

Image Motion Compensation

A tutorial including analysis of a catadioptric ultra-telephoto lens

1. Introduction

Imaging systems are always subject to mechanical disturbances. They move because they are held by a person, or mounted on a moving vehicle. The structures they are mounted to are subject to mechanical vibration. If the angular motion of the camera over an integration time is comparable to the instantaneous field-of-view of a pixel, then the image will be smeared.

In general, image motion compensation refers to active control of something (optical element position, focal plane position, index of refraction, surface curvatures, etc) to stabilize the object space line-of-sight (LOS) of the focal plane array (FPA). The goal is to compensate for unwanted motions of the camera. This requires measuring that motion using some sort of gyroscope or other motion sensor. A discussion of the measurement of camera motion is outside the scope of this tutorial.

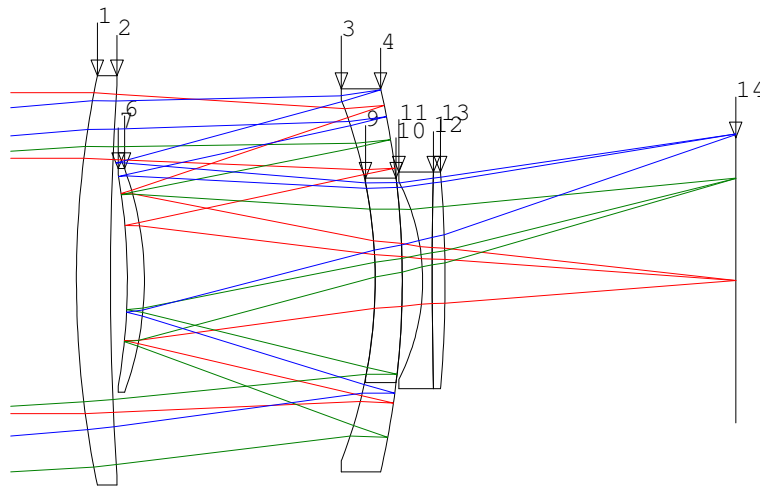


Figure 1: A catadioptric ultra-telephoto lens (Ogino, 1981)

As an example, consider the catadioptric lens system of Figure 1. This is one of the realizations described in the US patent (Ogino, 1981), assigned to the Minolta corporation. The original Japanese patent number is 61-48132. This and several other similar lenses are in the Code-V patent lens database, referenced by the Japanese patent number. It was chosen as a representative catadioptric system of moderate complexity. The lens focal length is 100 mm. The focal ratio is $f/4.5$. Note that about 40% of the pupil area is obscured, so from a radiometric perspective the system $f/\#$ is more like $4.5/\sqrt{0.6} = 5.8$. In the camera world, such lenses are known as ultra-telephoto mirror lenses.

The rays drawn are associated with three objects at infinity. The field angles and corresponding normalized field positions are

Field identifier	Field angle (deg)	Normalized field position
F1	0	0.00
F2	3.42	0.71
F3	4.84	1.00

Table 1: Reference field positions

The normalized pupil positions chosen for illustration range from ± 0.65 to ± 1.00 .

The optical elements of this system are

- A positive meniscus lens (Surfaces 1 and 2)
- A multiple-pass, primary negative meniscus lens/mirror (Surfaces 3—5, 9—10)
- A second-surface, secondary negative meniscus lens/mirror (Surfaces 5—8)
- A lens doublet (Surfaces 11—13)

Consider next a rotation of the entire camera system by 0.1 deg during a sensor integration period. This is equivalent to a -0.1 deg change of the field angles. This is illustrated in Figure 2.

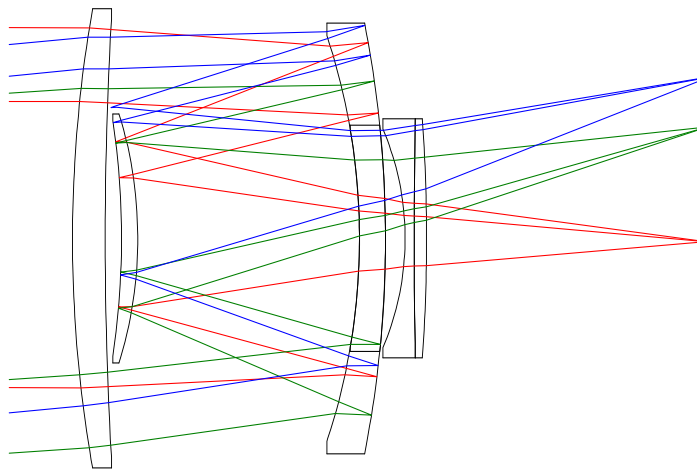


Figure 2: Fields rotated by 0.1 deg, resulting from a motion of the camera

The fields have changed so slightly that it is barely perceptible in the figure. Note that this is more than sufficient to smear the image. Think of Figure 1 as corresponding to the system at the beginning of a detector integration period and Figure 2 as corresponding to the system at the end of an integration period.

The goal of motion compensation is to modify the system so that the rotated fields of Figure 2 come to focus on the same point on the focal plane as the fields of Figure 1.

2. Common methods of image motion compensation

There are four more-or-less common methods of compensating for image motion:

1. Moving the entire optical system
2. Moving the focal plane array
3. Adding a flat, fast steering mirror (FSM)
4. Moving optical groups

It is the last method, moving optical groups, that we are most interested in. But we shall mention the other three methods briefly.

The first method is to move the entire optical system, as illustrated schematically in Figure 3. For this method, the camera is mounted on an inertially stabilized platform.

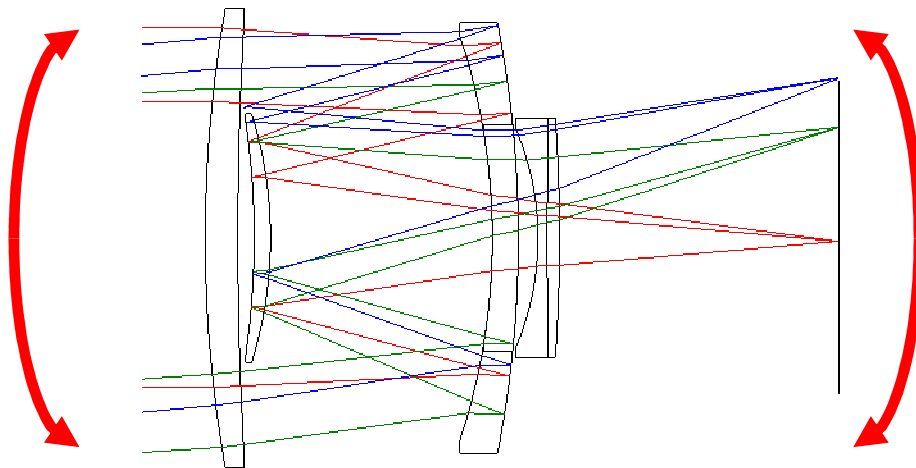


Figure 3: Image motion compensation rotating the entire camera system

Normally this is done with a serial gimbal system (e.g., first an azimuthal rotation, then an elevation rotation). There are exceptions, though. For an example of a parallel mechanism used for camera stabilization see (Gosselin, 1997). That particular system probably does not have a gyroscope for measuring camera motion.

This method of image motion compensation is convenient and often required if one already has a gimbal system for pointing the camera. It is thus common for vehicle-mounted cameras, which often have such gimbals. Clearly it is inconvenient or impossible for reasons of size and/or cost for many applications, such as hand-held cameras.

The advantage of this method is that it does not introduce optical aberrations. So as long as the motion stabilization system works perfectly, there is no degradation of image quality. In fact, though, the motion stabilization system will not work perfectly, and there will be image smear. So there is limit to the performance that can be achieved by such systems. For very high performance systems, a gimbal control system will be used in combination with one of the other methods (moving the focal plane, adding a FSM, moving optical groups).

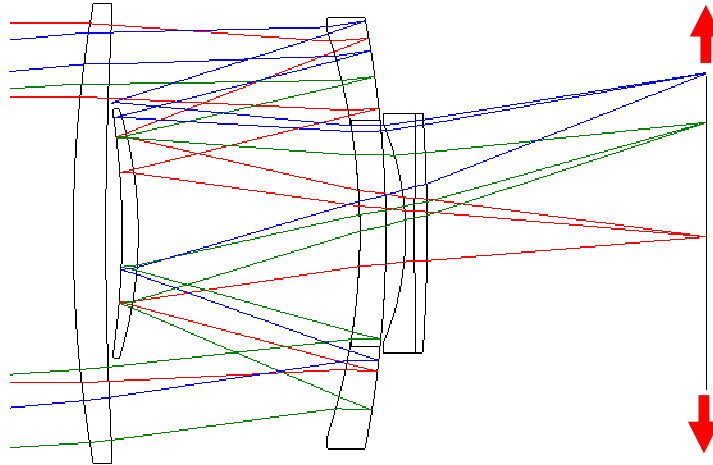


Figure 4: Image motion compensation moving the focal plane

The next method is moving the focal plane array. This is very straightforward conceptually. If the field angle changes by an angle of $\Delta\theta_{\text{Field}} = 0.1 \text{ deg} = 1.745 \text{ mrad}$, then the focal plane needs to be moved by

$$\Delta\varepsilon_{\text{FPA}} = f \cdot \Delta\theta_{\text{Field}} \quad (1)$$

where f is the system focal length, 100 mm for our example. So in this case the focal plane needs to be moved by $174.5 \mu\text{m}$. This method of motion compensation is feasible both technically and economically. Consumer camera manufacturers using this technique include Sony, Pentax, Olympus, Fujifilm, Samsung, Cassio, and Ricoh Caplio (Wikipedia, 2008). Note that the FPA is not always easily accessible for motion compensation, particularly when it is inside a cryogenically cooled Dewar assembly.

An advantage of moving the focal plane is that no optical aberrations are introduced. Of course field-dependent aberrations will change due to the change of field angle, but no new aberrations are introduced.

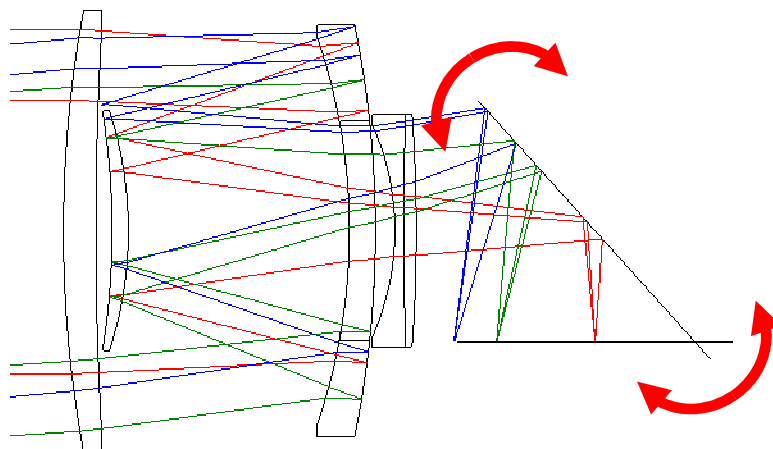


Figure 5: Image motion compensation using a flat, fast steering mirror

Next we consider motion compensation via the introduction of a flat FSM, as illustrated in Figure 5. The steering mirror is assumed to be flat. In this case, no aberrations are introduced. The mirror is rotated small amounts using something like electromagnetic or piezoelectric actuators.

This method is often applied for high performance, larger systems such as space and airborne imagers. For smaller systems, packaging becomes an issue. The author is not aware of this method having been applied to consumer applications.

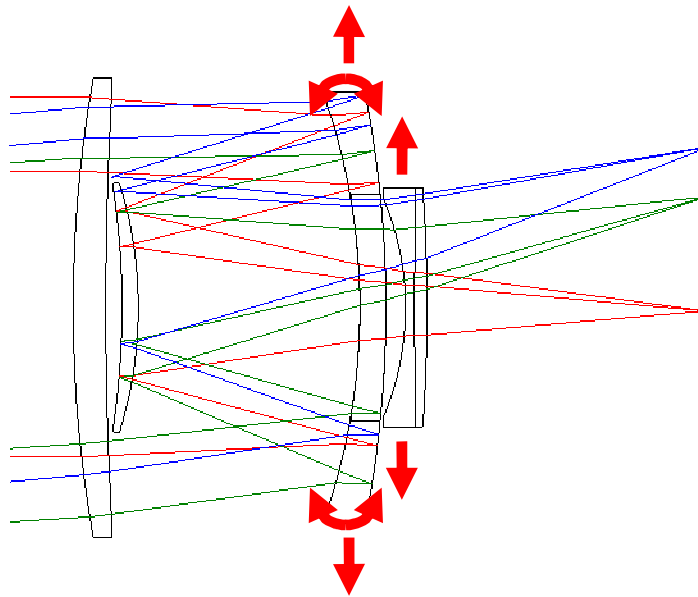


Figure 6: Image motion compensation moving powered optical elements

Finally we consider motion compensation by moving powered elements. For our example system, three possibilities come to mind

1. Tilting of the primary mirror (multi-pass, catadioptric element)
2. Lateral displacement of the primary mirror
3. Lateral displacement of the doublet

It would also be possible in theory to tilt or displace the meniscus lens and secondary mirror assembly. For this discussion it is assumed that would be precluded due to packaging constraints. We'll look at each of the three listed methods in Section 3.

Before moving on though, we note that this method of motion compensation is used by several commercial camera companies, in particular Canon and Nikon.

3. Analysis of catadioptric ultra-telephoto system

The optical prescription of the system is given in Table 2.

Surface #	Radius	Thickness	Index	Abbe #	Semi-Aperture
Object	Infinity	Infinity			
Stop	60.5720	2.0000	1.5176	53	11.1372
2	173.8170	15.6000			11.0649
3	-29.8250	1.6000	1.5168	64	10.2000
4	-51.1740	-1.6000	1.5168	64	10.4000
5	-29.8250	-13.6000			9.6902
6	-19.3730	-1.0000	1.5168	64	5.9000
7	-36.9270	1.0000	1.5168	64	5.7641
8	-19.3730	13.6000			5.8103
9	-29.8250	1.6000	1.5168	54	5.2979
10	-51.1740	1.2000			5.3649
11	-13.0150	0.5600	1.5168	81	5.3657
12	266.6470	0.7600	1.7400	62	5.6363
13	-77.9790	17.0322			5.7000
Image	Infinity	0.0981			8.6788

Table 2: Optical prescription (All distances in mm)

The system was focused in Code-V using the minimum wavefront error (WFE) criterion (Quick Best Focus). This minimizes the weighted RMS of the RMS WFE for three wavelengths and three field positions. There are two averaging operations. For each wavelength, the RMS error across the pupil is computed. Then a weighted RMS of the three wavelengths is computed. In this case, equal weights are used. The result shall be referred to as a *weighted RMS WFE* as a reminder that the WFE is an average across multiple wavelengths.

The field positions were given in Table 1. The reference wavelengths are given in Table 3. The system was focused once and only once. This is consistent with the fact that the system cannot be refocused during an integration period. For a fixed focus system it would be more appropriate to focus the system at a hyperfocal distance rather than infinity.

Wavelength identifier	Wavelength (nm)
W1	656
W2	589
W3	434

Table 3: Reference wavelengths

For each field angle, there are two performance criteria of interest: image location and image quality. The following metrics are used:

Image location metric: Image height y_{image} (mm) of the ray going through the $y_p = 0.65$ pupil position for the W2 (589 nm) wavelength.

Image quality: Weighted RMS of RMS wavefront error (waves). RMS wavefront error is computed for each wavelength across the pupil. Then a second RMS across wavelengths is performed.

Again, the wavelength weights are equal, but the adjective *weighted* is retained as a reminder that we are averaging across wavelengths. As if that weren't enough averaging, we then average across the field angles!

Wavefront errors and image positions for the nominal system shown in Figure 1 are given in Table 4.

Field	Weighted ΔW_{RMS}	y_{Image} of W2, $y_p = 0.65$
0.00	0.092406	-0.00169
0.71	0.630788	6.06034
1.00	0.832963	8.67177
AVE:	0.580905	

Table 4: Reference wavefront errors and image positions

Next we decrement the field angles by 0.1 deg, corresponding to Figure 2. Remember that this field angle change is associated with motion of the camera, not the object. So really we want the image positions to be the same. But of course they are not. The image positions move by about 176 μm , as expected.

Field	Weighted ΔW_{RMS}	y_{Image} of W2, $y_p = 0.65$
0.00	0.094484	-0.17703
0.71	0.609964	5.87943
1.00	0.820255	8.48489
AVE:	0.568817	

Table 5: Wavefront errors and image positions of disturbed system without motion compensation

Our next step is to compensate for this image motion by moving optical elements.

3.1. IMC using tilt of primary mirror groups

Our first strategy is to compensate by tilting the primary mirror. Unfortunately, this "mirror" is really a multi-pass, catadioptric system. It turns out that most of the image motion comes from the rotation of the reflective surface S4. But we have to consider the refractive surfaces S3, S5, S9 and S10 as well.

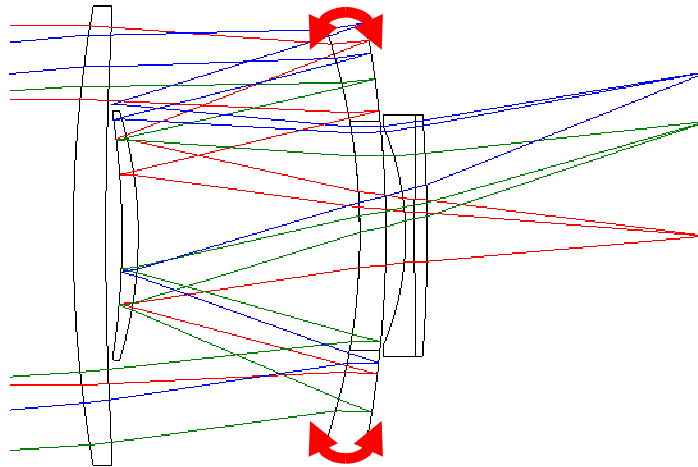


Figure 7: Image motion compensation via tilt of the primary mirror groups

It's useful to decompose the surfaces of this mirror into two optical groups.

- Group G1 consists of surfaces S3, S4 and S5
- Group G2 consists of surfaces S9 and S10

Gaussian reduction of these groups is performed using the Matlab script listed in the Appendix. The properties of these groups are:

- Group G1
 - The first and second principal planes are coincident, $d_{G1} = d'_{G1} = 1.0359$ mm to the right of the vertex of surface S3
 - The power is $\phi_{G1} = 0.026178$ mm⁻¹, corresponding to a focal length of $f_{G1} = 38.2$ mm
- Group G2
 - The first principal plane is $d_{G2} = -1.5123$ mm to the right of the vertex of surface S9 (in other words, it is 1.5123 mm to the left of the vertex)
 - The second principal plane is $d'_{G2} = -2.5948$ mm to the right of the vertex of surface S10
 - The distance between the principal planes is $PP'_{G2} = 1.7175$ mm
 - The power is $\phi_{G2} = -0.007044$ mm⁻¹, corresponding to a focal length of $f_{G2} = -141.9592$ mm

The sensitivities of image motion to element motion can be computed using the methods described in (Burge, 2006). To this end, we need to know some parameters of the Gaussian equivalent system such as the marginal ray bundle diameters and associated numerical apertures. This information was obtained by a paraxial raytrace of a marginal ray associated with an on-axis object at infinity (zero field angle). We determine, for example, that the marginal ray bundle diameters of groups G1 and G2 are $B_{G1} = 19.9481$ mm and $B_{G2} = 5.2348$ mm. The numerical aperture exiting groups G1 and G2 are $NA_{G1} = 0.1885$ and $NA_{G2} = 0.1885$. The final, system numerical aperture is $NA = 0.1114$.

Let ε_{G1} and ε_{G2} be the focal plane motion due to movement of groups G1 and G2. Of course these groups move together, so the total motion is $\varepsilon = \varepsilon_{G1} + \varepsilon_{G2}$. By design, assume that the mirror is rotated by an actuated mechanism about the principal point of group G1. Let θ_{G1} be the angle of this rotation. Then the sensitivities of focal plane motion with respect to angle are

$$\frac{d\varepsilon_{G1}}{\theta_{G1}} = 2 \cdot f/\# \cdot B_{G1} = 179.0758 \text{ mm/rad} \quad (2)$$

$$\frac{d\varepsilon_{G2}}{\theta_{G1}} = \varphi_{G2} \cdot (d_{G2} - d_{G1}) \cdot f/\# \cdot B_{G2} - PP'_{G2} \frac{NA_{G2}}{NA} = -2.4847 \text{ mm/rad} \quad (3)$$

The combined sensitivity is then

$$\frac{d\varepsilon}{\theta_{G1}} = \frac{d\varepsilon_{G1}}{\theta_{G1}} + \frac{d\varepsilon_{G2}}{\theta_{G1}} = 176.5911 \text{ mm/rad} \quad (4)$$

We see that the sensitivity is dominated by the mirror of group G1. The rotation required to compensate for the field change of $\Delta\theta_{\text{Field}} = 0.1 \text{ deg} = 1.745 \times 10^{-3} \text{ rad}$ is then

$$\theta_{G1} = \left(\frac{d\varepsilon}{\theta_{G1}}\right)^{-1} f \cdot \Delta\theta_{\text{Field}} = -988.14 \text{ } \mu\text{rad} \quad (5)$$

The optical surfaces are displaced accordingly in the Code-V sequence file. Computing the wavefront errors and tracing the 0.65 pupil position ray yields the following errors and image positions.

Field	Weighted ΔW_{RMS}	y_{Image} of W2, $y_p = 0.65$
0.00	0.096466	-0.00015
0.71	0.617780	6.06217
1.00	0.824420	8.67363
AVE:	0.572908	

Table 6: Wavefront errors and image positions of given a compensatory rotation of the primary mirror groups

3.1. IMC using lateral displacement of primary mirror groups

Our second strategy is to compensate by displacing the primary mirror laterally. This is illustrated schematically in Figure 8.

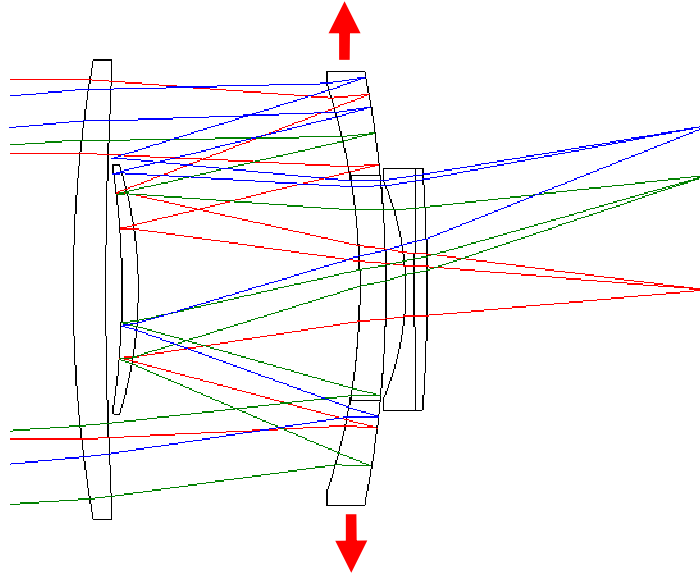


Figure 8: Image motion compensation via lateral displacement of the primary mirror groups

All information necessary to compute the sensitivity of focal plane motion with respect to lateral displacements of groups G1 and G2 has already been presented. Let s_{G1} be the upwards, lateral displacement of group G1 and thus G2. Then the sensitivities of focal plane motion with respect to this displacement are

$$\frac{d\varepsilon_{G1}}{s_{G1}} = \varphi_{G1} \cdot f/\# \cdot B_{G1} = 2.3439 \text{ mm/mm} \quad (6)$$

$$\frac{d\varepsilon_{G2}}{s_{G1}} = \varphi_{G2} \cdot f/\# \cdot B_{G2} = -0.1655 \text{ mm/mm} \quad (7)$$

The combined sensitivity is then

$$\frac{d\varepsilon}{s_{G1}} = \frac{d\varepsilon_{G1}}{s_{G1}} + \frac{d\varepsilon_{G2}}{s_{G1}} = 2.1784 \text{ mm/mm} \quad (8)$$

Again the sensitivity is dominated by group G1. The displacement required to compensate for the field change of $\Delta\theta_{\text{Field}} = 0.1 \text{ deg} = 1.745 \times 10^{-3} \text{ rad}$ is then

$$s_{G1} = \left(\frac{d\varepsilon}{s_{G1}} \right)^{-1} f \cdot \Delta\theta_{\text{Field}} = 0.0801 \text{ mm} \quad (9)$$

The optical surfaces are displaced accordingly before recomputing the wavefront errors and image positions.

Field	Weighted ΔW_{RMS}	y_{Image} of W2, $y_p = 0.65$
0.00	0.204793	-0.00624
0.71	0.523547	6.05752
1.00	0.741858	8.66957
AVE:	0.516871	

Table 7: Wavefront errors and image positions of given a compensatory lateral displacement of the primary mirror groups

3.1.IMC using lateral displacement of the lens doublet

Our final strategy is to compensate by laterally displacing the lens doublet (group G3). This is illustrated schematically in Figure 8.

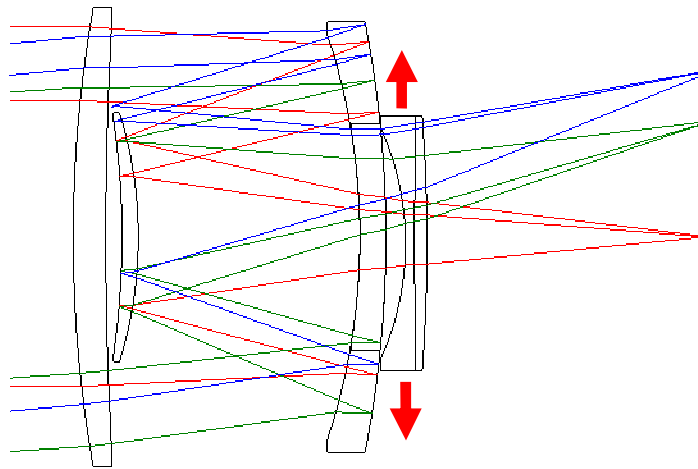


Figure 9: Image motion compensation via lateral displacement of the lens doublet

We consider doublet as a group of three surfaces:

- Group G3 consists of surfaces S11, S12 and S13

Gaussian reduction of these groups is performed using the Matlab script listed in the Appendix. The properties of this group are:

- Group G3
 - The first principal plane is $d_{G3} = -0.2737$ mm to the right of the vertex of surface S11 (in other words, it is 0.2737 mm to the left of the vertex)
 - The second principal plane is $d'_{G3} = -1.0882$ mm to the right of the vertex of surface S12
 - The distance between the principal planes is $PP'_{G3} = 0.5055$ mm
 - The power is $\phi_{G3} = -0.029069$ mm⁻¹, corresponding to a focal length of $f_{G3} = -34.4012$ mm

We also need the marginal ray bundle diameter. From a paraxial ray trace we determine that $B_{G3} = 4.0371$ mm.

Let s_{G3} be the upwards, lateral displacement of group G3. Then the sensitivity of focal plane motion with respect to this displacement is

$$\frac{d\varepsilon}{s_{G3}} = \frac{d\varepsilon_{G3}}{s_{G3}} = \varphi_{G3} \cdot f/\# \cdot B_{G3} = -0.5267 \text{ mm/mm} \quad (10)$$

The displacement required to compensate for the field change of $\Delta\theta_{\text{Field}} = 0.1 \text{ deg} = 1.745 \times 10^{-3} \text{ rad}$ is

$$s_{G3} = \left(\frac{d\varepsilon}{s_{G3}}\right)^{-1} f \cdot \Delta\theta_{\text{Field}} = -0.3313 \text{ mm} \quad (11)$$

The optical surfaces of the doublet are displaced accordingly before recomputing the wavefront errors and image positions.

Field	Weighted ΔW_{RMS}	y_{Image} of W2, $y_p = 0.65$
0.00	0.315091	0.01273
0.71	1.038235	6.10751
1.00	1.277895	8.74154
AVE:	0.932778	

Table 8: Wavefront errors and image positions of given a compensatory lateral displacement of the doublet group

3.1. Performance comparison

Not enough analysis has been performed to making any grand conclusions. Still it is interesting to compare the performance of the three methods. First we consider the wavefront errors.

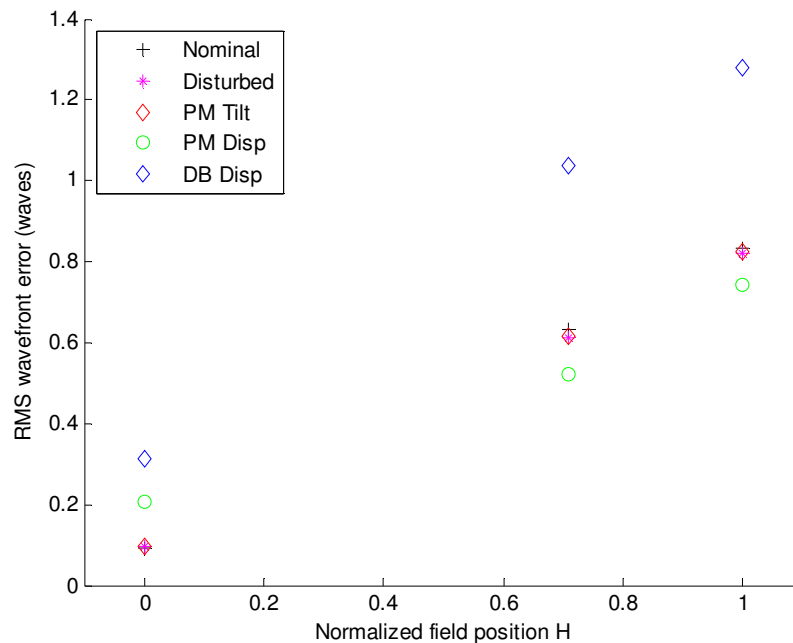


Figure 10: RMS wavefront error comparison

The *Nominal* data corresponds to Figure 1, where there is no image displacement. The *Disturbed* data corresponds to Figure 2, where the field is rotated by 0.1 deg due to camera motion. This is a small change of field relative to the field-of-view of ± 4.84 deg, so the wavefront error changes very little. There is not enough data to make conclusions about the performance of tilting the primary mirror groups (*PM Tilt* data) and laterally displacing the primary mirror groups (*PM Disp* data). What is clear is that the wavefront errors resulting from laterally displacing the doublet (*DB Disp* data) are significantly larger.

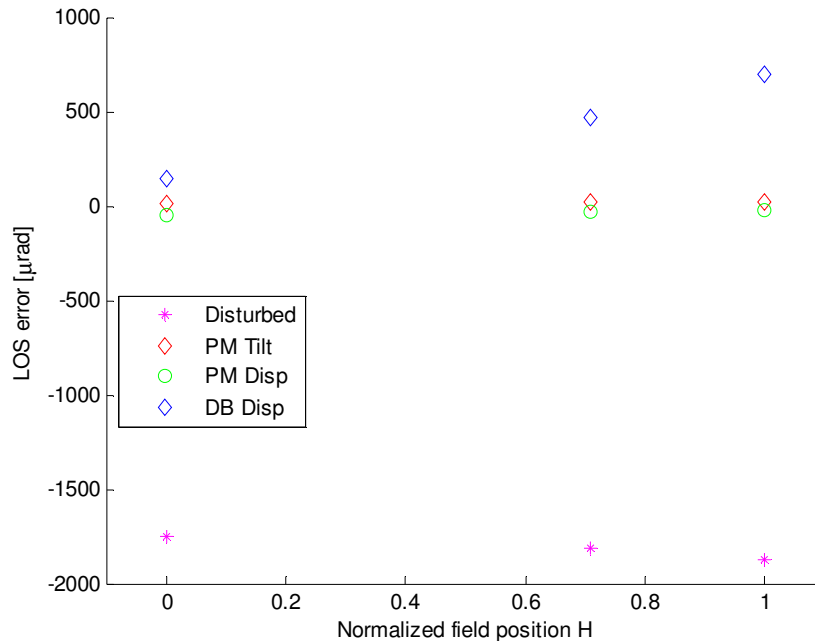


Figure 11: Line-of-sight error comparison

Next consider the line-of-sight errors. The nominal LOS corresponding to Figure 1 is used as a reference. For each case, the reference image height is subtracted from the corresponding image height. This yields an error in mm. That is then divided by the system focal length of 100 mm to convert to angle.

Again there is not enough data to make conclusions about the performance of tilting the primary mirror groups (*PM Tilt* data) and laterally displacing the primary mirror groups (*PM Disp* data). What is clear is that the LOS errors resulting from laterally displacing the doublet (*DB Disp* data) are significantly larger. Indeed, there is evidence of overcompensation. The most likely reasons for this are

- The displacement of the doublet (-0.33 mm) is large enough that the linear sensitivity relation is inaccurate.
- The author made an analytical or computational mistake.

4. Conclusion

An overview was given of four more-or-less common methods of compensating for image motion:

1. Moving the entire optical system
2. Moving the focal plane array
3. Adding a flat, fast steering mirror (FSM)
4. Moving optical groups

Special attention was given to the latter method of moving optical groups. We saw that for a given optical system there are a number of ways of changing the LOS. The optical performance (RMS WFE, LOS pointing error) will depend on which elements are moved and how.

5. Bibliography

Burge, J. (2006). An easy way to relate optical element motion to system pointing stability. *Current Developments in Lens Design and Optical Engineering VII*. 6288, pp. 62880I-1--12. San Diego: SPIE.

Gosselin, C. (1997). *L'oeil agile*. Retrieved November 23, 2008, from Laboratoire de robotique: <http://robot.gmc.ulaval.ca/fr/recherche/theme103.html>

Ogino, S. (1981). *Patent No. 4,264,136*. United States of America.

Wikipedia. (2008). *Image stabilization*. Retrieved November 23, 2008, from Wikipedia: http://en.wikipedia.org/wiki/Image_stabilization

Appendix A: Matlab listings

This appendix contains the listings to two Matlab files. The first file, `imcTutorial.m`, is a script that does a number of things like Gaussian reduction of subsystems, paraxial raytracing, and generation of compensatory rotations and displacements. The second file, `genSeqFile.m`, is a function that generates Code-V command sequence files. Together these files have all information necessary to duplicate the results.

A.1: `imcTutorial.m`

```
% OPTI-521, E.D. Fasse
% Tutorial first-order analysis

RSurface = zeros(13,1); % Surface radii of curvature (mm)
RSurface(1) = 60.5720;
RSurface(2) = 173.8170;
RSurface(3) = -29.8250;
RSurface(4) = -51.1740;
RSurface(5) = -29.8250;
RSurface(6) = -19.3730;
RSurface(7) = -36.9270;
```

```
RSurface(8) = -19.3730;
RSurface(9) = -29.8250;
RSurface(10) = -51.1740;
RSurface(11) = -13.0150;
RSurface(12) = 266.6470;
RSurface(13) = -77.9790;

nNext = zeros(13,1); % Index of next material
nNext(1) = 1.517600;
nNext(2) = 1;
nNext(3) = 1.516800;
nNext(4) = -1.516800;
nNext(5) = -1;
nNext(6) = -1.516800;
nNext(7) = 1.516800;
nNext(8) = 1;
nNext(9) = 1.516800;
nNext(10) = 1;
nNext(11) = 1.516800;
nNext(12) = 1.740000;
nNext(13) = 1;

nPrev = zeros(13,1); % Index of previous material
nPrev(1) = 1;
nPrev(2:13) = nNext(1:12);

t = zeros(13,1); % Thickness of subsequent space
t(1) = 2;
t(2) = 15.6;
t(3) = 1.6;
t(4) = -1.6;
t(5) = -13.6;
t(6) = -1;
t(7) = 1;
t(8) = 13.6;
t(9) = 1.6;
t(10) = 1.2;
t(11) = 0.56;
t(12) = 0.76;
t(13) = 17.0322;

tau = t ./ nNext; % Associated optical thickness

phi = (nNext - nPrev) ./ RSurface; % Surface power

% Marginal ray trace, on-axis ray
y = zeros(14,1);
y(1) = 11.1372;
u = zeros(14,1);
u(1) = 0;
z = zeros(14,1);
z(1) = 0;

N = 13;
for j=1:N
```

```

    u(j+1) = u(j) - phi(j) * y(j);
    y(j+1) = y(j) + tau(j) * u(j+1);
    z(j+1) = z(j) + t(j);
end

figure(1); clf; hold on;
plot(z(1:N),y(1:N), 'b');
axis equal

f = - y(1) / u(end);
fNo = f / (2*y(1));
NA = - u;
B = 2 * y;

% Construct Gaussian equivalent system

% Gaussian reduction of catadioptric group G1
phi1 = phi(3) + phi(4) - phi(3) * phi(4) * tau(3);
d1_G1 = nPrev(3) * tau(3) * phi(4) / phi1;
dp1_over_n_G1 = - tau(3) * phi(3) / phi1;

tau2 = tau(4) - dp1_over_n_G1;
phi_G1 = phi1 + phi(5) - phi1 * phi(5) * tau2;
d2_G1 = nPrev(3) * tau2 * phi(5) / phi_G1;
dp2_G1 = -nNext(5) * tau2 * phi1 / phi_G1;

d_G1 = d1_G1 + d2_G1;
dp_G1 = dp2_G1;
tPP_G1 = t(3) + t(4) - d_G1 + dp_G1;

% Gaussian reduction of lens (Group G2)
phi_G2 = phi(9) + phi(10) - phi(9) * phi(10) * tau(9);
d_G2 = nPrev(9) * tau(9) * phi(10) / phi_G2;
dp_G2 = - nNext(10) * tau(9) * phi(9) / phi_G2;

tPP_G2 = t(9) - d_G2 + dp_G2;

% Gaussian reduction of lens doublet group G3
phi1 = phi(11) + phi(12) - phi(11) * phi(12) * tau(11);
d1_G3 = nPrev(11) * tau(11) * phi(12) / phi1;
dp1_over_n_G3 = - tau(11) * phi(11) / phi1;

tau2 = tau(12) - dp1_over_n_G3;
phi_G3 = phi1 + phi(13) - phi1 * phi(13) * tau2;
d2_G3 = nPrev(11) * tau2 * phi(13) / phi_G3;
dp2_G3 = -nNext(13) * tau2 * phi1 / phi_G3;

d_G3 = d1_G3 + d2_G3;
dp_G3 = dp2_G3;
tPP_G3 = t(11) + t(12) - d_G3 + dp_G3;

% Gaussian equivalent system parameters (System GE)
phi_GE = [phi(1:2); phi_G1; phi(6:8); phi_G2; phi_G3];
t_GE = [t(1); ...

```



```

t(2) + d_G1; ...
t(5) - dp_G1; ...
t(6); ...
t(7); ...
t(8) + d_G2;
t(10) - dp_G2 + d_G3; ...
t(13) - dp_G3];

tPP = zeros(8,1);
tPP(3) = tPP_G1;
tPP(7) = tPP_G2;
tPP(8) = tPP_G3;

nNext_GE = [nNext(1:2); nNext(5:8); nNext(10); nNext(end)];

tau_GE = t_GE ./ nNext_GE;

% Marginal ray trace, on-axis ray
N = length(phi_GE);

y_GE = zeros(N+1,1);
y_GE(1) = 11.1372;
u_GE = zeros(N+1,1);
u_GE(1) = 0;
zPP2 = zeros(N+1,1);
zPP2(1) = 0;
zPP1 = zeros(N+1,1);

for j=1:N
    u_GE(j+1) = u_GE(j) - phi_GE(j) * y_GE(j);
    y_GE(j+1) = y_GE(j) + tau_GE(j) * u_GE(j+1);
    zPP1(j+1) = zPP2(j) + t_GE(j);
    if(j<N)
        zPP2(j+1) = zPP1(j+1) + tPP(j+1);
    else
        zPP2(j+1) = zPP1(j+1);
    end
end

for j=1:N
    plot([zPP2(j), zPP1(j+1)], [y_GE(j), y_GE(j+1)], 'r:');
    if( tPP(j) ~= 0 )
        plot([zPP1(j), zPP2(j)], [y_GE(j), y_GE(j)], 'r:');
    end
end

axis equal

f_GE = - y_GE(1) / u_GE(end);
fNo_GE = f_GE / (2 * y_GE(1));
NA_GE = - u_GE;
B_GE = 2 * y_GE;

% Sanity check
U_U_GE = [u, [u_GE(1:3)];

```

```

zeros(2,1); u_GE(4:7); 0; u_GE(8); zeros(2,1); u_GE(9)]];

fprintf('f = %f, %f\n', f, f_GE);
fprintf('f/# = %f, %f\n', fNo, fNo_GE);

% LOS sensitivities
% Group G1 sensitivity with respect to angle and decenter
B_G1 = B_GE(3);
deG1_dthG1 = 2 * fNo * B_G1;      % [mm/rad]
deG1_dsG1 = phi_G1 * fNo * B_G1; % [mm/rad]

% Group G2 sensitivity with respect to angle and decenter
NA_G2 = NA_GE(7);
B_G2 = B_GE(7);
deG2_dthG1 = phi_G2 * (d_G2 - d_G1) * fNo * B_G2 ...
    - tPP_G2 * NA_G2 / NA(end);
deG2_dsG1 = phi_G2 * fNo * B_G2; % [mm/mm]

% Group G3 sensitivity with respect to decenter
B_G3 = B_GE(8);
deG3_dsG3 = phi_G3 * fNo * B_G3; % [mm/mm]

% Displacements required to offset a 0.1 deg LOS disturbance
angDistDeg = -0.1;
angDistRad = angDistDeg * pi / 180;
eDist = angDistRad * f;

de_dthG1 = deG1_dthG1 + deG2_dthG1;
de_dsG1 = deG1_dsG1 + deG2_dsG1;
de_dsG3 = deG3_dsG3;

thG1 = eDist / de_dthG1; % [rad]
sG1 = -eDist / de_dsG1; % [mm]
sG3 = -eDist / de_dsG3; % [mm]

dz = zeros(13,1);
dy = zeros(13,1);

% Nominal case, no disturbance or compensation
genSeqFile( 0, 0, dy, dz, 'Nominal' );

% Add a steering mirror (really a fold mirror just for illustration)
genFoldedSeqFile( 0, 0, dy, dz, 'Folded' );

% Add disturbance but no compensation
genSeqFile( angDistDeg, 0, dy, dz, 'Disturbed' );

% Add disturbance and compensation
zPP1_G1 = zPP1(3);
dz(3) = (z(3) - zPP1_G1) * (cos(thG1) - 1);
dy(3) = (z(3) - zPP1_G1) * sin(thG1);
dz(4) = (z(4) - zPP1_G1) * (cos(thG1) - 1);
dy(4) = (z(4) - zPP1_G1) * sin(thG1);
dz(5) = (z(5) - zPP1_G1) * (cos(thG1) - 1);

```

```

dy(5) = (z(5) - zPP1_G1) * sin(thG1);
dz(9) = (z(9) - zPP1_G1) * (cos(thG1) - 1);
dy(9) = (z(9) - zPP1_G1) * sin(thG1);
dz(10) = (z(10) - zPP1_G1) * (cos(thG1) - 1);
dy(10) = (z(10) - zPP1_G1) * sin(thG1);

genSeqFile( angDistDeg, thG1, dy, dz, 'thG1' );

dz = zeros(13,1);
dy = zeros(13,1);
dy(3) = sG1;
dy(4) = sG1;
dy(5) = sG1;
dy(9) = sG1;
dy(10) = sG1;
genSeqFile( angDistDeg, 0, dy, dz, 'sG1' );

dz = zeros(13,1);
dy = zeros(13,1);
dy(11) = sG3;
dy(12) = sG3;
dy(13) = sG3;

genSeqFile( angDistDeg, 0, dy, dz, 'sG3' );

```

A.2: genSeqFile.m

```

function genSeqFile( angDeg, thG1, dy, dz, suffix )

fid = fopen(['imcWFE_', suffix, '.seq'],'w');

fprintf(fid, 'RDM\n');
fprintf(fid, 'LEN          "VERSION: 9.82 SR1          ');
fprintf(fid, 'LENS VERSION: 72          Creation Date: 22-Nov-2008"\n');
fprintf(fid, 'TITLE  ''JAPAN PATENT 61-48132, US PATENT 4264136''\n');
fprintf(fid, 'EPD    22.2744910585\n');
fprintf(fid, 'DIM    M\n');
fprintf(fid, 'WL     656.0 589.0 434.0\n');
fprintf(fid, 'REF    2\n');
fprintf(fid, 'WTW    1 1 1\n');
fprintf(fid, 'INI    ''EDF''\n');

fprintf(fid, 'XAN    0.0 0.0 0.0\n');

fprintf(fid, 'YAN    ');
fprintf(fid, '%8.6f ', angDeg);           % Field pos 0.0
fprintf(fid, '%8.6f ', 3.42 + angDeg); % Field pos 0.707
fprintf(fid, '%8.6f \n', 4.84 + angDeg); % Field pos 1.0

fprintf(fid, 'WTF    1.0 1.0 1.0\n');
fprintf(fid, 'VUX    0.0 0.00375008447266 0.00840859277344\n');
fprintf(fid, 'VLX    0.0 0.00375008447266 0.00840859277344\n');
fprintf(fid, 'VUY    0.0 0.16102590015 0.276740350947\n');

```

```

fprintf(fid, 'VLY    0.0 -0.0112381989716 -0.0135748956353\n');

% Object
fprintf(fid, 'SO      0.0 0.1e11\n');

thG1Deg = thG1 * 180 / pi;

% Surfaces
fprintf(fid, 'S1      60.572 2.0 517600.535000\n');
fprintf(fid, ' STO\n'); % Stop
fprintf(fid, 'S2      173.817 15.6\n');
fprintf(fid, 'S3      -29.825 1.6 516800.641000\n');
fprintf(fid, ' YDE S3 %8.6f\n', dy(3));
fprintf(fid, ' ZDE S3 %8.6f\n', dz(3));
fprintf(fid, ' ADE S3 %8.6f\n', thG1Deg);
fprintf(fid, ' DAR\n');
fprintf(fid, 'S4      -51.174 -1.6 REFL\n');
fprintf(fid, ' YDE S4 %8.6f\n', dy(4));
fprintf(fid, ' ZDE S4 %8.6f\n', dz(4));
fprintf(fid, ' ADE S4 %8.6f\n', thG1Deg);
fprintf(fid, ' DAR\n');
fprintf(fid, 'S5      -29.825 -13.6\n');
fprintf(fid, ' YDE S5 %8.6f\n', dy(5));
fprintf(fid, ' ZDE S5 %8.6f\n', dz(5));
fprintf(fid, ' ADE S5 %8.6f\n', thG1Deg);
fprintf(fid, ' DAR\n');
fprintf(fid, 'S6      -19.373 -1.0 516800.641000\n');
fprintf(fid, 'S7      -36.927 1.0 REFL\n');
fprintf(fid, 'S8      -19.373 13.6\n');
fprintf(fid, 'S9      -29.825 1.6 516800.549100\n');
fprintf(fid, ' YDE S9 %8.6f\n', dy(9));
fprintf(fid, ' ZDE S9 %8.6f\n', dz(9));
fprintf(fid, ' ADE S9 %8.6f\n', thG1Deg);
fprintf(fid, ' DAR\n');
fprintf(fid, 'S10     -51.174 1.2\n');
fprintf(fid, ' YDE S10 %8.6f\n', dy(10));
fprintf(fid, ' ZDE S10 %8.6f\n', dz(10));
fprintf(fid, ' ADE S10 %8.6f\n', thG1Deg);
fprintf(fid, ' DAR\n');
fprintf(fid, 'S11     -13.015 0.56 516800.819000\n');
fprintf(fid, ' YDE S11 %8.6f\n', dy(11));
fprintf(fid, ' DAR\n');
fprintf(fid, 'S12     266.647 0.76 740000.624300\n');
fprintf(fid, ' YDE S12 %8.6f\n', dy(12));
fprintf(fid, ' DAR\n');
fprintf(fid, 'S13     -77.979 17.0321811073\n');
fprintf(fid, ' YDE S13 %8.6f\n', dy(13));
fprintf(fid, ' DAR\n');

% Image
fprintf(fid, 'SI      0.0 0.09842981\n');
fprintf(fid, 'GO \n');

% Visualize
fprintf(fid, 'VIE\n');
fprintf(fid, ' LAB N\n');

```

```
%fprintf(fid, '  NBR SUR YES\n');
fprintf(fid, '  RFR NO\n');
fprintf(fid, '  RSI F1 W2 0  0.65\n');
fprintf(fid, '  RSI F1 W2 0 -0.71\n');
fprintf(fid, '  RSI F1 W2 0  1.00\n');
fprintf(fid, '  RSI F2 W2 0 -0.65\n');
fprintf(fid, '  RSI F2 W2 0  0.71\n');
fprintf(fid, '  RSI F2 W2 0 -1.00\n');
fprintf(fid, '  RSI F3 W2 0  0.8\n');
fprintf(fid, '  RSI F3 W2 0 -0.8\n');
fprintf(fid, '  RSI F3 W2 0  0.95\n');
fprintf(fid, 'GO\n');

% Real ray trace
fprintf(fid, 'RSI SO..I W2 F3 0 0.65; GO\n');
fprintf(fid, 'RSI SO..I W2 F2 0 0.65; GO\n');
fprintf(fid, 'RSI SO..I W2 F1 0 0.65; GO\n');

% Wavefront error
fprintf(fid, 'WAV; BES NO; NOM YES; THR NO; GO\n');

fclose(fid);
```