



DM Actuator Slaving

Author: Justin Mansell
Revision: 6/12/11

1 Abstract

In this application note, we present evidence that slaving of edge actuators creates a reduction in DM performance when using a DM such that the edge actuators are well illuminated.

2 Introduction

One common technique for using a DM in a practical system is to command the edge actuators to their nearest neighbor or average of their nearest neighbors. In this application note we investigate when this kind of approach might be beneficial when compensating Kolmogorov-spectrum turbulence.

3 Approach

This study began by generating an actuator grid pattern and their corresponding influence functions based on clamped plate theory. For this note, we will present the results from the DM with 7 actuators across and 37 total actuators. We performed nearest neighbor slaving. Figure 3.1 shows the actuator grid and the slaving map.

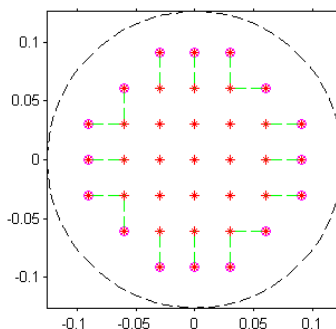


Figure 3.1- 37-Actuator Grid and Slaving Map

Figure 3.2 shows a select set of influence functions. The edge influence functions are not as localized as the central ones, but they do provide control out to the edge of the mirror surface. It is this effect that enables the DM to compensate aberrations out to the edge. We have measured edge influence functions on our plate-

type DMs and find that the influence functions generated with clamped plate theory matches well with the laboratory results.

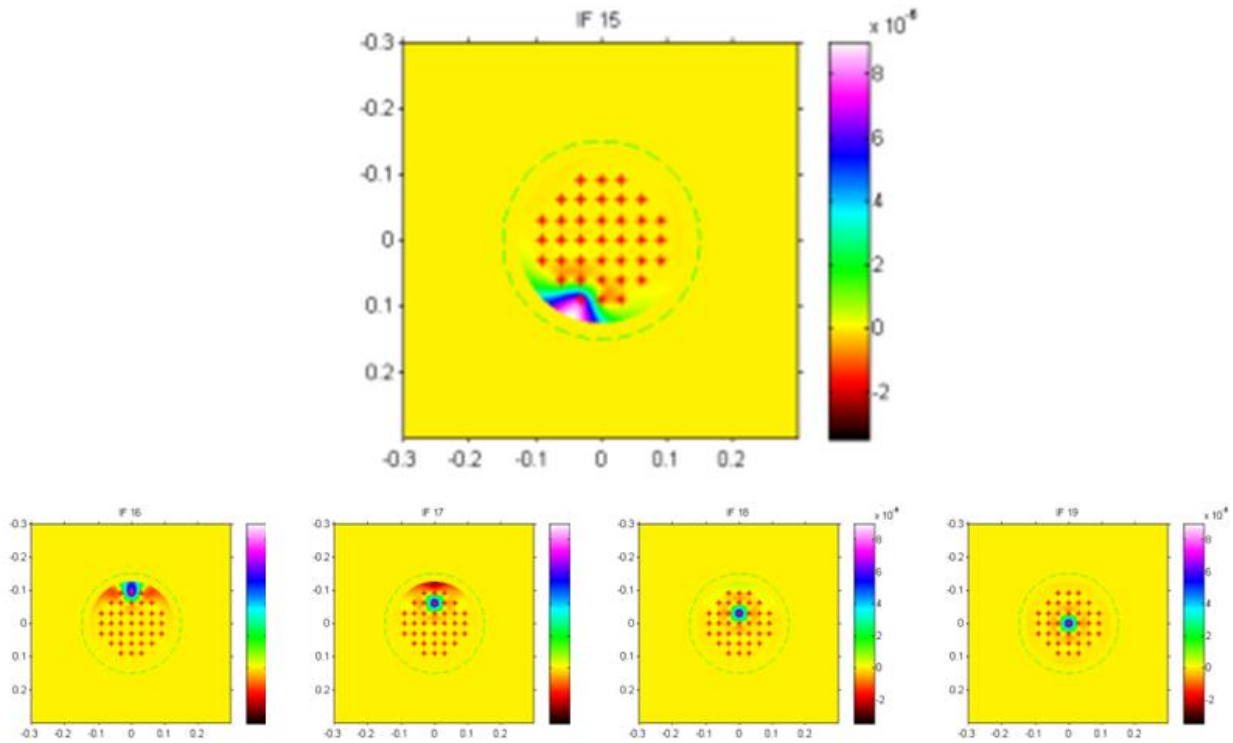


Figure 3.2 - Select DM Influence Functions

During this study we generated 20 independent frames of Kolmogorov-spectrum turbulence using the FFT of the PSD. The phase screen amplitude was scaled to achieve a D/r_0 of 2.0. This phase screen was scaled laterally as it would be with varying telescope magnification. Figure 3.3 shows three example scaled phase screens.

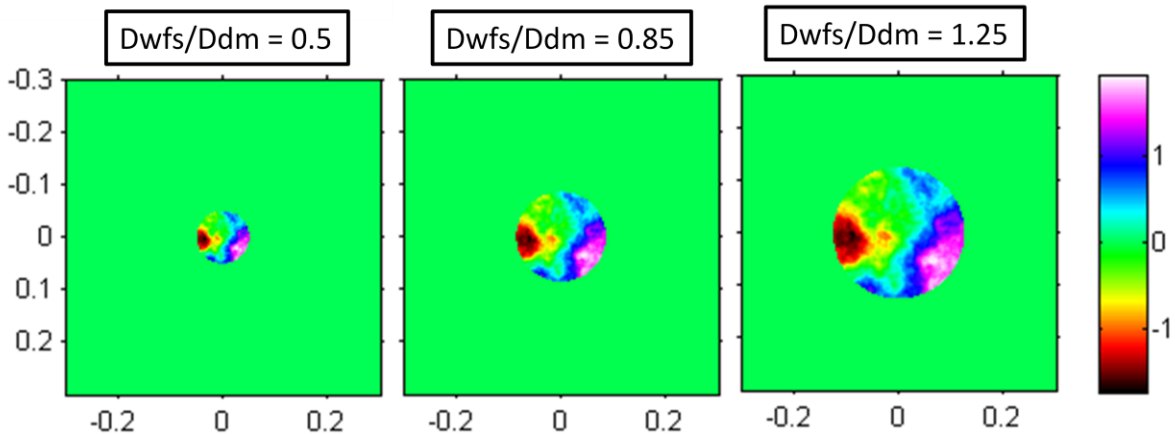
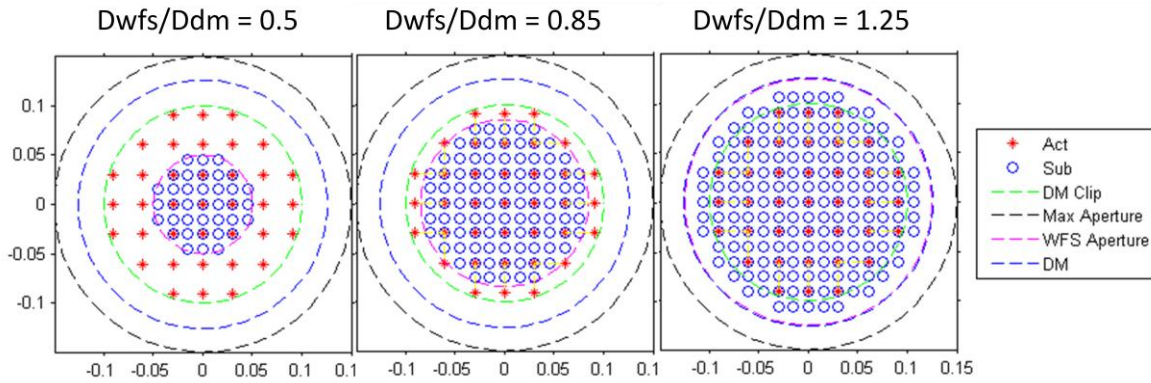


Figure 3.3 - Scaled Kolmogorov phase screens

We then generated a Hartmann wavefront sensor sub-aperture grid covering the phase screen aperture with a fixed spacing of half the actuator spacing. This over-resolved grid reduced the effects of a limited resolution wavefront sensor as can be seen with a Fried geometry.



We generated a poke matrix using an idealized wavefront sensor routine and a fitting (control) matrix by inverting the poke matrix. During the inversion, any SVD gains that were less than 1% of the maximum gain were eliminated. We then fit the Kolmogorov phase screen and evaluated the residual of the fit for each of the 20 phase screens while scanning the effective telescope magnification (phase screen lateral scaling). This fit was done with both slaving and without slaving.

Figure 3.4 shows the results of this modeling. When looking at this chart, it is important to remember that there are 7 actuators across the DM diameter. When the beam was between 4 and 5 actuator spacing in diameter the slaving and non-slaving results were very similar. After that, the slaving results were always worse than the non-slaved results. The best system performance was at a diameter between 7 and 8.

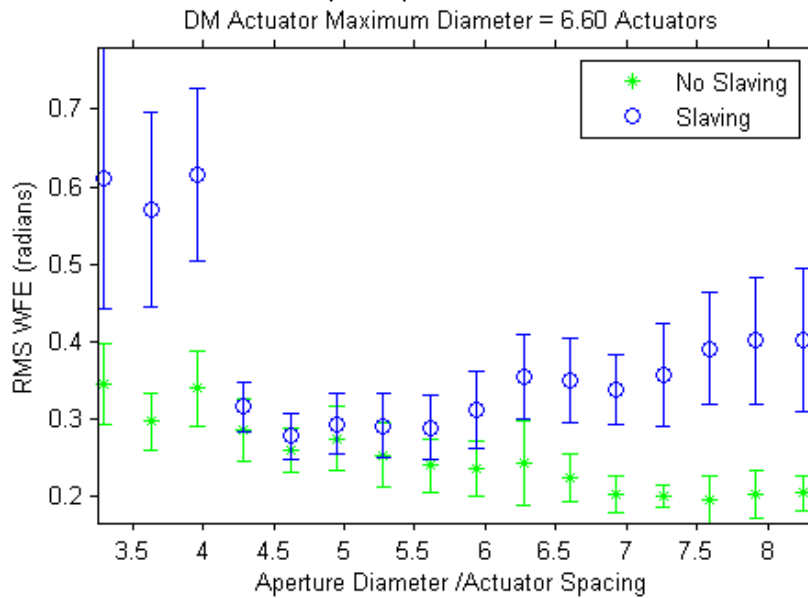


Figure 3.4 - Results of varying the scaling for slaved and non-slaved operation

4 Conclusions

We repeated this simulation with different numbers of actuators, but always found that the best results were without slaving and by illuminating a diameter just larger than the actuator spacing times the number of actuators across the aperture.

5 Appendix: Simulation Code

```

% slaving vs no slaving
setup;
ppt=1;

Dap = 30e-2;
dact = 6e-3;
nxy = 512;
Nframes=20;
for Nact1D = 7;
    if (ppt)
        TitleSlidePowerPoint(sprintf('Nact1D = %i',Nact1D));
    end;
    Mag = Dap / (Nact1D*dact*sqrt(2));
    r0 = 3e-3 * Mag;
    dxy = Dap*2.0/nxy;
    rBound = Dap*2;
    dsub = dact/2;
    Dclip = Dap/1.5;
    DclipWFS = Dclip;

    xxx=whos;
    for ii=1:length(xxx)
        t{ii} = sprintf('%s = %.2f',xxx(ii).name,eval(xxx(ii).name));
    end;
    if (ppt)
        TextToPowerPoint(t, 'Variables',16,15);
    end;

%% setup grids and plot
gDM=MakeGridWFS(Nact1D,Nact1D,dact*Mag,dact*Mag,Dclip);
xact = gDM.xsub; yact = gDM.ysub;
ract = sqrt(xact.^2 + yact.^2);
rDM = dact*Mag + max(ract);
apDM = aperture(nxy,dxy,rDM,0,0,0,0);
g = makeGrid(nxy,dxy);

%% generate IFs
s.nxy = nxy;
s.a = rBound;
s.dxy = dxy;
s.xact = xact; s.yact = yact;
[IF, info] = MakePlatedDMIFs2(s);

%% apply aperture to the DM
for ii=1:length(IF)
    IF{ii}=IF{ii}.*apDM;
end;

%% plot the IFs
nf;
for ii=1:length(IF);
    show(g.x,g.y,IF{ii}); title(sprintf('IF %i',ii));
end;

```

```

hold on; plot(xact,yact,'r*');
plotCircle(Dap,0,0,'g--');
drawnow;
if (ppt)
    ToPPT(gcf,'IFs',[Nact1D Nact1D ii],ii==1);
end;
end;

%% ID Slaving
nf; plot(xact,yact,'r*');
hold on;
Dslave = Dclip - dact*Mag;
ract = sqrt(xact.^2 + yact.^2);
slaveSet = find(ract>Dslave/2);
plot(xact(slaveSet),yact(slaveSet),'mo');
drawnow;
masterSet = setdiff(1:length(xact),slaveSet);

%now ID master indices
for ii=1:length(slaveSet)
    [mv,mi] = sort( sqrt( (xact(masterSet)-xact(slaveSet(ii))).^2 +
(yact(masterSet)-yact(slaveSet(ii))).^2) );
    master(ii) = masterSet(mi(1));
    plot([xact(slaveSet(ii)) xact(master(ii))],[yact(slaveSet(ii))
yact(master(ii))],'g--');
end;
hold on; plotCircle(rDM*2,0,0,'k--');
axis image;
drawnow;

if (ppt)
    ToPPT('Slaving Overlay');
end;

%% scan over the aperture size
sfcnt=0;
DclipWFS0 = Dclip;
sfVecMax = (2*rDM/DclipWFS0)
sfVec = 0.5:0.05:sfVecMax; %max is 1.0 because of the DM clip
for sf = sfVec;
    sfcnt=sfcnt+1;
    DclipWFS = DclipWFS0 .* sf
    DtotalClip = min([DclipWFS 2*(rDM-dsub*Mag)])
    ap = aperture(nxy,dxy,DclipWFS/2);
    gWFS = makegridwfs(5*Nact1D + 2, 5*Nact1D + 2,...
        dsub*Mag,dsub*Mag,DtotalClip);
    xsub = gWFS.xsub; ysub = gWFS.ysub;

    nf; plot(xact,yact,'r*'); hold on;
    plot(xsub,ysub,'bo');
    plotCircle(Dclip,0,0,'g--');
    plotCircle(Dap,0,0,'k--');
    plotCircle(DclipWFS,0,0,'m--');
    plotCircle(rDM*2,0,0,'b--');
    legend('Act','Sub','DM Clip','Max Aperture','WFS
Aperture','DM','Location','EastOutside');

```

```

    for ii=1:length(slaveSet)
        plot([xact(slaveSet(ii)) xact(master(ii))],[yact(slaveSet(ii))
yact(master(ii))], 'y--');
    end;
    axis image; drawnow;
    if (ppt)
        ToPPT('Overlay');
    end;

    %% make a poke matrix
    P.x = g.x; P.y = g.y;
    clear poke;
    for ii=1:length(IF);
        P.g = IF{ii};
        [sx, sy, phs, svec]=idealizedWFS(xsub, ysub, P, dsub, dsub);
        poke(:,ii)=svec;
    end;
    nf; imagesc(poke); title('Poke');
    if (ppt)
        ToPPT(gcf, 'Poke and Controls', [1 2 1], 1);
    end;

    %% make control matrix
    [ctrl, gains] = svdinverse(poke, 0);
    relativeGains = gains./max(gains);
    modesRemoved = sum((relativeGains<=0.01))
    [ctrl, gains] = svdinverse(poke, modesRemoved);

    ctrlSlaved = ctrl;
    for ii=1:length(slaveSet);
        ctrlSlaved(slaveSet(ii), :) = ctrlSlaved(master(ii), :);
    end;

    nf;
    subplot(1,2,1); imagesc(ctrl); title('Control');
    subplot(1,2,2); imagesc(ctrlSlaved); title('Slaved');
    if (ppt)
        ToPPT(gcf, 'Poke and Controls', [1 2 2], 0);
    end;

    nf; semilogy(gains, 'b*-');
    hold on;
    semilogy(gains(1:(length(gains)-modesRemoved)), 'ro');
    title('SVD Gains');
    drawnow;
    if (ppt)
        ToPPT(gcf, 'SVD Gains', [], 1);
    end;

    ctrlMatrix{1} = ctrl;
    ctrlMatrix{2} = ctrlSlaved;

    %% fit frames of Kolmogorov turbulence
    clear RMSWFE;
    for kk=1:length(ctrlMatrix);

```

```

k = KolmogorovTurbulenceScreen();
k.nxyFull = nxy;
k.nxySection = nxy;
k.dxy = dxy;
Nact = length(IF);
%nf([ 26          237          1315          900]);
nf([32          79          1060          600]);
for ii=1:Nframes;
    k.seed = ii;
    k.GenerateKTS();
    P.g = interp2(g.x,g.y',k.Screen,g.x/sf,g.y'/sf);
    P.g = ZeroNans(P.g);

    %this is not scaled properly
    [sx,sy,phs,svec]=idealizedWFS(xsub,ysub,P,dsub,dsub);
    cmds = ctrlMatrix{kk} * svec;
    phsDM = 0.0 .* IF{1};
    for jj = 1:Nact;
        phsDM = phsDM + cmds(jj) .* IF{jj};
    end;
    phsDM = phsDM - phsDM(nxy/2+1,nxy/2+1);
    delta = P.g - phsDM;
    delta = delta .* ap .* apDM;
    clf; subplot(2,3,1); show(P.x,P.y,P.g.*ap);
title('Aberration');
    subplot(2,3,2); show(P.x,P.y,phsDM.*ap.*apDM); title('DM &
SAs');

    %hold on; plot(xsub,ysub,'w*');
    %hold on; plot(xact,yact,'ko');
    subplot(2,3,3); show(P.x,P.y,delta); title('delta & SAs');
    %hold on; plot(xsub,ysub,'w*');
    subplot(2,1,2); plot(svec);title('Slopes Vector');

    % rms analysis
    deltaAp = delta .* ap .* apDM;
    RMSWFE{kk}(ii) = NonZeroRMS(deltaAp(:));
    drawnow;
    if (ppt && ii==1)
        ToPPT('Example Frame');
        pause(1.0);
    end;
end;
RMSWFESave(sfcnt, kk) = mean(RMSWFE{kk});
RMSWFESTDSave(sfcnt, kk) = std(RMSWFE{kk});
end;

nf;
for kk=1:length(ctrlMatrix);
    plot(RMSWFE{kk},getLineSpec(kk)); hold on;
end;
legend('Unslaved','Slaved');
if (ppt)
    ToPPT(sprintf('Result for SF=%.2f',sf));
end;
close all;
end;

```



```

%% plot the final results
nf;
for ii=1:size(RMSWFESave,2);
    errorbar(sfVec,RMSWFESave(:,ii),RMSWFESTDSave(:,ii),...
        getLineStyle(ii,'ColorAndMarker'));
    hold on;
end;
xlabel('Scaling Factor');
xlabel('D_w_f_s / D_d_m');
ylabel('RMS WFE (radians)');
legend('No Slaving','Slaving');
title(sprintf('DM Actuator Maximum Diameter at SF = 1.0'));
axis tight;
if (ppt)
    ToPPT('Summary Result');
end;

%% plot the final results
nf;
for ii=1:size(RMSWFESave,2);
    errorbar(sfVec.*DclipWFS0/(dact*Mag),RMSWFESave(:,ii),...
        RMSWFESTDSave(:,ii),getLineStyle(ii,'ColorAndMarker'));
    hold on;
end;
xlabel('Aperture Diameter /Actuator Spacing');
ylabel('RMS WFE (radians)');
legend('No Slaving','Slaving');
title(sprintf('DM Actuator Maximum Diameter = %.2f
Actuators',DclipWFS0./dact./Mag));
axis tight;
if (ppt)
    ToPPT('Summary Result');
end;
close all;
end;

```