ACTIVE OPTICAL SYSTEMS

AN066

Efficacy of Adaptive Optics When Rotation Exists Between a Deformable Mirror and a Hartmann Wavefront Sensor

Author: Dr. Justin Mansell
Revision: 7/29/25

1 Introduction

It is common that the rotational alignment of a WFS relative to a DM can be affected by the optical system transmitting the beam between the two. In this application note, we analyze whether this rotation can still create an effective control matrix for compensation.

2 Approach

For this example, we modeled the performance of an AO system with the 97-actuator pattern in Figure 1. As we evaluated the AO system performance, we changed both the pitch of the sub-apertures and their rotation relative to the fixed DM actuators. The nominal pitch (scale factor of 1.0) is the Fried geometry. We created a poke matrix-based control matrix at each AO overlay. We generated an aberration as a random sum of the first 12 Zernike terms (neglecting piston) and normalized its amplitude to make interpretation of the RMS residual wavefront error easier to interpret. We executed this model with plate-type influence functions. We eliminated any SVD mode that was less than 0.5% of the gain of the first mode, which happened to eliminate exactly 1 mode from each control matrix generation. We did not slave any actuators for this evaluation. We scanned the rotation angle from 0 to 45-degrees and the spacing scale factor from 1.0 (Fried) to 3.0 (triple-Fried). We evaluated the RMS WFE in the 30cm diameter region only.

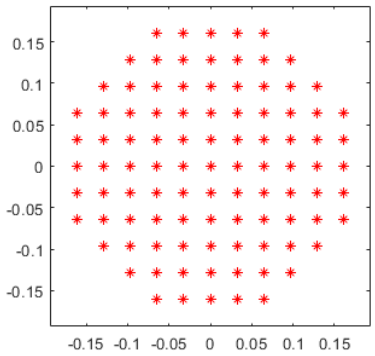

Figure 1: Actuator Pattern in Output Space

© Active Optical Systems, LLC

For each alignment condition, we examined the results with the plot shown in Figure 2. The plots from top-left to bottom-right reading left to right top to bottom are the AO overlay with different clip diameters, the residual wavefront error, the aberration phase, the fit to that phase, and the SVD gains with the condition number on the title of the plot.

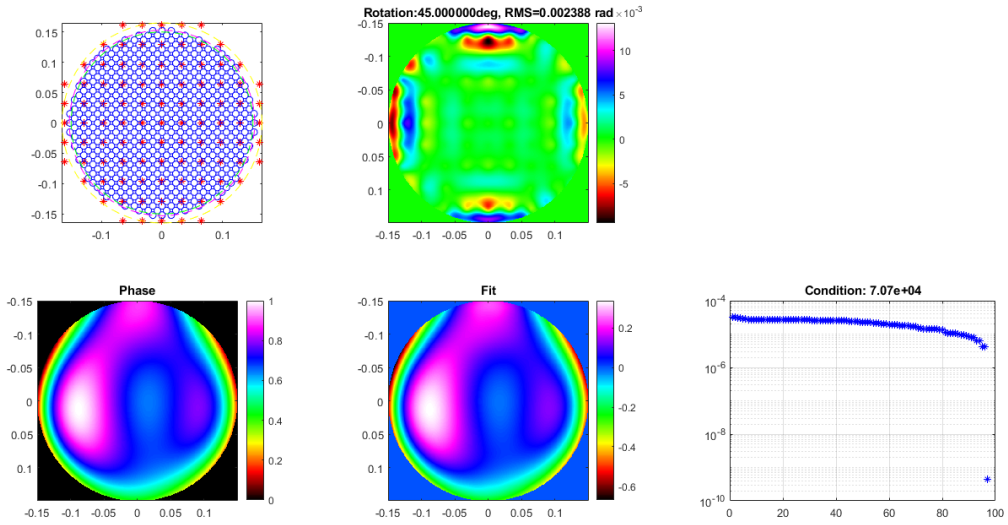

Figure 2: Single Point Result at Triple Fried and 45-degree Rotation

3 Results

Initial model runs were executed with a ring of actuators outside the WFS measurement space. This ended up producing results that showed a strong cliff in performance at the double-Fried geometry (scale factor of 2.0). We then tuned the system to include sub-apertures between the outer edge and the next inner edge of actuators and the results changed significantly. Figure 3 shows the output with the aberration shown in Figure 2. Near Fried geometry and just beyond it, the rotation has a more deleterious effect than beyond a sub-aperture scaling factor of 1.2.

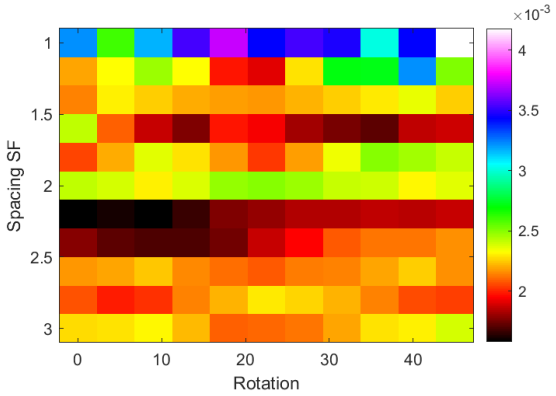

Figure 3: RMS WFE vs. Spacing Scale Factor and Rotation Angle (degrees)

We ran the model again with a different random Zernike sum and got largely the same results. Figure 4 shows these RMS WFE results versus SA pitch and rotation for this second random realization. Again, the results beyond about 1.5x Fried geometry were largely rotation independent.

© Active Optical Systems, LLC

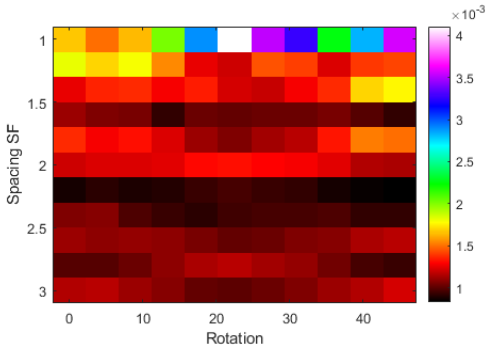

Figure 4: RMS WFE vs. SA Spacing and Rotation for a Different Random Zernike Realization

4 Conclusions

From this testing, we conclude that beyond a sub-aperture pitch 1.5x smaller than the actuator pitch, the AO system performance is largely independent of rotation angle.

© Active Optical Systems, LLC

5 Appendix: Analysis Code

```
%
Setup;
mov=1;
ppt=1;


%% Setup Propagation
Dap = 0.3;
RATIOsvd = 200;
AOSF = 1.1;
ClipSF = 1.02;
rZern = Dap/2;
wavelength = 1e-6;
nxy = 512;
dxy = Dap/nxy;
p=Propagation;
g = p.makeGrid(nxy,dxy);
% E = exp(-1*(g.r/w).^20);
E = (g.r<Dap/2*1.25);
ap = (g.r<Dap/2);
E = E ./ sqrt(sum(sum(abs(E).^2)));
P = sum(sum(abs(E).^2))

%% Setup Aberration
v = [0 0 0 1 0] * 2*pi; %focus only
% randn('seed',123);
randn('seed',12345);
v = randn(1,12).*2*pi;
phs = ZernikeCompose2(v,nxy,dxy,rZern);
phs = phs - nonZeroMin(ap.*phs);
phs = phs ./ nonZeroPV(ap.*phs);
E = E .* exp(1j*phs);

%% Setup AO
wfsType = 1;
Nact1D = 11;
ActSF = 1+2/Nact1D;
dact = Dap*ActSF/Nact1D;
a = AdaptiveOptics();
a.SetGrid(nxy,dxy);
a.SetWFSType(wfsType);
a.Dap = Dap;
a.wavelength = wavelength;
a.MakeActGrid(dact,Dap*ActSF);
a.IFtype = 2;
a.MakeIFs();

if (mov)
    v = VideoWriter(sprintf('Test_%s.avi',datestr(now,'')));
    v.FrameRate = 5;
    open(v);
end;
```


© Active Optical Systems, LLC

```
nf([.9 .8]);
dtVec=(0:0.1:1.0)*pi/4;
dsubSFVec = 1.0:0.2:3.0;
c=0;
c1 = 0;
for dsubSF = dsubSFVec;
    c1=c1+1;
    dsub = dact/dsubSF;
    aoxoff=dsub/4;
    aoyoff=dsub/4;
    c2=0;
    for dt = dtVec;
        c2=c2+1;
        c=c+1;
        a.MakeSubapertureGrid(dsub,Dap*AOSF,0,0);
        if (min(abs(a.xsub))>0)
            aoxoff = -1*min(abs(a.xsub)) - dsub/2;
            aoyoff = -1*min(abs(a.ysub)) - dsub/2;
        else
            aoxoff = -1*min(abs(a.xsub));
            aoyoff = -1*min(abs(a.ysub));
        end;
        a.MakeSubapertureGrid(dsub,Dap*AOSF*1.4,aoxoff,aoyoff);
        Nsa = length(a.xsub);
        for i=1:Nsa;
            ra = sqrt(a.xsub(i).^2+a.ysub(i).^2);
            ta = atan2(a.ysub(i),a.xsub(i));
            ta = ta + dt;
            a.xsub(i) = ra .* cos(ta);
            a.ysub(i) = ra .* sin(ta);
        end;

        % eliminate sub-apertures at edges
        ra = sqrt(a.xsub.^2+a.ysub.^2);
        idxvec = find(ra<Dap*ClipSF/2.0);
        a.xsub = a.xsub(idxvec);
        a.ysub = a.ysub(idxvec);

        clf;
        subplot(2,3,1);
        plot(a.xact,a.yact,'r*'); hold on; plot(a.xsub,a.ysub,'bo'); axis image;
        plotCircle(Dap,0,0,'g--');
        plotCircle(Dap*ClipSF,0,0,'m--');
        plotCircle(Dap*AOSF,0,0,'y--');
        Nact = length(a.xact)
        a.MakeControlMatrix();
        if (cond(a.poke)>RATIOsvd)
            gains = a.svdGains;
            gains=gains./max(gains(:));
            modesRemoved(c1,c2) = sum(gains<1/RATIOsvd)
        end;
        a.MakeControlMatrix();
        % nf; imagesc(a.ctrl); colorbar;
        [a,fit,cmds]=a.FitSlopes(E);
        fit = fit - fit(nxy/2+1,nxy/2+1);
```


© Active Optical Systems, LLC

```
        resid = fit-phs;
        resid = resid - resid(nxy/2+1,nxy/2+1);
        subplot(2,3,2); show(g.x,g.y,resid.*ap);
        RMSWFE(c) = NonZeroRMS(resid.*ap);
        RMSWFE2D(c1,c2) = NonZeroRMS(resid.*ap);
        title(sprintf('Rotation:%fdeg, RMS=%.6f rad',dt*180/pi,RMSWFE(c)));
        subplot(2,3,4); show(g.x,g.y,phs.*ap); title('Phase'); colorbar;
        subplot(2,3,5); show(g.x,g.y,fit.*ap); title('Fit'); colorbar;
        subplot(2,3,6); semilogy(a.svdGains,'b*'); grid on; title(sprintf('Condition:
%.2e',cond(a.poke)));
        drawnow;
        if (ppt) ToPPT2(); end;
        if (mov)
            writeVideo(v,getframe(gcf));
        end

        if (RMSWFE(c)>1.0)
            disp('Model Error');
            pause(1.0);
        end;
    end;
end;

if(mov) close(v); end;
%% final plots
nf; imagesc(dtVec*180/pi,dsubSFVec,maxmin(RMSWFE2D,.1,-.1));
xlabel('Rotation');
ylabel('Spacing SF');
colorbar;
if (ppt) ToPPT2(); end;
```


© Active Optical Systems, LLC