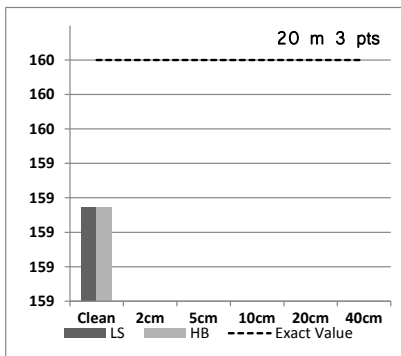
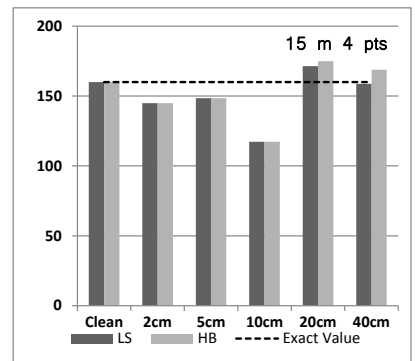
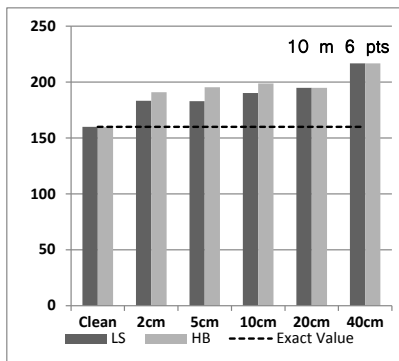
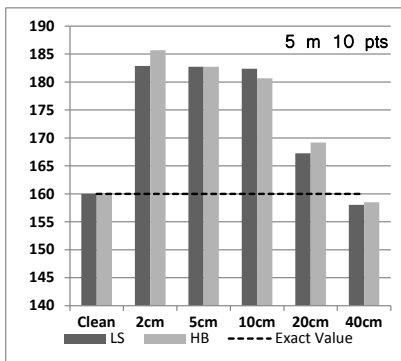
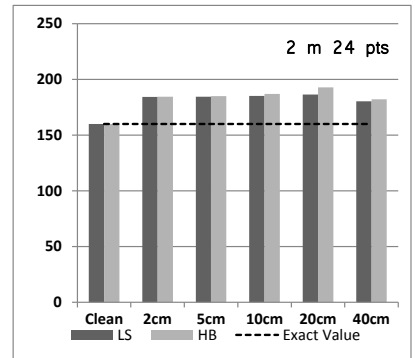
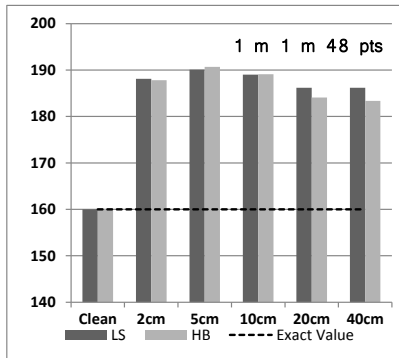
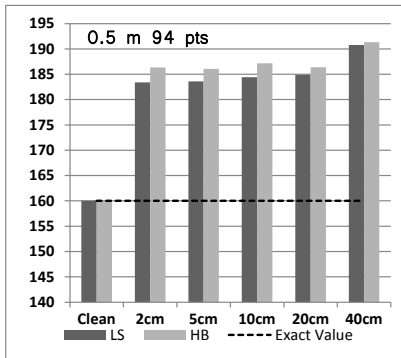
Circular Curve #9



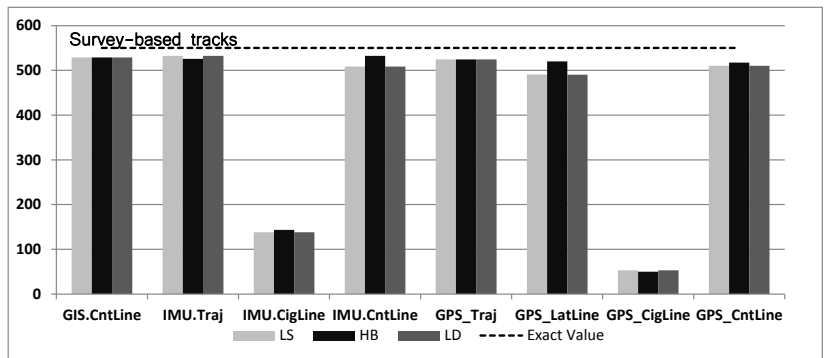
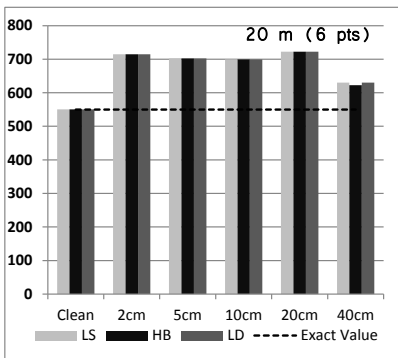
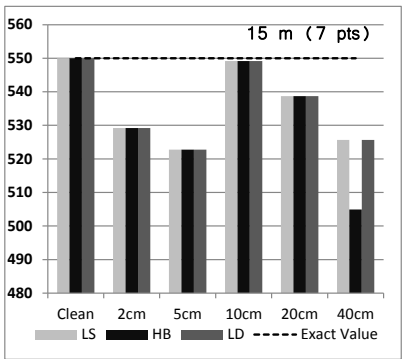
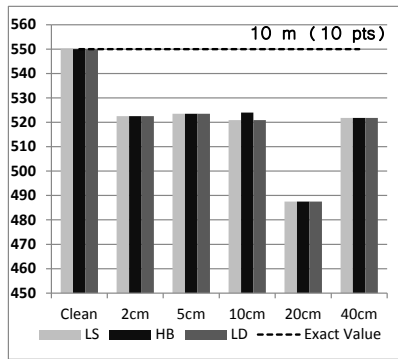
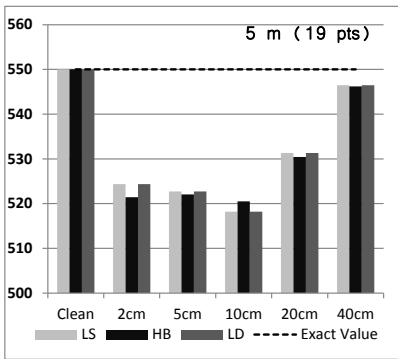
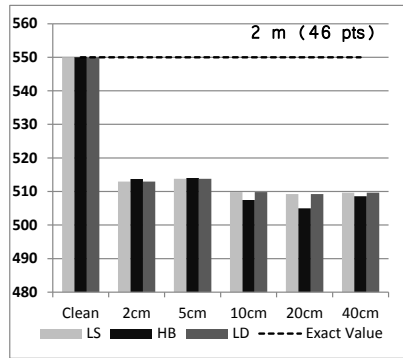
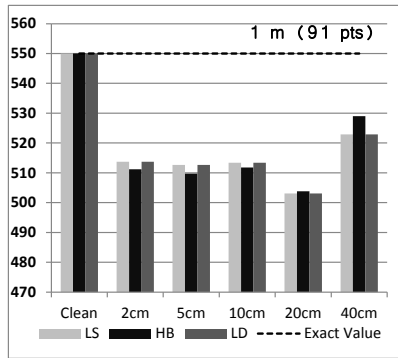
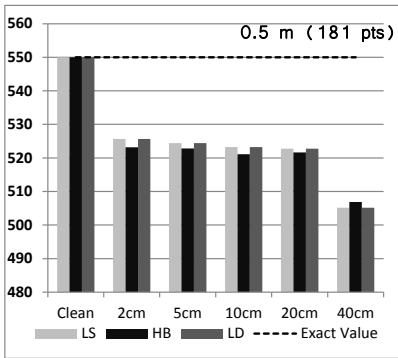
Clothoid #10



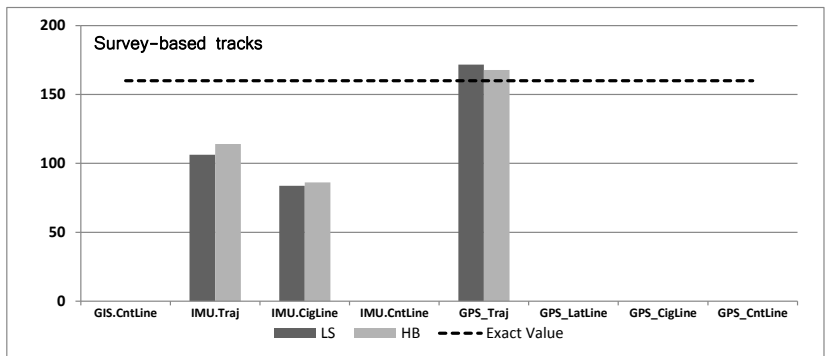
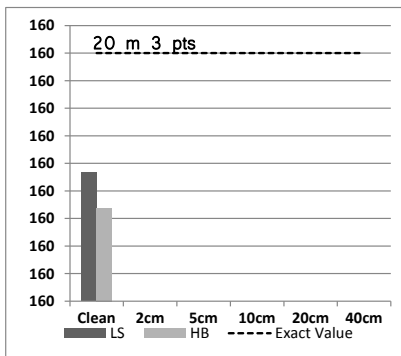
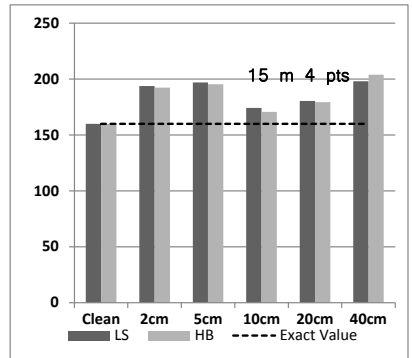
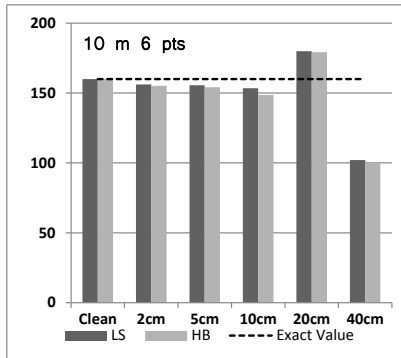
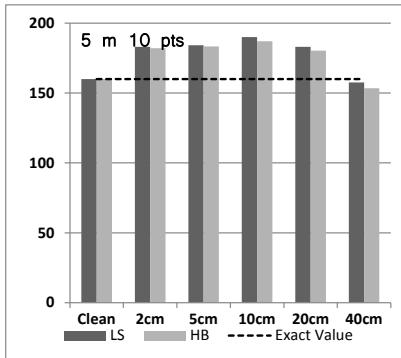
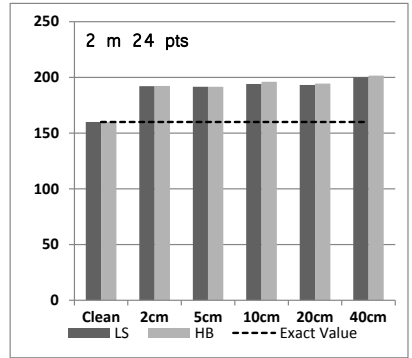
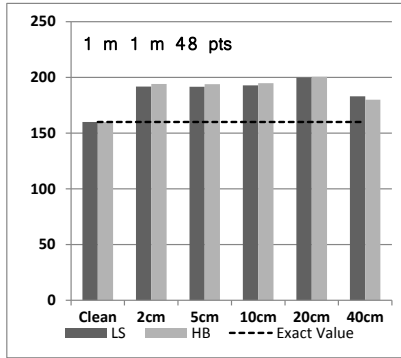
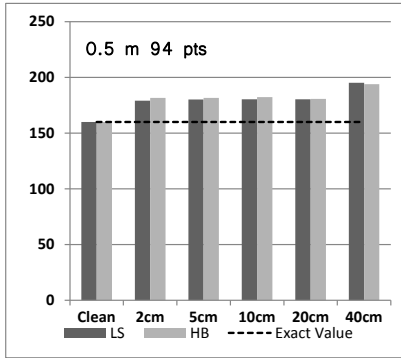
Tangent # 11



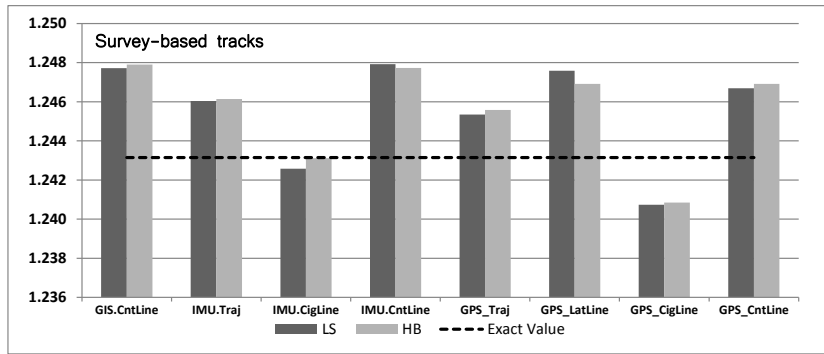
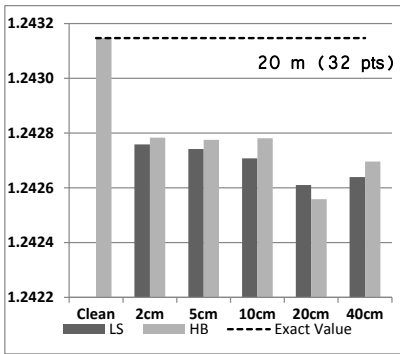
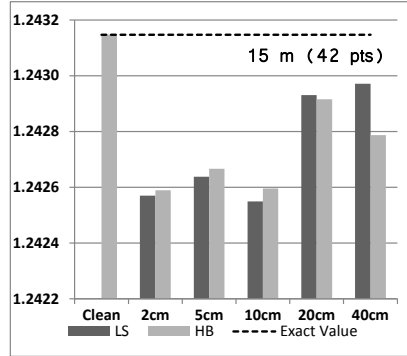
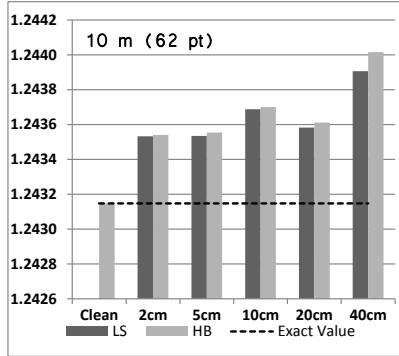
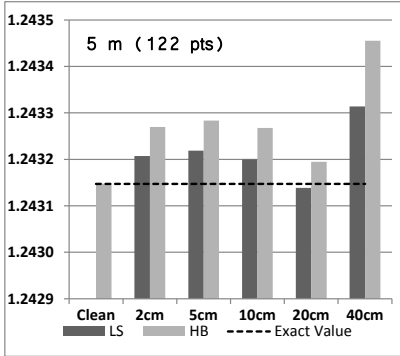
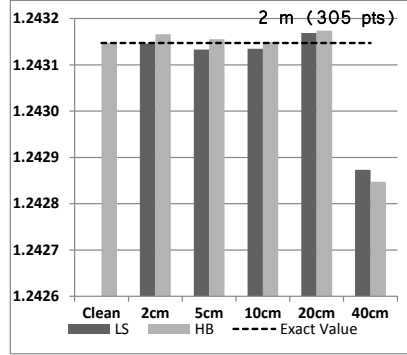
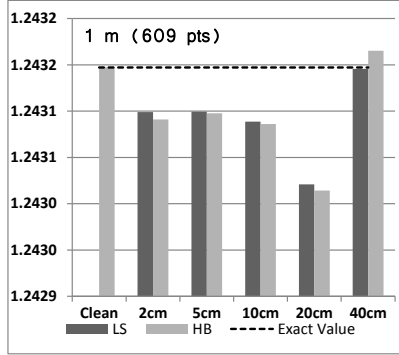
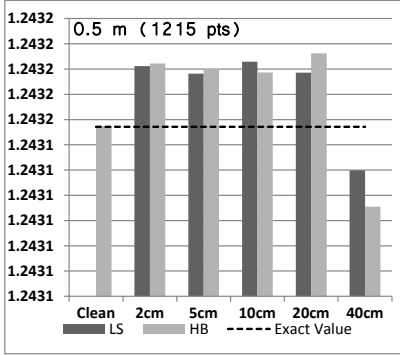
Clothoid #12



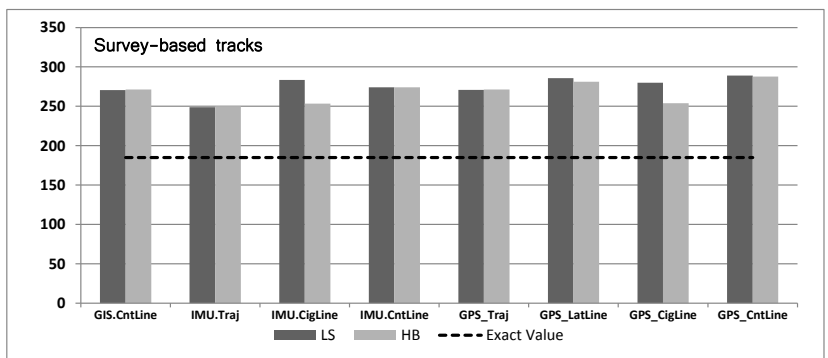
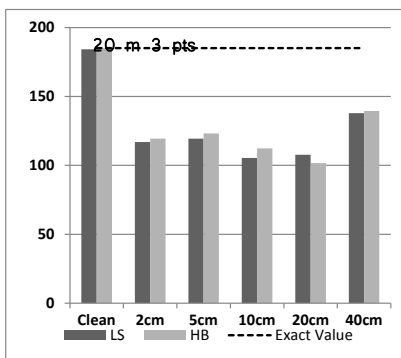
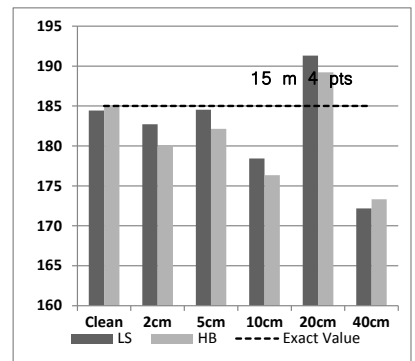
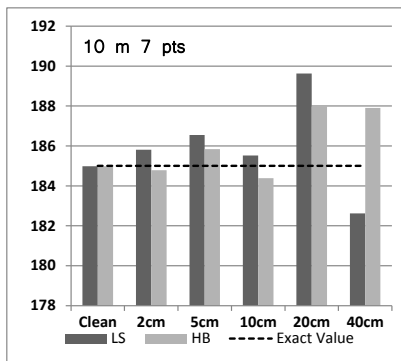
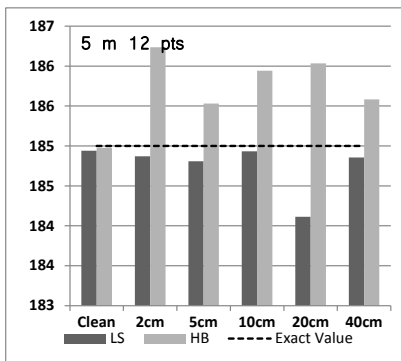
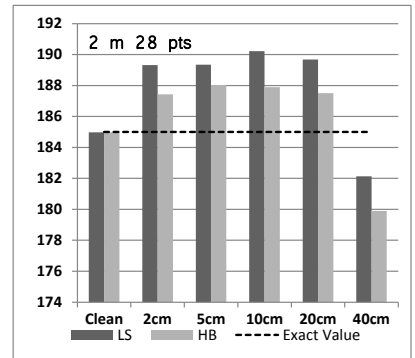
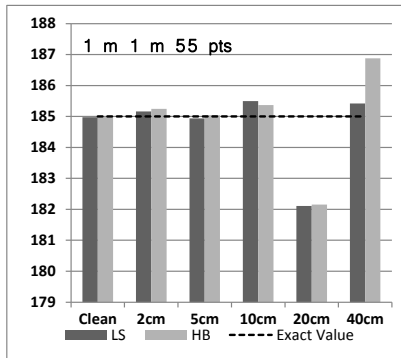
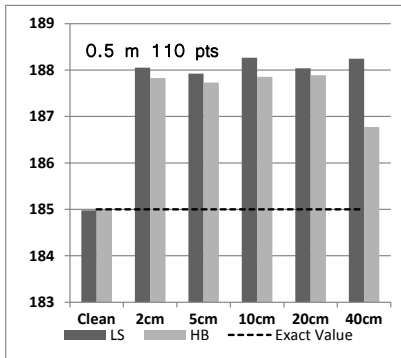
Circular curve #13



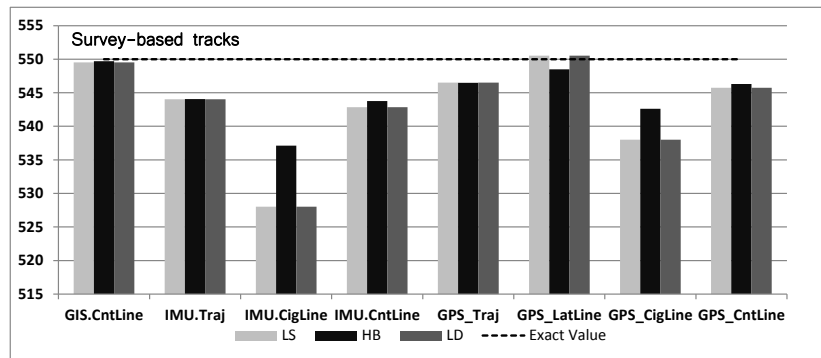
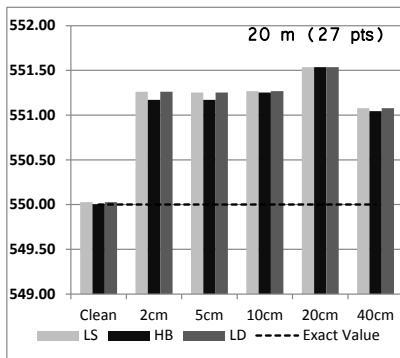
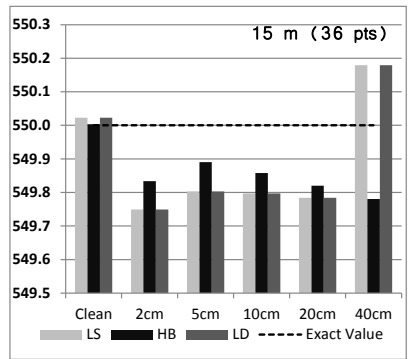
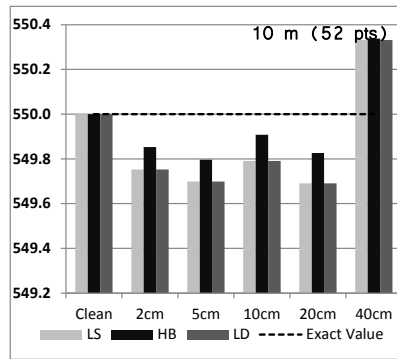
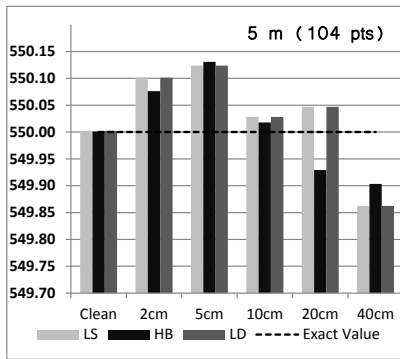
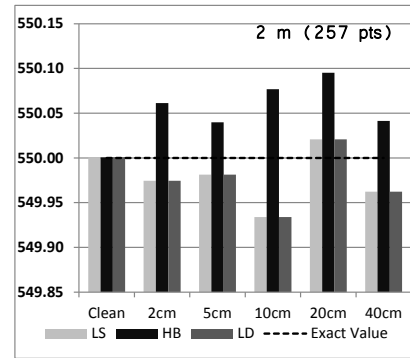
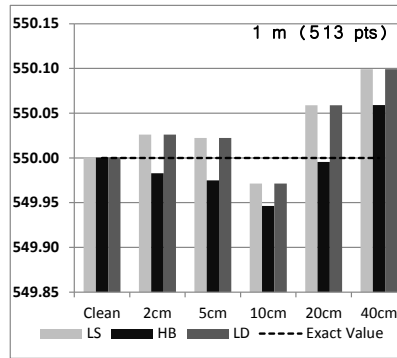
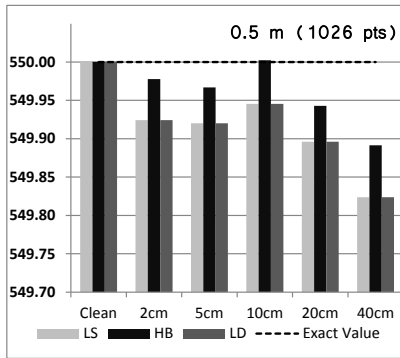
Clothoid #14



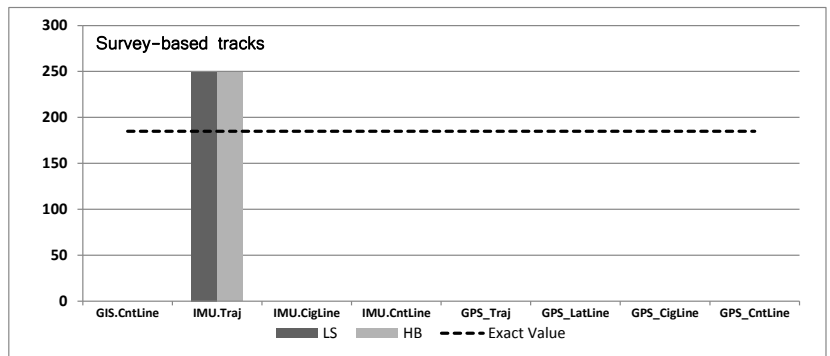
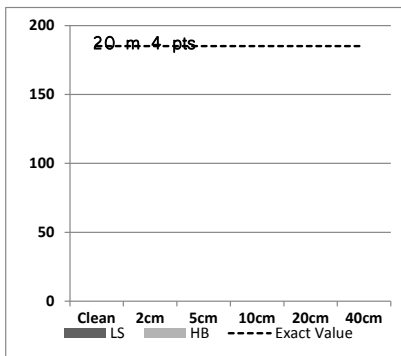
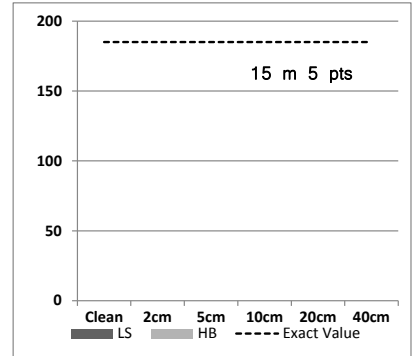
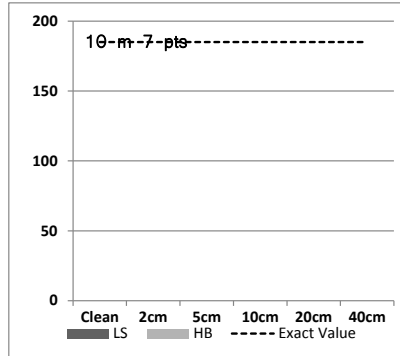
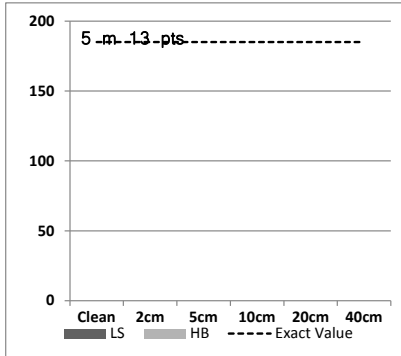
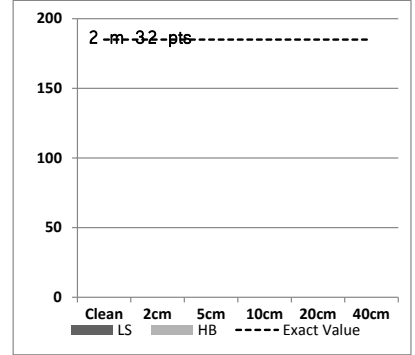
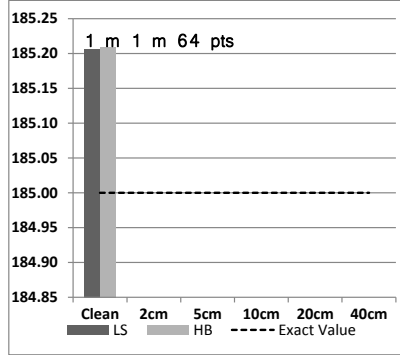
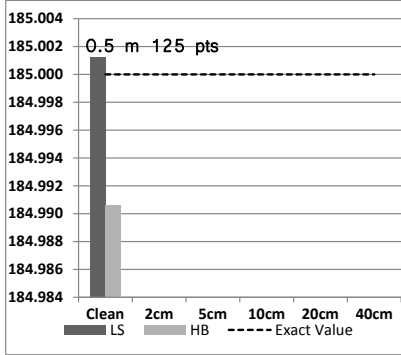
Tangent # 15



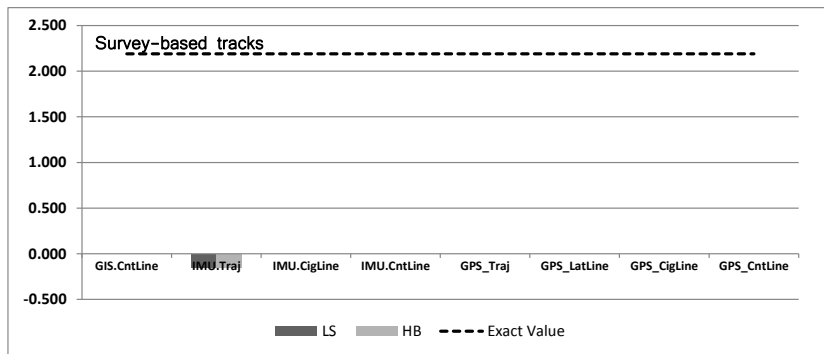
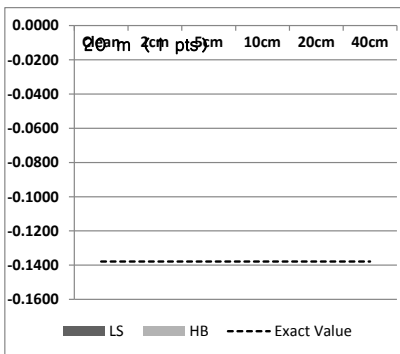
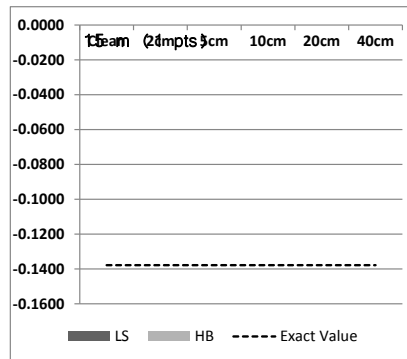
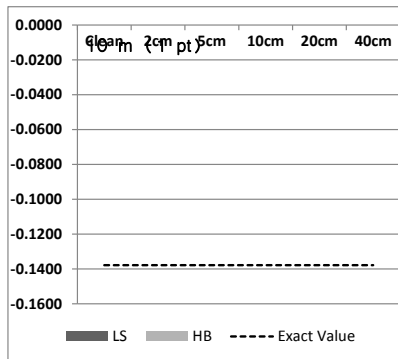
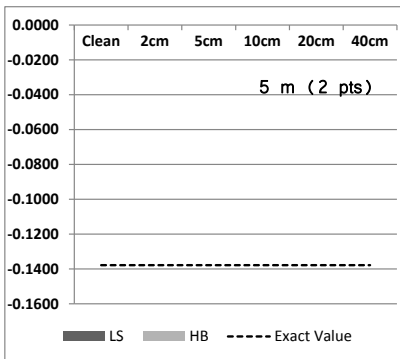
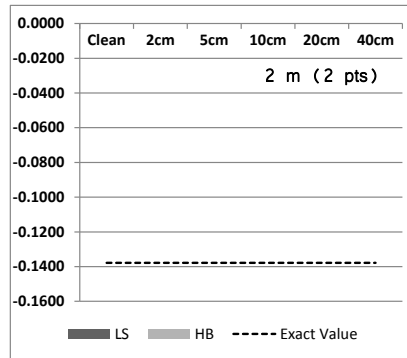
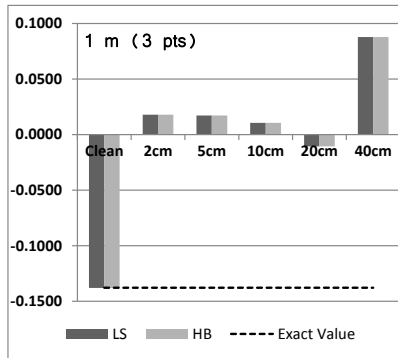
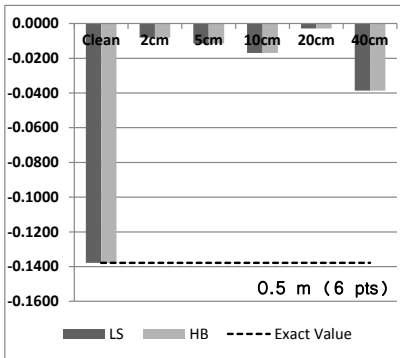
Clothoid #16



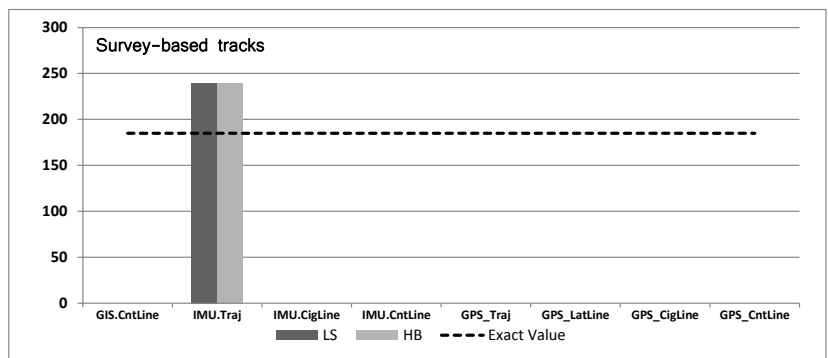
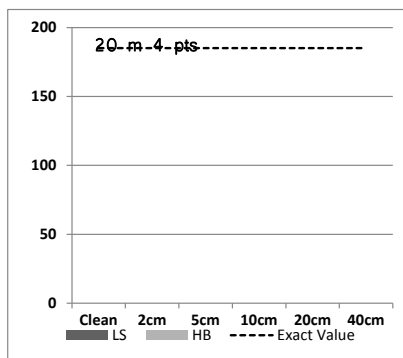
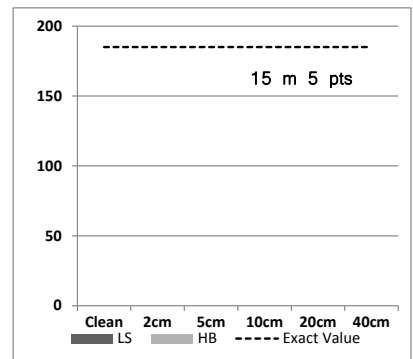
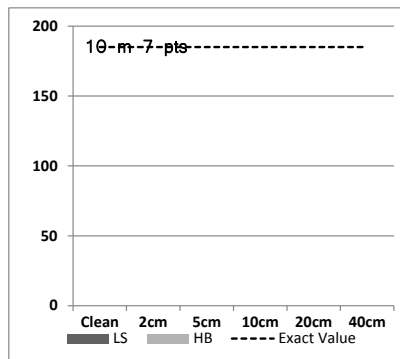
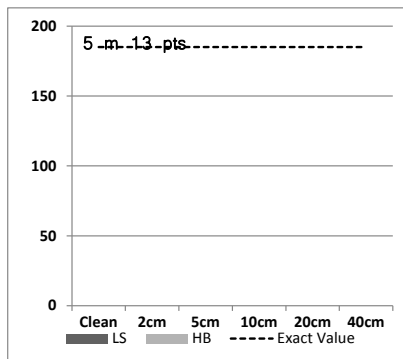
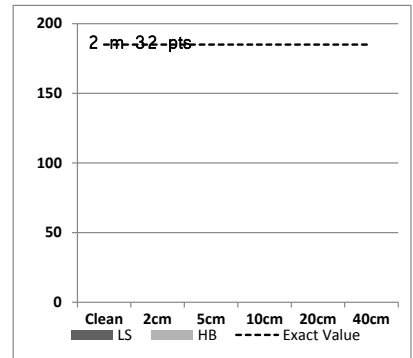
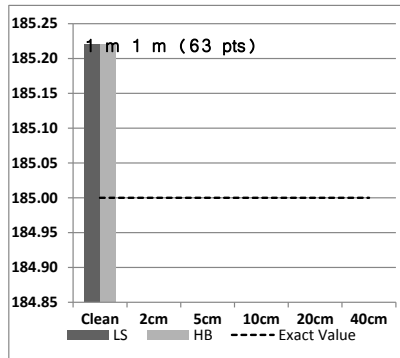
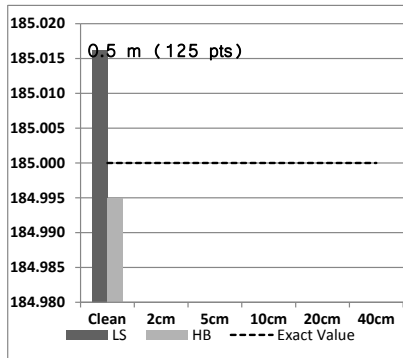
Circular curve # 17



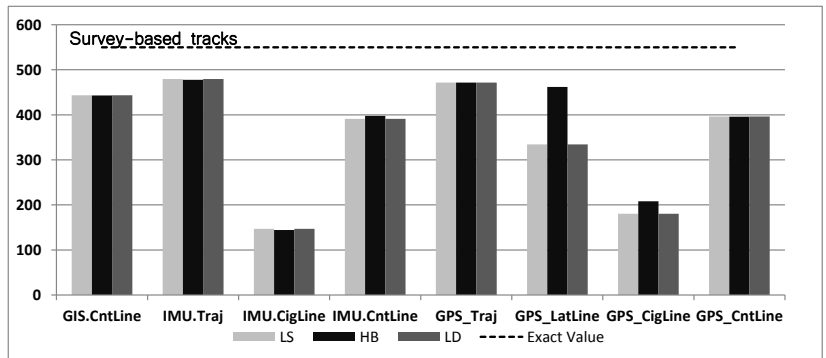
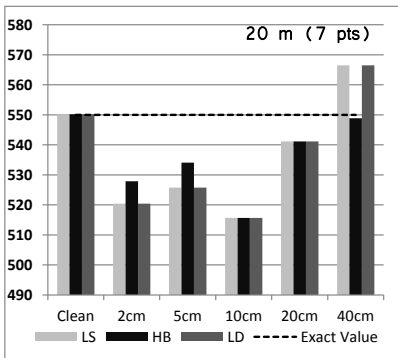
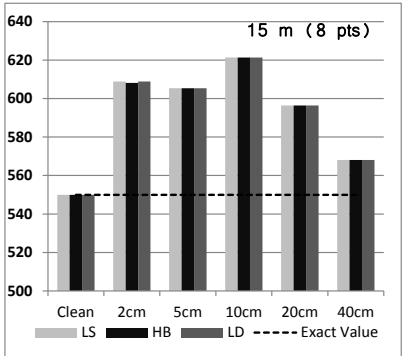
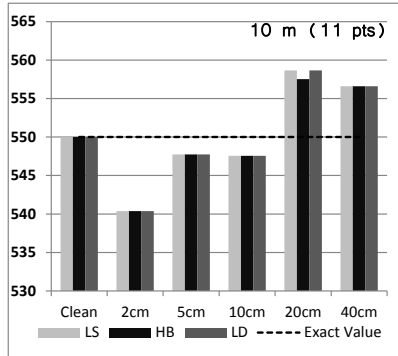
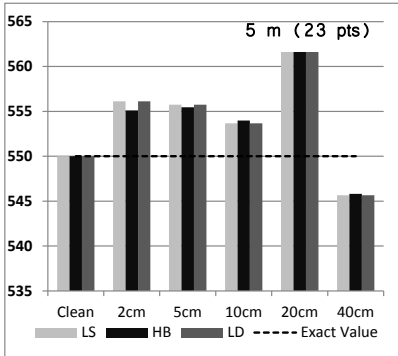
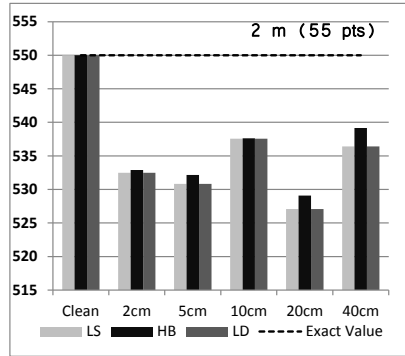
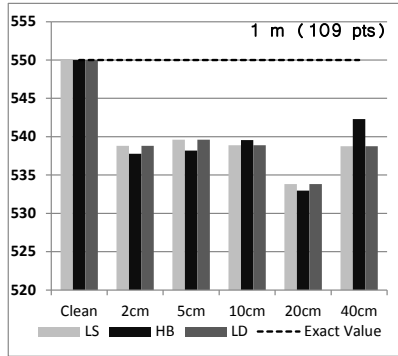
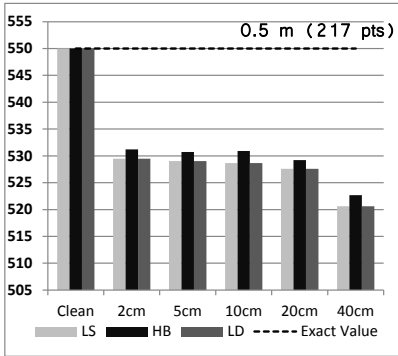
Clothoid #18



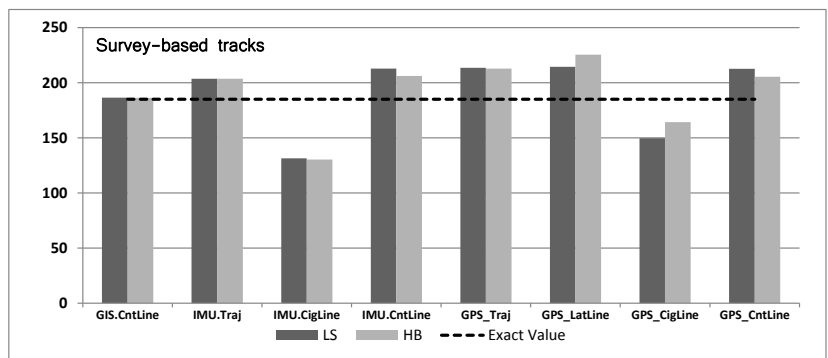
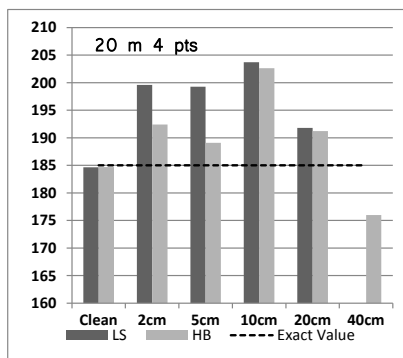
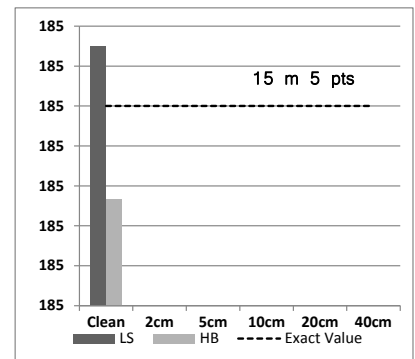
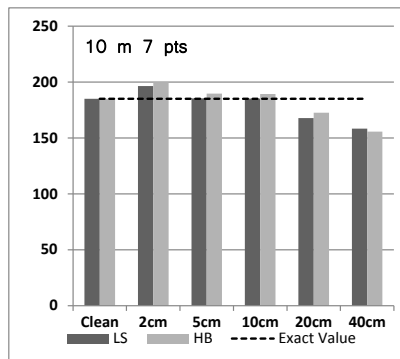
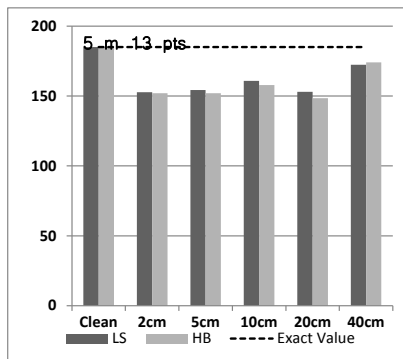
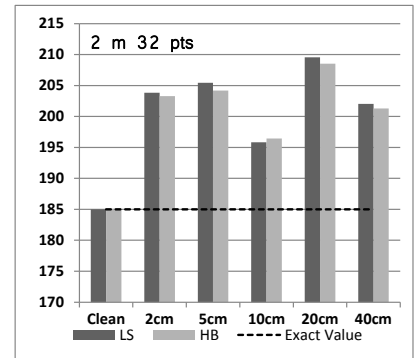
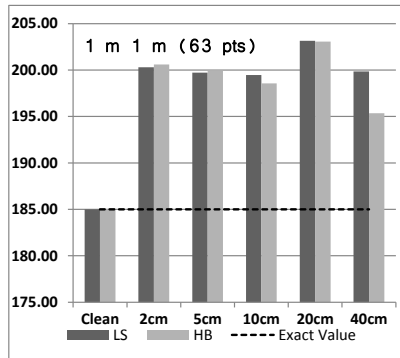
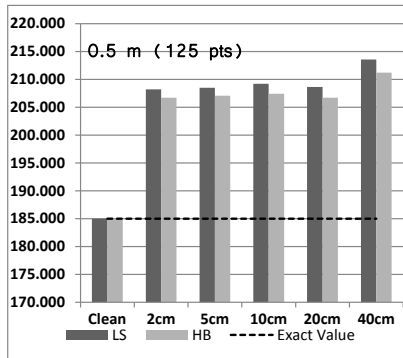
Tangent # 19



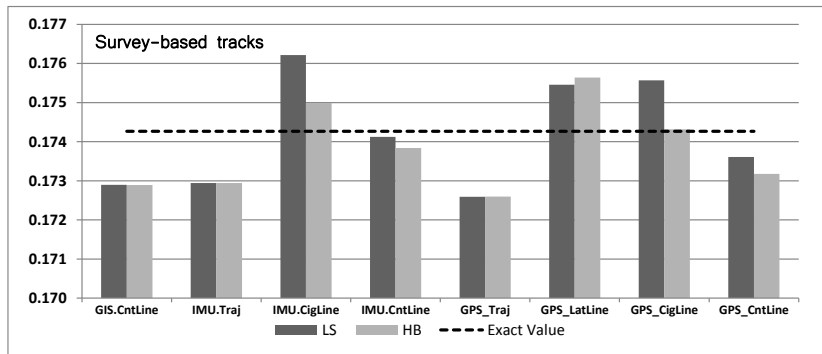
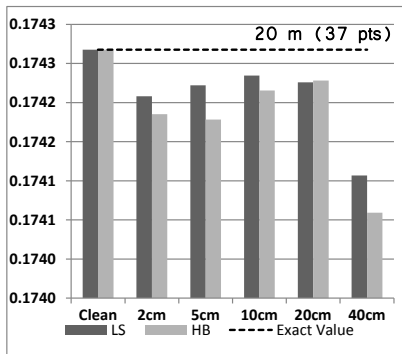
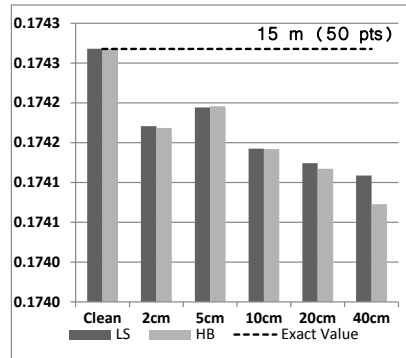
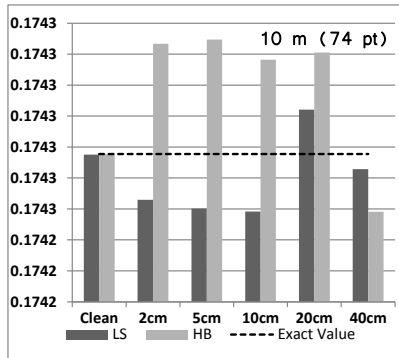
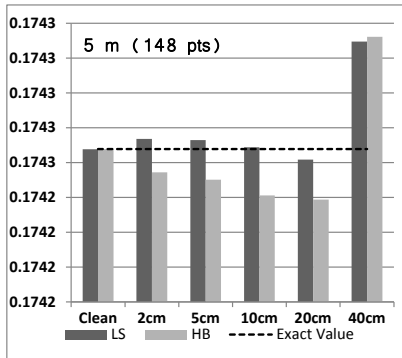
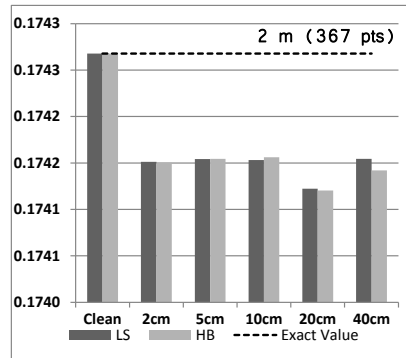
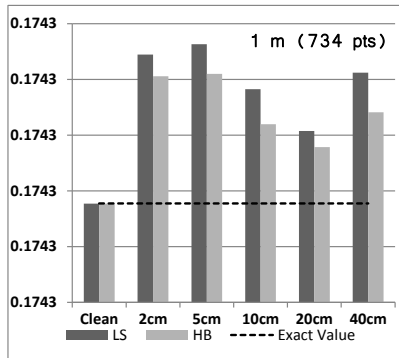
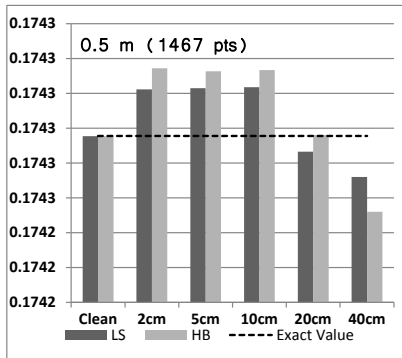
Clothoid #20



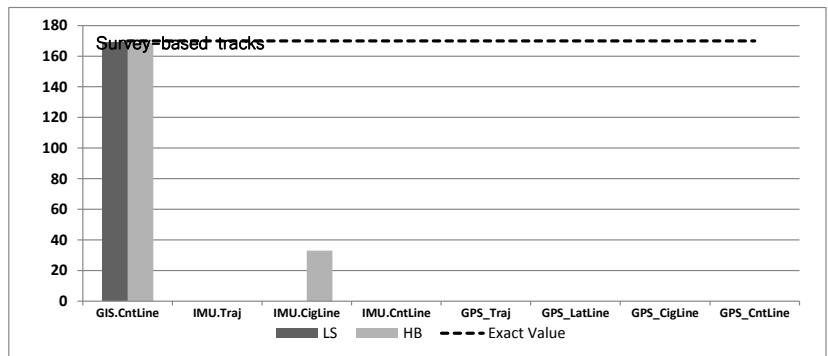
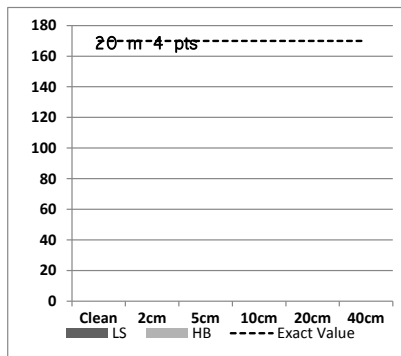
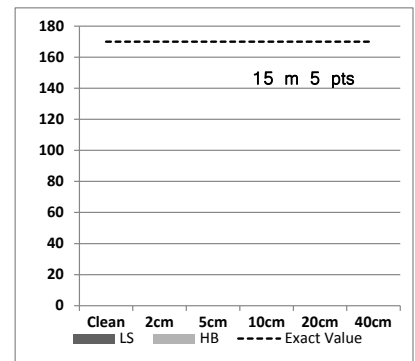
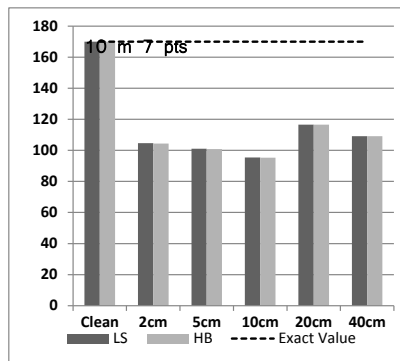
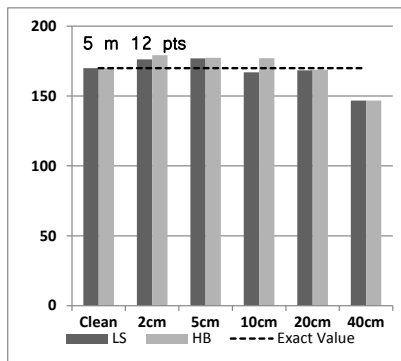
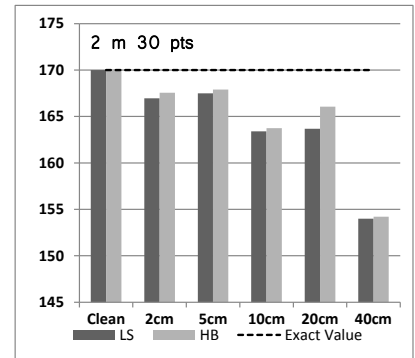
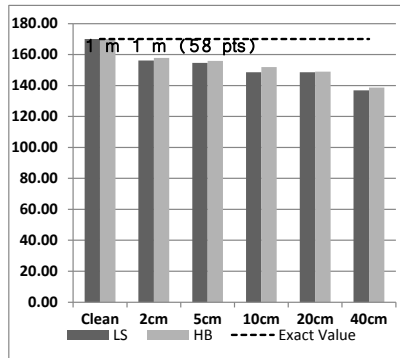
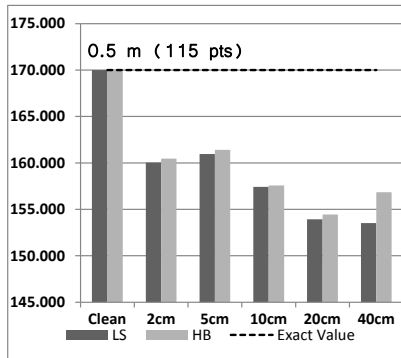
Circular curve #21



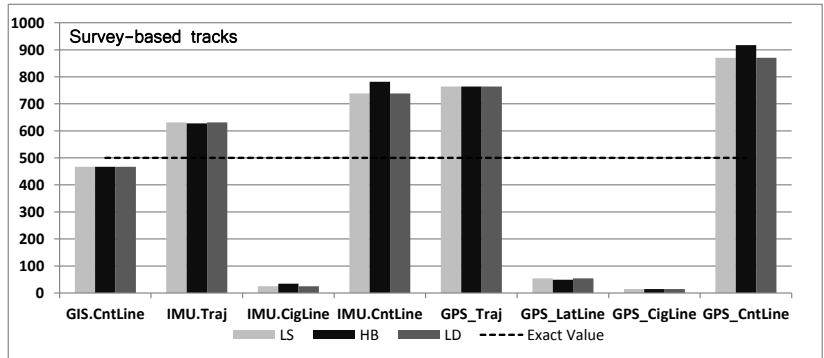
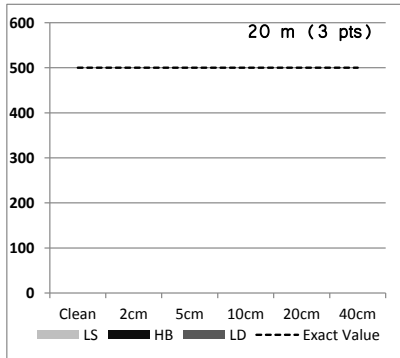
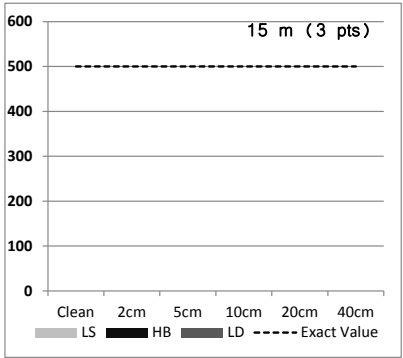
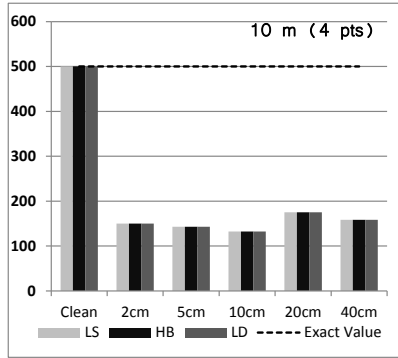
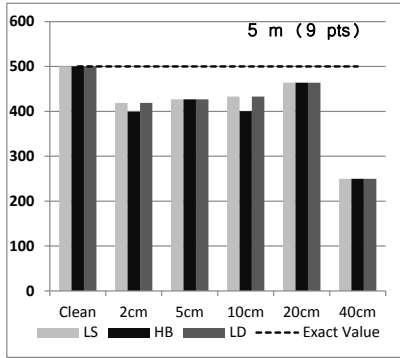
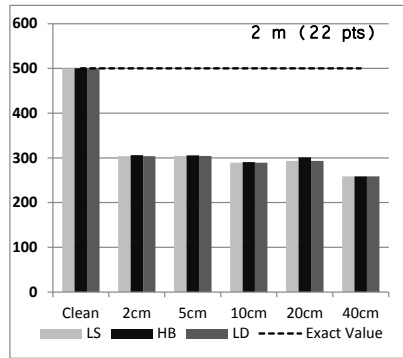
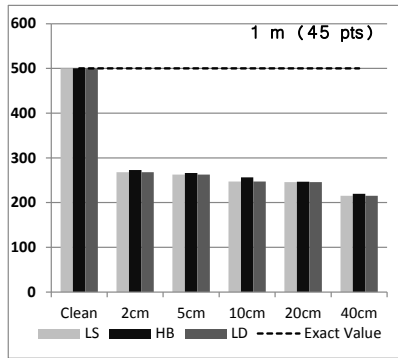
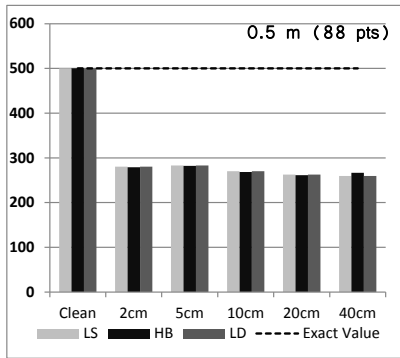
Clothoid #22



Tangent #23



Clothoid #24



Circular curve #25

R eferences

1. Abdel-Aty, M., & Essam Radwan, A. (2000). Modeling traffic accident occurrence and involvement. *Accident Analysis & Prevention* 35, 633-645.
2. Agostinacchio, M., Ciampa, D., & Oita, S. (2010). *Strade, Ferrovie e Aeroporti - La progettazione geometrica in sicurezza*. Roma: EPC Libri.
3. American Association of State Highway Transportation Officials. (2010). *Highway Safety Manual*.
4. American Society of Photogrammetry and Remote Sensing. (2004). *Manual of Photogrammetry*.
5. Andrews, D. F., Bickel, P. J., Hampel, F. R., Huber, P. J., Rogers, W. H., & Tukey, J. W. (1972). *Robust Estimates of Location. Survey and Advances*. Princeton, N.J.: Princeton University Press.
6. Anil, P., & Natarajan, S. (2010). Automatic Road Extraction from High Resolution Imagery Based On Statistical Region Merging and Skeletonization. *International Journal of Engineering Science and Technology*, 165-171.
7. Bacher, U., & Helmut, M. (2005). Automatic road extraction from multispectral high resolution. *ISPRS Workshop CMRT 2005*, (p. 29-34). Vienna.
8. Barzohar, M., & Cooper, D. (1996). Automatic finding of main roads in aerial images by using geometric-stochastic models and estimation. *IEEE transactions on pattern analysis and machine intelligence* 18, 707-721.
9. Bassani, M., Dalmazzo, D., & Marinelli, G. (2013). Variables Influencing Speed Distribution On Urban Arterials And Collectors. *TRB 92nd Annual Meeting Compendium of Papers*. Washington DC, US.
10. Bassani, M., Dalmazzo, D., Marinelli, G., & Cirillo, C. (2014). The effects of road geometrics and traffic regulations on driver-preferred speeds in northern Italy. An exploratory analysis. *Transportation Research Part F: Traffic Psychology and Behaviour*, 10-26.
11. Bassani, M., Lingua, A. M., Piras, M., De Agostino, M., Marinelli, G., & Petrini, G. (2012). Alignment Data Collection of Highways using Mobile Mapping and Image Analysis Techniques. *TRB 91st Annual Meeting Compendium of Papers DVD*. Washington DC, US: Transportation Research Board.

12. Beaton, A. E., & Tukey, J. W. (1974). The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data. *Technometrics* 16(2), 147-185.
13. BhattaCharya, U., & Parui, K. (1997). An improved backpropagation neural network for detection of road-like features in satellite imagery. *International journal of remote sensing* 18, 3379-3394.
14. Caliendo, C., Guida, M., & Parisi, A. (2007). A crash-prediction model for multilane roads. *Accident Analysis & Prevention*, 657-670.
15. Casella, V., Galetto, R., Surace, L., Ferretti, L., Bianchini, G., & Cavalli, A. (2001). Esperienze di fotogrammetria supportata da GPS/INS. *Bollettino SIFET, vol 4*, 35-49.
16. Chang, L.-Y., & Chen, W.-C. (2005). Data mining of tree-based models to analyze freeway accident frequency. *Journal of Safety Research* 36, 365-375.
17. Charalambros, P. (2014). Tensor-Cuts: A simultaneous multi-type feature extractor and classifier and its application to road extraction from satellite images. *ISPRS Journal of Photogrammetry and Remote Sensing* 95, 93-108.
18. Chaudhuri, D., Kushwaha, N. K., & Samal, A. (2012). Semi-automated road detection from high resolution satellite images by directional morphological enhancement and segmentation techniques. *Journal of selected topics in applied earth observations and remote sensing* 5, 1538-1544.
19. Cina, A. (2007). *Trattamento delle misure topografiche*. Torino: Celid.
20. Crisman, B., & Robba, A. (2004). Safety Evaluation: Practical Use Of Collected-Data Vehicle To Obtain Geometric Information Of Existing Roadways. *II S.I.I.V. International Congress*. Florence: New Technologies and Modeling Tools for Road applications to design and management.
21. Dal Poz, A., Gallis, R., Da Silva, J., & Martins, E. (2012). Object-space road extraction in rural areas Using stereoscopic aerial images. *IEEE geoscience and remote sensing letters* 9, 654-658.
22. Dong, H., Easa, S., & Li, J. (2007). Approximate Extraction of Spiralled Horizontal Curves. *Journal of Surveying Engineering*.
23. Easa, S., Dong, H., & Li, J. (2007). Use of satellite imagery for establishing road horizontal alignments. *Journal of Surveying Engineering*, 29-35.
24. European Parliament and Commission. (2008). *EU Guideline 2008/96/CE*.
25. Ewald, G. (1971). *Geometry: an introduction*. Belmont Wadsworth Publishing.
26. Farrell, J. A., & Barth, M. (1998). *The Global Positioning System & Inertial Navigation*. Mc Graw Hill.
27. Forsythe, A. B. (1972). Robust Estimation of Straight Line Regression Coefficients by Minimizing pth Power Deviations. *Technometrics* 14(1), 159-166.
28. Greibe, P. (2003). Accident prediction models for urban roads. *Accident Analysis and Prevention* 35, 273-285.
29. Hadi, M. H., Aruldas, J., Chow, L.-F., & Wattleworth, J. (1993). Estimating safety effects of cross-section design for various highway types using negative binomial regression. *Transportation Research Record* 1500, 169-177.
30. Huber, P. J. (1964). Robust Estimation of a Location Parameter. *The annals of mathematical statistics* 35(1), 73-101.
31. Huber, P., & Ronchetti, E. (2009). *Robust Statistics*. John Wiley & Sons.

32. Imran, M., Hassan, Y., & Patterson, D. (2006). GPS-GIS-Based Procedure for Tracking Vehicle Path. *Computer-Aided Civil and Infrastructure Engineering* 21, 383-394.
33. Inghilleri, G. (1974). *Topografia generale*. Torino: UTET.
34. Klang, D. (1998). Automatic detection of changes in road data bases using satellite imagery. *ISPRS Commission IV Symposium on GIS*, (p. 293-298). Stuttgart.
35. Landau, U. (1987). Estimation of a circular arc center and its radius. *Computer Vision, 490 Graphics and Image Processing* 39, 317-326.
36. Li, Z., Chitturi, M. V., Bill, A. R., & Noyce, D. A. (2012). Automated identification and extraction of horizontal curve information from geographic information system roadway maps. *Transportation Research Record* 2291, 80-92.
37. Lingua, A., Balbo, S., Giorgi, F., & Piras, M. (2009). GEOWASP: Software di acquisizione e trattamento dati rilevati da un sistema di rilevamento dinamico a basso costo. *Atti del 13° Convegno ASITA*, (p. 217-222). Bari.
38. Manzano, A. M. (2003). Introduzione all'uso del filtro di Kalman in Geodesia. *Bollettino della società italiana di fotogrammetria e topografia*, p. Vol.2, 63-100.
39. Marinelli, G. (2011). *Indagine sperimentale sull'impiego delle tecniche di Mobile Mapping per la determinazione delle caratteristiche geometriche e cinematiche della traiettoria di un veicolo*. Torino.
40. Mathworks. (2011). *MATLAB User Manual*.
41. McCall, J. C., & Trivedi, M. M. (2004). An integrated, robust approach to lane marking detection and lane tracking. *Intelligent Vehicles Symposium*, (p. 533-537). Parma.
42. Mena, J., & Malpica, J. (2005). An automatic method for road extraction in rural and semi-urban areas starting from high resolution satellite imagery. *Pattern Recognition Letters*, 1201-1220.
43. Milton, J., & Mannering, F. (1998). The relationship among highway geometrics, traffic-related elements and motor-vehicle accident frequencies. *Transportation* 25, 395-413.
44. Ministero delle Infrastrutture e dei Trasporti. (1993). *Nuovo codice della strada*.
45. Ministero delle Infrastrutture e dei Trasporti. (2001). *Norme Funzionali e Geometriche per la Costruzione delle Strade*. Rome.
46. Ministero delle Infrastrutture e dei Trasporti. (2012). *Linee Guida per la gestione della sicurezza delle infrastrutture stradali*.
47. Özkaya, M. (2012). Road Extraction from high resolution satellite images. *XXII ISPRS Congress*. Melbourne.
48. Ramsay, J. O. (1977). A Comparative Study of Several Robust Estimates of Slope, Intercept, and Scale in Linear Regression. *Journal of the American Statistical Association* 72(359), 608-615.
49. Rasdorf William, Findley, D. J., Zeeger, C. V., Sundstrom, C. A., & Hummer, J. E. (2012). Evaluation of GIS Applications for Horizontal Curve Data Collection. *Journal of Computing in Civil Engineering* 26(2), 191-203.
50. Shankar, V., Mannering, F., & Barfield, W. (1995). Effect of roadway geometrics and environmental factors on rural freeway accident frequencies. *Accident Analysis and Prevention*, 371-389.
51. Thomas, S., & Chan, Y. (1989). A Simple Approach for the Estimation of Circular Arc Center and 492 Its Radius. *Computer Vision, Graphics and Image Processing*, 362-370.

52. Tiberius, C. (1998). *PHD Thesis "Recursive data processing for kinematic GPS surveying"*. Delft, University of Technology.
53. Tournaire, O., & Papanoditis, N. (2009). A geometric stochastic approach based on marked point processes for road mark detection from high resolution aerial images. *Journal of Photogrammetry and Remote Sensing* 64, 621-631.
54. US Dept. - Federal Highway Administration. (1987). *Safety effects of cross-section design for two-lane roads - Final Report*.
55. Wang, C., Zhencheng, H., & Uchimura, K. (2008). Precise curvature estimation by cooperating with digital road map. *Intelligent Vehicles Symposium, 2008 IEEE* (p. 859-864). Eindhoven: IEEE.
56. Wegner, J., Montoya-Zegarra, J., & Schindler, K. (2013). A higher-order CRF model for road network extraction. *IEEE Conference on Computer Vision and Pattern Recognition*, (p. 1698-1705). Zurich.
57. World Health Organization. (2013). *Global status report on road safety*.
58. Zhao, H., Kumagai, J., Nakagawa, M., & Shibasaki, R. (2012). Semi-automatic road extraction from high resolution satellite image. *International society for photogrammetry and remote sensing archive*.

Acknowledgements

Al termine di questo lavoro desidero ringraziare le persone che mi hanno affiancato e supportato durante questo percorso di 3 anni che mi ha dato occasione di crescere molto, sia personalmente che professionalmente.

Il primo ringraziamento va al **professor Marco Bassani**, per avermi offerto la possibilità di compiere questa lunga attività di ricerca insieme, per aver condiviso con me le sue conoscenze, le sue esperienze, per avermi permesso di entrare in un mondo intrigante e dinamico, per avermi dato la possibilità di vivere avventure (italiche e estere!) che non avrei avuto modo di conoscere altrimenti.

Un grazie anche per tutto il tempo impiegato a insegnarmi i primi passi in questo mondo e per la voglia di continuare a lavorare insieme anche in futuro.

Un altro “grazie” appartiene a tutto il gruppo di ricerca di Strade, Ferrovie e Aeroporti del DIATI del Politecnico di Torino, i professori **Ezio Santagata** e **Orazio Baglieri** e tutto il fantastico gruppo dei “baracchinati e non” che, chi più chi meno, ha vissuto a fianco a me questi tre anni: **Pier, Beppe, Davide, Lucy, Mike, Roberto, Sergio, Ary, Angela, Mauro, Muhammad** e **Peppino!**

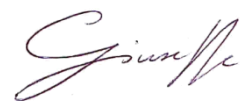
Un “grazie” speciale è rivolto a chi mi ha affiancato e aiutato durante la mia ricerca, anche se non direttamente coinvolto, e che mai si è tirato indietro nel darmi un suggerimento e un consiglio quando richiesto: i professori **Ambrogio Manzino, Alberto Cina, Andrea Lingua, Tamara Bellone** e il mitico **Marco Piras**.

Non posso non ringraziare tutti i miei amici, che mi hanno dato la possibilità di evadere dagli stress universitari! Un pensiero speciale va alla compagnia “TeatroMatto”, con tutte le sue recenti estensioni, che mi ha fatto vivere grandi emozioni e con le quali sono contento di poter dire di aver realizzato qualcosa! Grazie!

Un grazie che viene dal profondo del cuore va alla mia famiglia, che ha dato un gusto meraviglioso agli avvenimenti che in questi 3 anni non sono mancati... Grazie a **Mamma e Papà** che sono sempre una forza della natura e che mi sono stati vicino in ogni momento, senza mollare mai... Grazie a **Tommy e Vero**, che ci stanno per regalare una grandissima emozione! Non vedo l'ora che arrivi il pupo! Grazie anche a **Rino e Silvana**, per avermi accolto come un nuovo figlio e grazie a **Daniele e Roxana** per regalarci la seconda grande emozione! Arianna ti aspettiamo!

Per te, **Eleonora**, la parola “grazie” non basta per poterti dire quanto sei stata fondamentale per me in questi anni. Ad oggi, che scrivo queste parole, sono 1 anno e 4 mesi che sei mia moglie, a cui si aggiungono 3 anni di amore senza confini. Sei il mio Tutto, la mia Vita e ti invidio per la forza e il coraggio con cui hai accettato di affrontare con me questa prossima avventura che ci aspetta...

Se mai saprò prima o poi ricambiarti con lo stesso coraggio, allora sarà quello il momento in cui potrò forse dire di essere Grande quanto sei dal primo giorno in cui ti ho incontrata. Semplicemente, **Ti Amo**.



PS: ... infine, ringrazio anche tutti coloro ai quali non sono *mai* riuscito a rispondere alla domanda:

“Ma in pratica, 'sto Dottorato di Ricerca che cos'è?” 😊