NGAO

Real Time Controller

Algorithms Design Document

Ver. 1.2

Revision History			
Version	Author	Date	Details
1.0	Marc Reinig and Donald Gavel	April 20, 2009	First draft version.
1.1	Donald Gavel	Nov 17, 2009	Updated
1.2	Donald Gavel	April 17, 2010	Updates in response to mini-review
1.3	Donald Gavel	May 30, 2010	Formatting and text adjustments for readibility

Table of Contents

1. Overview **1**
2. Algorithms (Mathematical Descriptions) **4**
2.1 Definition of terms 4
2.2 Centroiding 8
2.3 Wave Front Reconstruction 10
2.4 Tomography 12
2.5 DM Command Generation 15
2.6 Pseudo Open Loop Control 18
2.7 Split Tomography 19
3. Algorithm Interpretation **21**
3.1 Spatial Domain Interpretation 21
3.2 Fourier Domain Interpretation 25
4. Requirements Flow Down **27**
4.1 Speed of Calculations 27
4.1.1 High-order controller 28
4.1.2 Tip/Tilt Controller – Science Path 28
4.1.3 Tip/Tilt Controllers – HOWFS Paths 29
4.1.4 Tip/Tilt Controllers – LOWFS Paths 29
4.2 Accuracy of Calculations 30
4.3 Interface Data 31
4.3.1 Cameras 31
4.3.2 DMs 31
4.3.3 Diagnostic and Telemetry Data 31
4.3.4 Offloading 31
4.4 Calibration 32
4.5 Parametric Data 32
5. References **33**
Appendix A: Table of symbols used in this document **34**

Table of Figures

Figure 1. Real-Time Controller Block Diagram 3
Figure 2. Geometry of ray projection from the k'th guide star. 5
Figure 3. Centroiding cell weights for center of mass calculation. Left: x weights, Right, y weights 9
Figure 4. Centroiding pixel weights for binned quad cell calculation. Left: x weights, Right, y weights 9
Figure 5. Block diagram of the algorithm for wavefront reconstruction 12
Figure 6. Block diagram of inverse tomography algorithm 15
Figure 7. Block diagram of woofer / tweeter split 17
Figure 8. Split Tomography block diagram, showing also the woofer/tweeter split 20

1. Overview

The real-time controller (RTC) has the overall task of taking raw camera data from multiple wavefront sensors and determining the commands to place on multiple wavefront correctors. The input camera data is in the form of data numbers (DN) as read by a camera hardware controllers and the output data is in the form of voltages to be sent to D/A converters that drive the deformable mirrors (DM) and tip/tilt mirrors (TTM).

The processing of data and creation of commands is to happen within time-slices determined by the wavefront sensor frame sample period, t_s, with the wavefront sensors acting as the master reference clock on the cycle. This clock is used as an interrupt drive for the RTC and all the operations of the RTC are thus synchronized to this clock.We will assume that for NGAO, that all high order wavefront sensors (HOWFS) run on the same cycle period, and that the low-order wavefront sensors (LOWFS) run at some integral multiple of this period. For NGAO, we have nominally set this period at t_s=500 μs, or f_s=2kHz.

Assuming the RTC hardware accepts the data and places it in the appropriate portions of memory, the RTC algorithm must perform the following activities:

- Process the raw camera data so that its values are proportional to photocounts
- Perform a reconstruction process that estimates the wavefront correction needed for the science path correction and for correction of the tip/tilt stars.
- Allocate the corrections to woofer and tweeter DMs.
- Invoke a dynamic compensator, such as integral feedback control, to stabilize the closed-loop path (involving the woofer DM)
- Invoke a non-linearity compensation for all of the DMs, to convert wavefront phase to command voltages.
- Process low-order (tip/tilt/focus/astigmatism) data from the LOWFS to determine tip/tilt for the science path and for the tip/tilt paths.

The present NGAO architecture is based on the use of "point-and-shoot" guidestars. In this architecture, 4 of the 7 laser guidestars (LGS) are arranged in a fixed asterism for tomography over a ~40 arcsec field of view (FoV) volume above the on-axis science path, while the remaining 3 roving LGS are pointed individually to the tip/tilt stars. The high-order wavefront control is separate for each tip/tilt star, based on its assigned LGS and HOWFS, and the tip/tilt correction of the tip/tilt star is determined in closed loop using its own assigned LOWFS. The tip/tilt signal is a combination of the closed-loop LOWFS tip/tilt command and its measured tip/tilt residual. All of the tip/tilt, as well as the one focus/astigmatism, sensor's signals are sent to the RTC for processing in order to determine the tip/tilt command to send to the science path fast-steering stage.

The algorithms implementing the above processes are operated in a parallel processing machine architecture (see Figure 1). The machine architecture is designed specifically to take advantage of the algorithms' inherent parallelism. Operations in parallel include the

independent point-and-shoot wavefront corrections, the independent camera data processing from multiple wavefront sensors and the multiple DM nonlinearity compensation. In addition, a great deal of parallelism is achieved in the inverse volume tomography algorithm by implementing it in the Fourier domain. The Fourier domain method is an approximate inverse, called a preconditioner, which must be iterated to convergence. Thus we have had to assure that time spent iterating don't defeat the parallelism benefit of the Fourier domain implementation. Our analyses have show that wavefront accuracy requirements for NGAO are achieved with one iteration per wavefront measurement cycle. Thus we take full advantage of the Fourier domain processing.

This document provides the mathematical description and justification of each of the algorithmic steps in the RTC processing. Although the algorithms are tuned to be implemented in a fast, massively parallel machine architecture, there are no approximations in the implementation of minimum variance control except for spatial and temporal sampling, and digitization. The wavefront reconstruction and tomography algorithms produce the unique minimum variance solutions taking in to account a-priori statistics and measurement noise statistics. To double-check the correctness of calculations, an RTC simulator has been coded so that results can be compared against a "traditional" matrix-multiply invocation of the minimum-variance computation. The RTC simulator will also later serve as a validation tool for use during RTC hardware/software integration and test.

The spatial and temporal sampling and the digitization (finite length machine word) are design choices. We have analyzed these and established their design values so as to meet the overall error budget.

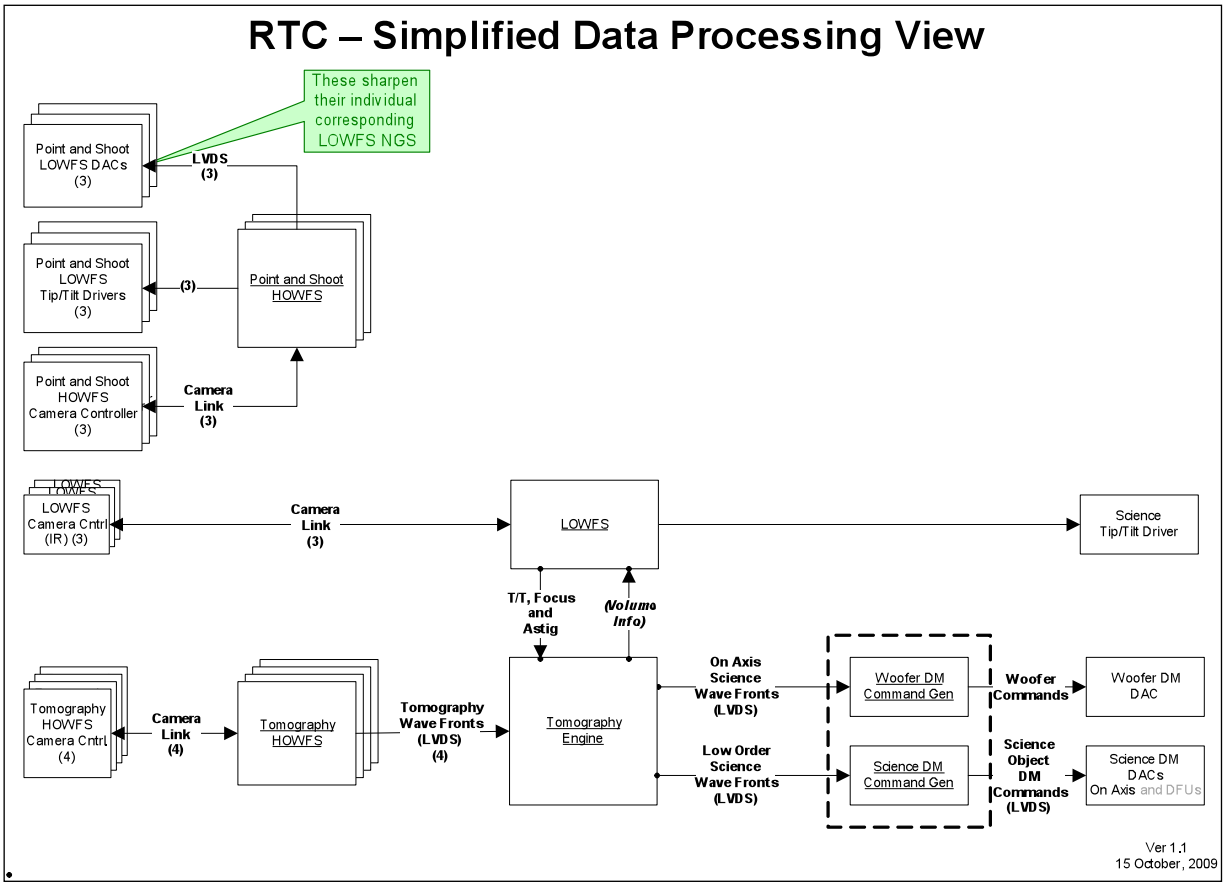

Figure 1. Real-Time Controller Block Diagram

2. Algorithms (Mathematical Descriptions)

2.1 Definition of terms

We start with a general description of the wavefront formation from guidestars and turbulent atmosphere. We then explain the nonlinearities involved with the wavefront sensing and wavefront control. Starting from these basic models and definitions, the subsequent sections will describe each portion of the RTC's algorithm set.

The common accepted practice in today's astronomical AO systems is to assume that the wavefront formation process is linear to first order, i.e. that there is a linear relationship between delta index perturbations in the volume of atmosphere above the telescope and the wavefront phase as it enters the telescope aperture. The assumption is valid for "weak" atmospheric turbulence, where we can assume that the optical path distance (OPD) of a ray traversing through the atmosphere departs from that of a vacuum path according to a line-integral of the delta-index of the air along that straight path. The weak turbulence assumption is valid over scales that are larger than the Fresnel zone, a transverse length scale associated with ray crossing at the ground due to higher altitude phase aberrations or, equivalently, valid for spatial frequency amplitude variations where the associated Talbot length is much longer that the atmospheric path. Hardy [(1)] discusses these quantities for Earth's atmosphere. For the NGAO case, where subaperture sizes, mapped to the primary, are greater than or equal to 15 cm and we are using the baseline C_n^2 profile for design purposes, the weak conditions apply with high accuracy. This is not to say that scintillation in the Hartmann wavefront sensing is negligible. This scintillation is an important contributor to the error budget, but is tolerable; hence the baseline RTC algorithms are not designed to utilize scintillation information to aid in wavefront reconstruction.

Given that the ray-traces are line integrals, we approximate them as a Riemann sum:

$$\begin{aligned}\phi_k(\vec{\mathbf{u}}) &= \left(\frac{2\pi}{\lambda}\right)\int_0^{z_k} \delta\eta\left(\alpha_k(z)\vec{\mathbf{u}} + \vec{\theta}_k z, z\right) dz \\ &\cong \sum_{l=0}^{L-1} \delta\phi\left(\alpha_{kl}\vec{\mathbf{u}} + \vec{\theta}_k z_l, z_l\right)\end{aligned} \tag{1}$$

where $\phi_k(\vec{\mathbf{u}})$ are the wavefront phases at the ground, at transverse positions on the telescope aperture $\vec{\mathbf{u}}$ from a guidestar in direction $\vec{\theta}_k$ and at distance z_k (sodium LGS are located at 90 km x sec(ζ) where ζ is the zenith pointing angle of the telescope, and natural stars are at $z_k = \infty$), $\delta\eta\left(\alpha_k(z)\vec{\mathbf{u}} - \vec{\theta}_k z, z\right)$ are the index of refraction changes due to atmospheric turbulence at altitude *z* along the line $\vec{\theta}_k$, $\alpha_k(z) = 1 - z/z_k$ is the scale factor associated with a cone-beam projection, λ is the wavelength of light for which the phase is defined, and L is the number of discrete layers, indexed by *l*, in the layered approximation.

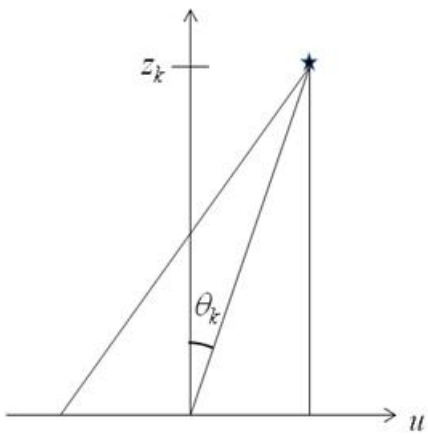

Figure 2. Geometry of ray projection from the k'th guide star.

Line up all the phase points (sampling in $\vec{\mathbf{u}}$-space) for all the wavefront sensors (indexed by k) and put them in a single vector, **y**. The number of elements in y is equal to the grand total count of all the subapertures in the system. Similarly, line up all the delta index-of-refractions over all the volume into one single vector, **x**. Note, the later are sampled in both $\vec{\mathbf{u}}$ (transverse) and z (longitudinal) space. With these definitions, the Riemann sum integral (1) can be written as a linear matrix equation

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{2}$$

Where **A** is a matrix that maps every volume element to every sample on every wavefront sensor. The **A** matrix can be thought of as consisting of 1's and 0's (with a wavelength dependent factor in front to convert OPD to phase), however, with the discrete transverse sampling at each layer the lines do not necessarily pierce the layers at discrete index points, so the entries in **A** are the appropriate interpolation coefficients.

Hartmann wavefront sensors measure subaperture-averaged slopes of wavefronts. Again, we can use a Riemann sum to generate a discrete approximation to the averaged gradient operator.

$$\begin{aligned} \vec{s}_k\left(\vec{\mathbf{u}}_j\right) &= \frac{1}{A_j} \int_{\text{subap } j} \vec{\nabla}\phi_k\left(\vec{\mathbf{u}}\right) d\vec{\mathbf{u}} \\ &\cong \sum_{j' \in \text{subap } j} g_{j,j'} \phi_k\left(\vec{\mathbf{u}}_{j'}\right) \end{aligned} \tag{3}$$

where $\vec{\mathbf{u}}_j$ represents each discrete position (indexed by j) of subapertures in the aperture, and A_j is the area of that subaperture. Note that the Riemann sum allows for sub-sampling of the phase within a subaperture. Each Hartmann slope measurement is a 2-vector for each location, one for the x slope and one for the y slope.

Line up all the phase points (again, sampled in $\vec{\mathbf{u}}$-space) for all the wavefront sensors into one vector, **y**. Line up all the discrete slope measurements into one vector, **s**. Then the Hartmann wavefront sensing operation is denoted as

$$\mathbf{s} = \mathbf{G}\mathbf{y} \tag{4}$$

That is a linear operation (**s** is linearly proportional to **y**).

A note on the storage in the vector **s**: It is arbitrary how the sensor data are arranged in s (so long as the rows of **G** are correspondingly arranged). We adopt the following convention. Each wavefront sensor has a contiguous block of elements in **s**. Within each block assigned to a wavefront sensor, the x slopes are arranged in raster-scan order, then the y slopes.

We should also distinguish whether the elements of **s** and **y** include every grid location, or just those enclosed in the aperture and illuminated by the guidestar. This is an important distinction for a reconstruction algorithm that utilizes Fourier domain techniques, since Fourier transforms need points to be on a rectangular grid and on a rectangular shaped domain. A similar consideration might apply for the elements of the **x** vector, whether or not it includes only volume elements pierced by guidestar rays or fills a rectanguloid shaped volume.

We adopt the following convention: **G** represents the matrix operation in (3) on a larger, aperture-enclosing square grid of subapertures. $\mathbf{G}_P$ represents a matrix with reduced row dimension that includes only illuminated subapertures within the telescope aperture. The pupil operator **P** is a dimension-reducing matrix that picks data (or matrix rows) that are inside the pupil. Thus

$$\mathbf{s}_P = \mathbf{P}\mathbf{s} \text{ and } \mathbf{G}_P = \mathbf{P}\mathbf{G} \tag{5}$$

The transpose of the pupil operator, $\mathbf{P}^T$ is a dimension-increasing operation that takes a vector consisting of data only from inside the pupil and produces a data vector with the data inside the pupil unchanged and entries for positions outside the pupil set to zero on an aperture-enclosing square grid. $\mathbf{P}^T\mathbf{P}$ is a masking operation that modifies data on a square grid by setting all the elements outside the pupil to zero.

The matrix **A** and vector **x** will not need to have this a distinction. They will always represent the full 3-d rectanguloid volume.

For the Fourier domain algorithm discriptions, it is convenient to clearly designate whether a certain operation occurs in the spatial domain or in the spatial frequency (Fourier) domain. To do this, we introduce the Fourier transform operator **F**. The elements of **F** are:

$$F_{fj} = \exp\left\{ -i2\pi\vec{\mathbf{u}}_j \cdot \vec{\mathbf{f}}_f \right\} \tag{6}$$

where $\vec{\mathbf{f}}_f$ is the spatial frequency, indexed by *f*. Note that

$$\mathbf{F}^{-1} = \frac{1}{N}\mathbf{F}^* \tag{7}$$

where *N* is the total number of sample points in the square grid. The Fourier transform of a vector of raster-scanned data is

$$\tilde{\mathbf{y}} = \mathbf{F}\mathbf{y} \tag{8}$$

We can also define the Fourier transform of a matrix so that it is consistent with matrix operations on vectors of data:

$$\tilde{\mathbf{G}} = \mathbf{F}\mathbf{G}\mathbf{F}^{-1} \tag{9}$$

The over-tilde designates the Fourier transform of an item. The Fourier transform can only be applied over aperture-enclosing square grids, thus it cannot be applied to *P* subscripted items directly (which consist only of data inside the aperture), but it can be applied to items that are inserted in to a square grid using the $\mathbf{P}^T$ operator. So we define

$$\tilde{\mathbf{s}}_P = \mathbf{F}\mathbf{P}^T\mathbf{s}_P \tag{10}$$

For tip/tilt sensors, the sensing process involves propagating to the far-field and determining the center of mass of the resulting far-field point spread function (PSF). Again, assuming the weak turbulence hypothesis, the center of mass shift is linearly proportional to the average wavefront tilt. This is not to say that there aren't significant errors in this assumption, i.e. the difference between "Z" tilt and "G" tilt is well documented [Ref Tyler]. In particular, an aberrated wavefront will have non-symmetric Zernike aberrations other than tip and tilt, such as coma, that will cause a further shift of the far-field PSF center of mass. In the NGAO case, this is considerably mitigated by the fact that the tip/tilt stars' wavefronts are partially corrected through the use of dedicated deformable mirrors, and by the woofer mirror, in the baseline point-and-shoot architecture. The resulting differences between Z and G tilt are then considered tolerable, and included in the error budget.

The incorporation of tip/tilt data into the wavefront control for the science beam is treated in the section titled Split Tomography.

Each wavefront sensor and wavefront corrector has associated with it certain nonlinearities as well as grid and alignment distortions. These effects are very significant and so must be incorporated in the data formulation model, and accommodated by the RTC.

A wavefront sensor nonlinearity is of a form that maps actual Hartmann centroid shifts to reported centroid shifts:

$$\mathbf{c} = f_H(\mathbf{s}) \tag{11}$$

Generally, we will treat each subaperture as having an independent nonlinearity, however, there may be a cross-dependence between the x and y channels in a given subaperture.

A deformable mirror has, in general, complete non-linear cross-dependence of the surface response of the mirror to the voltages given to its actuators. We adopt the conventions presented in [Reference Morzinski and Gavel] where we presume shift-invariance, a linear plate-equation model of the deflection of the surface in response to forces, and arbitrary dependence of a given actuator's force as a function of voltage, but with no cross-coupling of actuators except through forces exerted by the top plate. The resulting formulation is:

$$\begin{aligned} f_p(\vec{\mathbf{u}}) &= \sum_j \left[f_e\left(V_j, \xi_j\right) + f_s\left(\xi_j\right) \right] \delta\left(\vec{\mathbf{u}} - \vec{\mathbf{u}}_j\right) \\ \nabla^4 \xi(\vec{\mathbf{u}}) &= f_p(\vec{\mathbf{u}}) \end{aligned} \tag{12}$$

where $f_p(\vec{\mathbf{u}})$ is the plate restorative force, $f_e\left(V_j, \xi_j\right)$ is the electrostatic deflection force of the *j*'th actuator, $f_s\left(\xi_j\right)$ is the spring return force of the *j*'th actuator, ξ_j is the

deflection of the mirror surface at the actuator post location, $\xi(\vec{\mathbf{u}})$ is the deflection of the mirror surface over the contiguous face sheet, V_j is the voltage applied to the *j*'th actuator and $\check{\mathbf{u}}_j$ is the location of the *j*'th actuator's post attachment to the face sheet. This model is not perfect, but it has been proven accurate to approximately 15 nm surface peak-to-valley on a Boston Micromachines MEMS deformable mirror. The error is incorporated in the error budget when applied to open-loop controlled DMs. It is not a critical issue for the woofer DM, which is controlled closed loop. The baseline woofer DM is a Cilas mirror that uses the latest technology piezo actuators having very little hysteresis. For the woofer, we may adopt the plate-equation and nonlinear actuator model (6), or a simpler linear superposition model:

$$\xi(\vec{\mathbf{u}}) = \sum_j d_j(V_j) r_j(\vec{\mathbf{u}} - \vec{\mathbf{u}}_j) \tag{13}$$

where $r_j(\vec{\mathbf{u}} - \vec{\mathbf{u}}_j)$ is a mirror displacement influence function and $d_j(V_j)$ is a non-linear voltage to displacement relation of the *j*'th actuator.

The grid and alignment distortion will, by convention, be referenced to a common coordinate system, which we arbitrarily attach to the 0 km conjugate optical plane at the location of the woofer DM. Even the woofer however will need a distortion grid map because the incident angle causes a foreshortening of one axis with respect to the other. We use the centered, symmetric Keck pupil to set the scale in x and y so that, nominally, Δx = Δy in SI units (meters) on the Keck primary. Each of the deformable mirror actuator grids and each of the wavefront sensor subaperture grids must be mapped to this common grid. The RTC must use distortion, rotation, and displacement coefficients to map wavefronts internally represented in the common coordinate system to subaperture and actuator locations, as they are truly located in the optical system. Transformations will be of the form

$$\vec{\mathbf{u}}'_k = R_k(\vec{\mathbf{u}}_0) \tag{14}$$

where $\vec{\mathbf{u}}'_k$ is the coordinate system set by the respective DM actuator pitch or WFS subaperture pitch (DM or WFS #*k*) and $\vec{\mathbf{u}}_0$ is the common coordinate system. The distortion mapping is anticipated to be a relatively small perturbation that can be implemented in the RTC through localized interpolations. The algorithm is described in the Centroiding and DM Command Generation sections.

2.2 Centroiding

The Hartman camera data consists of an array of pixel readouts in the form of data numbers (DN) provided the camera hardware controllers. The first step consists of removing the bias and background from these arrays and then applying a flat-fielding correction, which is a scale factor.

The next step is take the subaperture regions of interest and apply the centroiding algorithm. The algorithm takes the form

$$c_{xj} = \sum_{p \in P_j} w_{xp} I_p$$
$$c_{yj} = \sum_{p \in P_j} w_{yp} I_p \tag{15}$$

where c_{xj} and c_{yj} are the centroids, w_{xp} and w_{yp} are the centroider weights, I_p is the intensity in the p'th pixel, and P_j is the region of the detector surface assigned to the j'th subaperture. The weights are arbitrary and can be set by the AO operator to implement any number of weighted centroiding algorithms. The following figure shows a couple of common algorithm settings for a 4x4 pixel subaperture.

-1.5	-0.5	0.5	1.5
-1.5	-0.5	0.5	1.5
-1.5	-0.5	0.5	1.5
-1.5	-0.5	0.5	1.5

1.5	1.5	1.5	1.5
0.5	0.5	0.5	0.5
-0.5	-0.5	-0.5	-0.5
-1.5	-1.5	-1.5	-1.5

Figure 3. Centroiding cell weights for center of mass calculation. Left: x weights, Right, y weights

-1	-1	1	1
-1	-1	1	1
-1	-1	1	1
-1	-1	1	1

1	1	1	1
1	1	1	1
-1	-1	-1	-1
-1	-1	-1	-1

Figure 4. Centroiding pixel weights for binned quad cell calculation. Left: x weights, Right, y weights

After finding the centroids, the inverse of the Hartmann sensor nonlinearity (5) must be applied. This is a 2-input 2-output lookup table.

Finally, the distortion grid correction is applied to map the slopes sampled on the distorted grid to a sampling on a common grid. This is accomplished using local interpolation

$$s_j = \sum_{j' \in \text{local}} c_{jj'} s_{j'} \tag{16}$$

where $s_{j'} = s(\mathbf{u}_{j'})$ represents the slope (either x or y) as computed on the distorted grid of the sensor, $s_j = s(\mathbf{u}_j)$ represents the interpolated slope on the common coordinate system, and $c_{jj'}$ are interpolation coefficients. The interpolation coefficients are non-zero on a

local region, $\left|\mathbf{u}_j - \mathbf{u}_{j'}\right| < \rho$ containing only a few discrete locations (4, 9, or 16 nearest neighbors) for computational efficiency, under the assumption that the distortions are small relative to grid spacing.

2.3 Wave Front Reconstruction

Wavefront reconstruction is the process of converting surface slopes to surface heights, under the assumption that the surface is continuous. In rough terms (for illustrative purposes only, do not implement) one technique would be to take the numerical divergence of the slopes, which is equal to the Laplacian of the phases, then numerically invert the Laplacian operator. Another technique (again for illustrative purposes, do not implement) is to take the pseudo-inverse of the G matrix in (4): $\mathbf{y} = \left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{x}$. In fact, although this is the basic idea, there must be considerable modification because of several difficulties:

- The noise statistics need to be taken into account to form a minimum variance estimate
- The measurement data has a boundary, the edge of the aperture. Solutions must take in to account the discontinuity at this boundary to prevent severe error.
- The matrix G is not full rank. Piston, and, for some geometries, waffle mode, are in the null space. Thus the pseudo-inverse cannot be formulated as shown.

The approach in the NGAO reconstructor is to solve for the conditional mean phase given the slope data confined to the aperture.

$$\left\langle \mathbf{y} \middle| \mathbf{s}_P \right\rangle = \left\langle \mathbf{y}\mathbf{s}_P^T \right\rangle \left\langle \mathbf{s}_P\mathbf{s}_P^T \right\rangle^{-1} \mathbf{s}_P \tag{17}$$

where $\langle \bullet \rangle$ is the expectation operator, which when applied to the outer product of vectors is the cross-covariance of the vectors. In matrix form

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{S}_y\mathbf{G}_P^T\left(\mathbf{G}_P\mathbf{S}_y\mathbf{G}_P^T + \mathbf{S}_n\right)^{-1}\mathbf{s}_p \\ &= \left(\sigma_n^2\mathbf{S}_y^{-1} + \mathbf{G}_P^T\mathbf{G}_P\right)^{-1}\mathbf{G}_P^T\mathbf{s}_p \end{aligned} \tag{18}$$

where $\hat{\mathbf{y}} = \left\langle \mathbf{y} \middle| \mathbf{s}_P \right\rangle, \mathbf{S}_y = \left\langle \mathbf{y}\mathbf{y}^T \right\rangle$, $\mathbf{S}_n = \left\langle \mathbf{n}\mathbf{n}^T \right\rangle = \sigma_n^2\mathbf{I}$, $\mathbf{n}$ is the measurement noise, and σ_n is the standard deviation of the measurement noise, which is assumed to be independent identically distributed Gaussian white noise for each channel of the Hartmann sensor. The second line of (18) is a result of applying the Matrix Inversion Lemma, also known as the Sherman-Morrison Theorem. We discuss how to specify the covariance matrices in section 4.5. For now, all matrices are treated as pre-specified for the purpose of describing the real-time computations. Occasionally, the covariance matrices may be changed by the supervisory controller, for example to reflect changes in seeing conditions.

Equation (18) (first or second form) must be implemented by the wavefront reconstructor section of the RTC. We now show how we plan to implement (18) using massively parallel computer architecture.

Using the second form of (18), transform to the Fourier domain:

$$\begin{aligned}\tilde{\hat{\mathbf{y}}} &= \mathbf{F}\left(\sigma_n^2\mathbf{S}_y^{-1} + \mathbf{G}_P^T\mathbf{G}_P\right)^{-1}\mathbf{G}_P^T\mathbf{s}_p \\ &= \left[\mathbf{F}\left(\sigma_n^2\mathbf{S}_y^{-1} + \mathbf{G}_P^T\mathbf{G}_P\right)\mathbf{F}^{-1}\right]^{-1}\mathbf{F}\,\mathbf{G}_P^T\mathbf{s}_P \\ &= \left(\sigma_n^2\tilde{\mathbf{S}}_y^{-1} + \tilde{\mathbf{G}}^H\mathbf{F}\mathbf{P}^T\mathbf{P}\mathbf{F}^{-1}\tilde{\mathbf{G}}\right)^{-1}\tilde{\mathbf{G}}^H\tilde{\mathbf{s}}_p\end{aligned} \tag{19}$$

where the superscript H denotes the Hermetian transpose (complex-conjugate of the transpose). Recall from our earlier definitions (equation (10)) that $\tilde{\mathbf{s}}_p$ is the slope data inserted in the aperture-enclosing square grid with zeros outside the pupil, then Fourier-transformed.

The advantage of using the Fourier domain is that matrices $\tilde{\mathbf{S}}_y^{-1}$, $\tilde{\mathbf{G}}$, and $\tilde{\mathbf{G}}^H$ are diagonal with

$$\left[\tilde{\mathbf{G}}\right]_{ff} = \begin{bmatrix} -i2\pi f_x \\ -i2\pi f_y \end{bmatrix} \quad \text{and} \quad \left[\tilde{\mathbf{G}}^H\right]_{ff} = \begin{bmatrix} i2\pi f_x & i2\pi f_y \end{bmatrix} \tag{20}$$

($\tilde{\mathbf{S}}_y^{-1}$ is defined in the section on parameters, below).

If it weren't for the nested pupil-mask inside the second term of the inverted matrix, this matrix too would be diagonal. Then equation (19) would become N independent scalar equations, amenable to an N-fold parallelization of its calculation.

Instead we use the iterative algorithm:

$$\tilde{\mathbf{c}} = \tilde{\mathbf{G}}^H\tilde{\mathbf{s}}_P \tag{21}$$

$$\text{iterate}\left\{\begin{aligned}\tilde{\varepsilon}^k &= \tilde{\mathbf{c}} - \left(\sigma_n^2\tilde{\mathbf{S}}_y^{-1} + \tilde{\mathbf{G}}^H\mathbf{F}\,\mathbf{P}^T\mathbf{P}\mathbf{F}^{-1}\tilde{\mathbf{G}}\right)\tilde{\hat{\mathbf{y}}}^k \\ \tilde{\hat{\mathbf{y}}}^{k+1} &= \tilde{\hat{\mathbf{y}}}^k + \gamma\,\tilde{\mathbf{C}}\tilde{\varepsilon}^k\end{aligned}\right\}\text{until } \tilde{\varepsilon}^k \sim 0 \tag{22}$$

where

$$\tilde{\mathbf{C}} = \left(\sigma_n^2\tilde{\mathbf{S}}_y^{-1} + \tilde{\mathbf{G}}^H\tilde{\mathbf{G}}\right)^{-1} \tag{23}$$

and γ is a positive gain-iteration factor (generally $\gamma \approx 0.5$).

$\tilde{\mathbf{C}}$ is a diagonal approximation of the needed matrix inverse and is referred to as the Fourier-preconditioner. The elements on the diagonal are

$$\tilde{C}(\mathbf{f}) = 1\Big/\left(\sigma_n^2\tilde{S}_y^{-1}(\mathbf{f}) + 4\pi^2|\mathbf{f}|^2\right) \tag{24}$$

Algorithm (22) operates in parallel across frequency domain components and independently for each wavefront sensor until the multiplication by $\mathbf{P}^T\,\mathbf{P}$ is required. This requires a transform into the spatial domain to apply the aperture mask, then a transform back to the Fourier domain to continue on with the iteration loop.

The flow diagram for the iteration process for a single wavefront sensor is shown in **Figure 5**. The computer hardware to do this is simply replicated one per wavefront sensor, so that all wavefront sensors are processed in parallel. Most of the operations in

the loop are multiply-accumulates of a single data number by a constant, one per spatial frequency element. This is done with a massive array of small processors, called compute elements, each dedicated to a single spatial frequency and all operating in parallel. A Fourier transform pair is required in the multiplication by $\widetilde{\mathbf{M}}$. The RTC will accomplish this using a fast algorithm tuned to the massive parallel processing architecture described in Appendix H of the RTC Design Document [2].

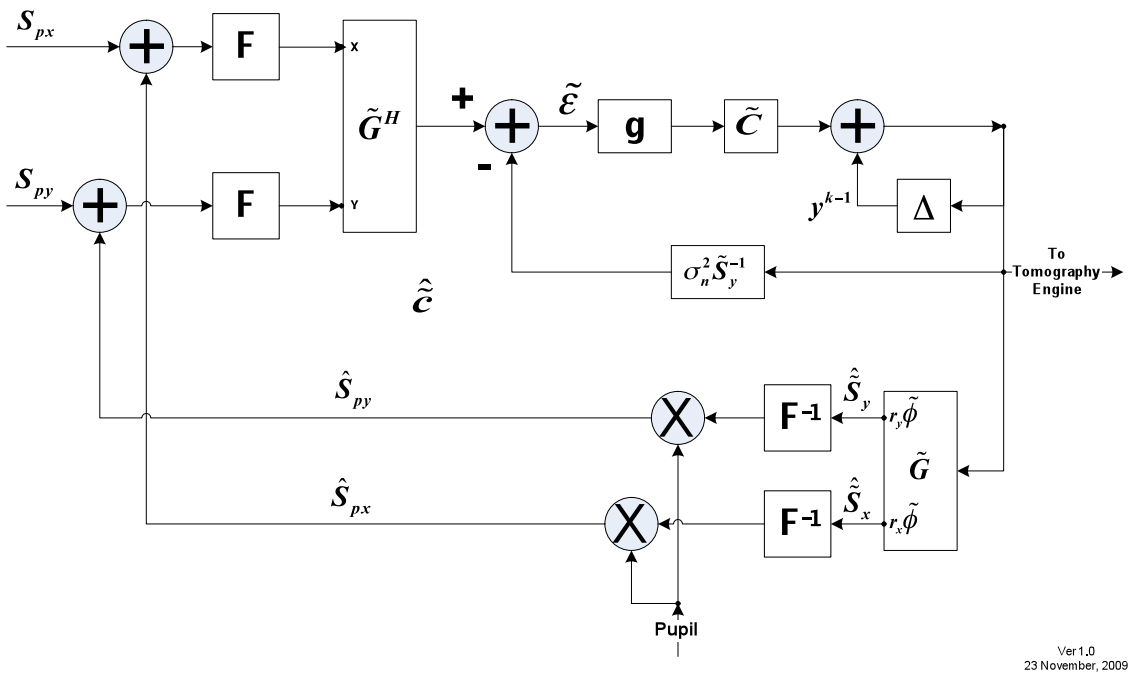

Figure 5. Block diagram of the algorithm for wavefront reconstruction

The iterative algorithm (22) is proven to be stably convergent [3] and our simulation experience has shown that about 30 iterations are necessary to converge to under 10 nm rms in r_0 = 15 cm seeing conditions, starting with $\tilde{\hat{\mathbf{y}}}^0 = 0$. We advocate the use of a "warm-restart" in the RTC, where the phase estimate from the prior data time step is the starting point for the current time step, i.e.

$$\tilde{\hat{\mathbf{y}}}^0(t) = \tilde{\hat{\mathbf{y}}}^K(t-1) \tag{25}$$

where K is the last iteration at time step t-1. In the moderate baseline case wind conditions for NGAO, 10 m/sec, and 1 kHz sample rates, our simulations using warm restart and K=1 (one iteration per time step) typically converge to the 10 nm level in less than 100 ms.

2.4 Tomography

In tomography, the key operations are the forward propagation **A** and the back-propagation, $\mathbf{A}^T$.

The **A** matrix is of dimension $NM \times NL$ where N is the number of grid points in the aperture-enclosing square grid of data points, M is the number of wavefront sensors, and L is the number of atmospheric layers. For later convenience, we arrange **A** as an $N \times N$ array of $M \times L$ blocks. The elements of **A** are (conceptually, ignoring the issue of interpolation coefficients when **u** does not land on a sample grid point):

$$A_{lk}\left(\mathbf{u}_0,\mathbf{u}_l\right) = \delta\left(\mathbf{u}_0,\left(\mathbf{u}_l - \theta_k z_l\right)/\alpha_{kl}\right) \tag{26}$$

where $\delta(\cdot,\cdot)$ is the Kroneker delta: =1 if the two index arguments are equal, 0 otherwise.

In the Fourier domain the elements of $\widetilde{\mathbf{A}}$ are:

$$\tilde{A}_{lk}\left(\mathbf{f}_0,\mathbf{f}_l\right) = \exp\left\{-i2\pi\mathbf{f}_l\cdot\theta_k z_l\right\}\delta\left(\mathbf{f}_0,\mathbf{f}_l/\alpha_{lk}\right) \tag{27}$$

In either domain, the output domain map is stretched to conform on the cone beam. In the spatial domain, the points at an altitude are stretched apart when mapped to the ground. In the frequency domain, the spatial frequencies at an altitude are shrunk to lower frequencies on the ground.

The back propagation operation $\mathbf{A}^T$ in the spatial domain is

$$A^T_{lk}\left(\mathbf{u}_l,\mathbf{u}_0\right) = \delta\left(\mathbf{u}_l, \alpha_{kl}\mathbf{u}_0 + \theta_k z_l\right) \tag{28}$$

and, in the frequency domain is

$$\tilde{A}^T_{lk}\left(\mathbf{f}_l,\mathbf{f}_0\right) = \exp\left\{i2\pi\mathbf{f}_l\cdot\theta_k z_l\right\}\delta\left(\mathbf{f}_l, \alpha_{lk}\mathbf{f}_0\right) \tag{29}$$

Thus, in a similar manner similar to the forward propagator, the back propagation operator's output map is stretched to conform to the cone beam, but now in the opposite sense.

To compute a minimum-variance tomography solution, we utilize the chain rule of conditional expected values [(3)] to connect the reconstructed phase estimates from algorithm (21) for each wavefront sensor to a volume estimate of turbulence:

$$\left\langle \mathbf{x}(t) \,\middle|\, \mathbf{s}_P \right\rangle = \left\langle \mathbf{x}(t-1) \right\rangle + \left\langle \mathbf{x}\mathbf{e}_P^T \right\rangle \left\langle \mathbf{e}_P\mathbf{e}_P^T \right\rangle^{-1} \left(\left\langle \mathbf{y}_P \,\middle|\, \mathbf{s}_P \right\rangle + \varphi_W - \mathbf{A}_P\hat{\mathbf{x}}(t-1) \right) \tag{30}$$

where $\left\langle \mathbf{x}(t-1) \right\rangle$ is the prior estimate of the volume and φ_W is the phase on the woofer deformable mirror. In the NGAO hybrid closed-loop open-loop architecture, the woofer DM is upstream of the wavefront sensors, subtracting its phase optically, thus we need to add it back in numerically.

In matrix form the estimation formula is:

$$\begin{aligned} \hat{\mathbf{x}}(t) &= \hat{\mathbf{x}}(t-1) + \mathbf{K}\mathbf{P}\left(\hat{\mathbf{y}}(t) + \varphi_W - \mathbf{A}\hat{\mathbf{x}}(t-1)\right) \\ \mathbf{K} &= \mathbf{S}_x\mathbf{A}_P^T\left(\mathbf{A}_P\mathbf{S}_x\mathbf{A}_P^T + \mathbf{S}_{n_y}\right)^{-1} \end{aligned} \tag{31}$$

where $\hat{\mathbf{x}} = \langle \mathbf{x} | \mathbf{s}_P \rangle$, $\mathbf{S}_x = \left\langle \left(\mathbf{x} - \hat{\mathbf{x}}(t-1) \right)\left(\mathbf{x} - \hat{\mathbf{x}}(t-1) \right)^T \right\rangle$ the covariance of the error in the current volume turbulence estimate and $\mathbf{S}_{n_y} = \left\langle \left(\mathbf{y}_P - \langle \mathbf{y}_P | \mathbf{s}_P \rangle \right)\left(\mathbf{y}_P - \langle \mathbf{y}_P | \mathbf{s}_P \rangle \right)^T \right\rangle$ the covariance of the error in the reconstructed phases in the wavefront sensors.

Assuming we can only approximate the matrix inverse by **Q**, the iterative reconstructor is

$$\begin{aligned} &\mathbf{e}_P(t) = \mathbf{P}\left(\hat{\mathbf{y}} + \varphi_W - \mathbf{A}\hat{\mathbf{x}}(t-1) \right) \\ &\text{iterate}\begin{cases} \mathbf{r}_P^k = \mathbf{e}_p(t) - \left(\mathbf{A}_P \mathbf{S}_x \mathbf{A}_P^T + \mathbf{S}_{n_y} \right)\mathbf{w}_P^k \\ \mathbf{w}_P^{k+1} = \mathbf{w}_P^k + \gamma\, \mathbf{Q}\mathbf{r}_P^k \end{cases} \text{until } \mathbf{r}^k \sim 0 \\ &\hat{\mathbf{x}}(t) = \hat{\mathbf{x}}(t-1) + \mathbf{S}_x \mathbf{A}_P^T \mathbf{w}_P^K \end{aligned} \tag{32}$$

Transforming this to the Fourier domain

$$\begin{aligned} &\tilde{\mathbf{e}}(t) = \tilde{\hat{\mathbf{y}}} + \tilde{\varphi}_W - \tilde{\mathbf{A}}\,\tilde{\hat{\mathbf{x}}}(t-1) \\ &\text{iterate}\begin{cases} \tilde{\mathbf{r}}^k = \tilde{\mathbf{e}}(t) - \left(\tilde{\mathbf{A}}\tilde{\mathbf{S}}_x \tilde{\mathbf{A}}^H + \tilde{\mathbf{S}}_{n_y} \right)\tilde{\mathbf{w}}_P^k \\ \tilde{\mathbf{w}}_P^{k+1} = \tilde{\mathbf{w}}_P^k + \gamma\, \mathbf{F}\mathbf{P}^T\mathbf{P}\mathbf{F}^{-1}\tilde{\mathbf{Q}}\mathbf{F}\mathbf{P}^T\mathbf{P}\mathbf{F}^{-1}\tilde{\mathbf{r}}^k \end{cases} \text{until } \tilde{\mathbf{r}}^k \sim 0 \\ &\hat{\mathbf{x}}(t) = \hat{\mathbf{x}}(t-1) + \tilde{\mathbf{S}}_x \tilde{\mathbf{A}}^H \tilde{\mathbf{w}}_P^K \end{aligned} \tag{33}$$

Where

$$\tilde{\mathbf{Q}} = \left(\tilde{\mathbf{A}}\tilde{\mathbf{S}}_x \tilde{\mathbf{A}}^H + \tilde{\mathbf{S}}_{n_y} \right)^{-1} \tag{34}$$

$\tilde{\mathbf{Q}}$ is a block diagonal matrix, with one block of size $M \times M$ for each spatial frequency. These blocks represent the spatial filtering that must be done to combine the *M* wavefront sensor phases so that they add constructively when they are back-propagated into the volume.

Accordingly, we call the quantity $\mathbf{w}_P^K$ the *preconditioned* wavefront error. When it is back propagated (and then post-conditioned by the covariance matrix $\mathbf{S}_x$) it will produce the minimum variance update to the volume estimate $\hat{\mathbf{x}}(t)$.

The inverse-tomography reconstructor of equation (34) is implemented in the RTC as shown in Figure 6. Most of the operations in the loop are multiply-accumulates of a single data number by a constant, one per spatial frequency element. This is done with a massive array of compute elements, each dedicated to a single spatial frequency and all operating in parallel. Two Fourier transform pairs are required in algorithm (33). The RTC will accomplish this using a fast algorithm tuned to the massive parallel processing architecture described in the RTC Design Document [(2)]. Interpolation of spatial frequencies is required during the forward and back propagation steps. This is not a small grid distortion (metapupils contract ~20%, so the sample points contract about a 6 subaps in the most extreme cases). The interpolation requires both shifting and the use of formula (16.

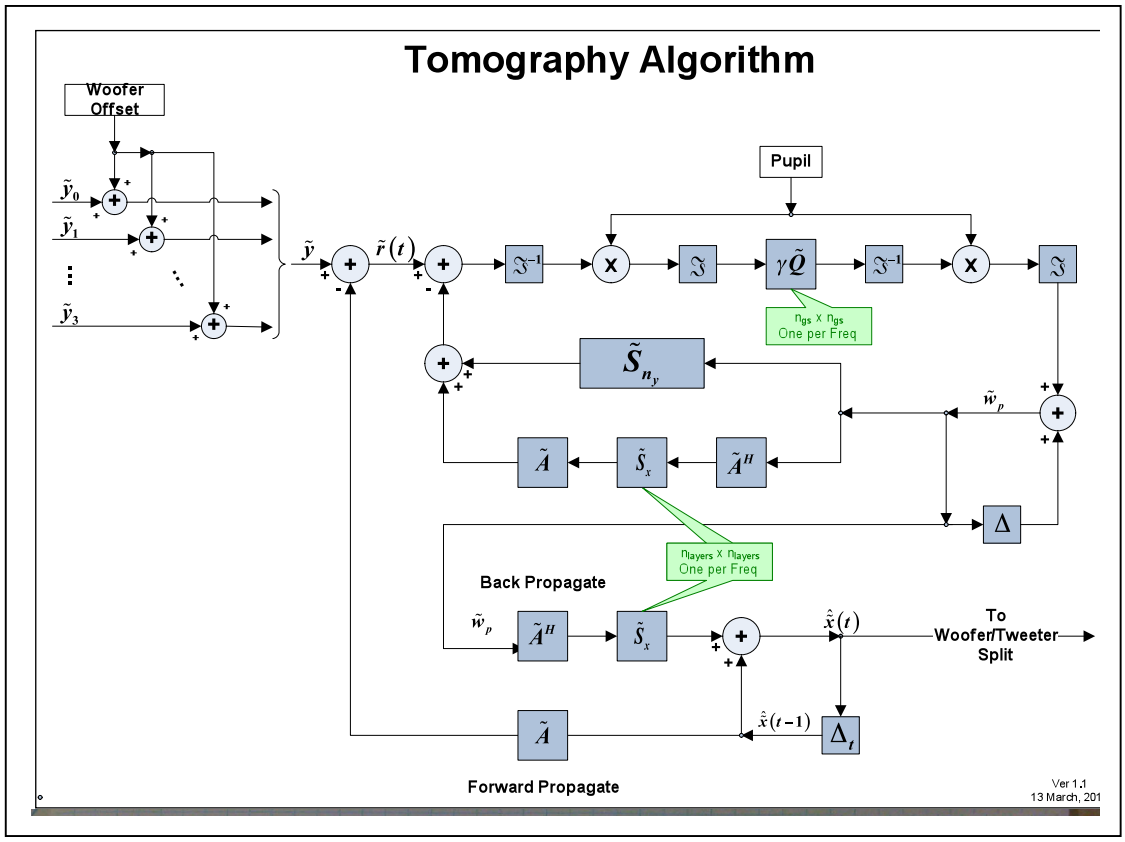

Figure 6. Block diagram of inverse tomography algorithm

Again, a "warm-restart" will be used in the RTC, where the preconditioned wavefront estimate from the prior time step is the starting point for the current time step. I.e.

$$\tilde{\mathbf{w}}_P^0(t) = \tilde{\mathbf{w}}_P^K(t-1) \tag{35}$$

where K is the last iteration at time step t-1. Simulation studies have shown that about 3 iterations per time step are sufficient under NGAO nominal conditions in order to maintain tomographic reconstruction error within error budget tolerance

2.5 DM Command Generation

DM command generation consists of the following steps:

1. Given the volume estimate of delta-index variations $\hat{\mathbf{x}}(t)$ produced by the tomography engine, forward propagate in the science direction (on-axis) to the ground layer.
2. Split the commands into woofer and tweeter components.
3. Do a least-squares fit of the woofer actuator influence functions to the desired woofer shape to produce woofer actuator commands
4. Apply the open-loop nonlinear model to the desired tweeter shape to produce tweeter actuator commands.

The forward propagation, since it is on-axis, involves simply adding all the layers in the volume:

$$\tilde{\phi}(\mathbf{f}) = \sum_{l=0}^{L} \tilde{x}(\mathbf{f}, z_l) = \tilde{\mathbf{A}}_s \tilde{x} \tag{36}$$

To determine which portion goes on the woofer, we make a least squares fit to the portion of the wavefront that has Frequency components less than the woofer's Nyquist frequency, f_w.

$$\tilde{\phi}_w(\mathbf{f}) = \begin{cases} \tilde{\phi}(\mathbf{f}); & f_x < f_w \text{ and } f_y < f_w \\ 0 \quad ; & f_x \geq f_w \text{ or } f_y \geq f_w \end{cases} \tag{37}$$

We assume that the woofer's response is a linear superposition of actuator influence functions, described by the spatial filter function $\tilde{W}(\mathbf{f})$.

$$\tilde{\varphi}_w(\mathbf{f}) = \tilde{W}(\mathbf{f})\tilde{a}_w(\mathbf{f}) \tag{38}$$

We ignore the nonlinearities because this DM is driven in closed loop. An approximation $\tilde{B}(\mathbf{f})$ to the inverse of $\tilde{W}(\mathbf{f})$ is used to provide the least-squares fit of actuator commands to the desired low-order shape.

$$\tilde{a}_w(\mathbf{f}) = \tilde{B}(\mathbf{f})\tilde{\phi}_w(\mathbf{f}); \quad \mathbf{f} < \mathbf{f}_w \tag{39}$$

Where we've used the shorthand notation $\mathbf{f} < \mathbf{f}_w$ to represent inside the Nyquist box $f_x < f_w$ and $f_y < f_w$. $\tilde{B}(\mathbf{f})$ is actually a parameter that can be tailored by the AO operator. Since DMs tend to act like low-pass filters, one might be tempted to "boost" the high frequency components in order to compensate. The penalty is that $\tilde{W}(\mathbf{f})$ can have (depending on the DM) significant spatial frequency components beyond Nyquist, and these will be aliased and amplified by $\tilde{B}(\mathbf{f})$. It might be a better trade-off to allow some spectral energy near the woofer's Nyquist be corrected instead by the tweeter in order to limit the amount of re-correction of aliased woofer commands. The exact choice of $\tilde{B}(\mathbf{f})$ will have to be done experimentally once $\tilde{W}(\mathbf{f})$ is known so as to optimize the use of stroke on both deformable mirrors.

Outside the woofer-Nyquist box, the spectrum of actuator commands $\tilde{a}_w(\mathbf{f})$ must be mirror-replicated to account for aliasing

$$\begin{aligned} \tilde{a}_w(f_x, f_y) &= \tilde{a}_w(f_w - f_x, f_y) \quad f_w < f_x < 2f_w \\ \tilde{a}_w(f_x, f_y) &= \tilde{a}_w(f_x - 2f_w, f_y) \quad 2f_w < f_x < 3f_w \\ &\vdots \end{aligned} \tag{40}$$

until the entire spatial frequency grid associated with the high-order wavefront sensor (HOWFS) is filled out.

Then, the shape of the woofer is

$$\tilde{\varphi}_w(\mathbf{f}) = \tilde{W}(\mathbf{f})\tilde{a}_w(f) \tag{41}$$

The desired shape on the tweeter is then

$$\tilde{\phi}_{tw}(\mathbf{f}) = \tilde{\phi}(\mathbf{f}) - \tilde{\varphi}_w(\mathbf{f}) \tag{42}$$

Figure 7 shows the woofer-tweeter split and woofer fit operations in block diagram form.

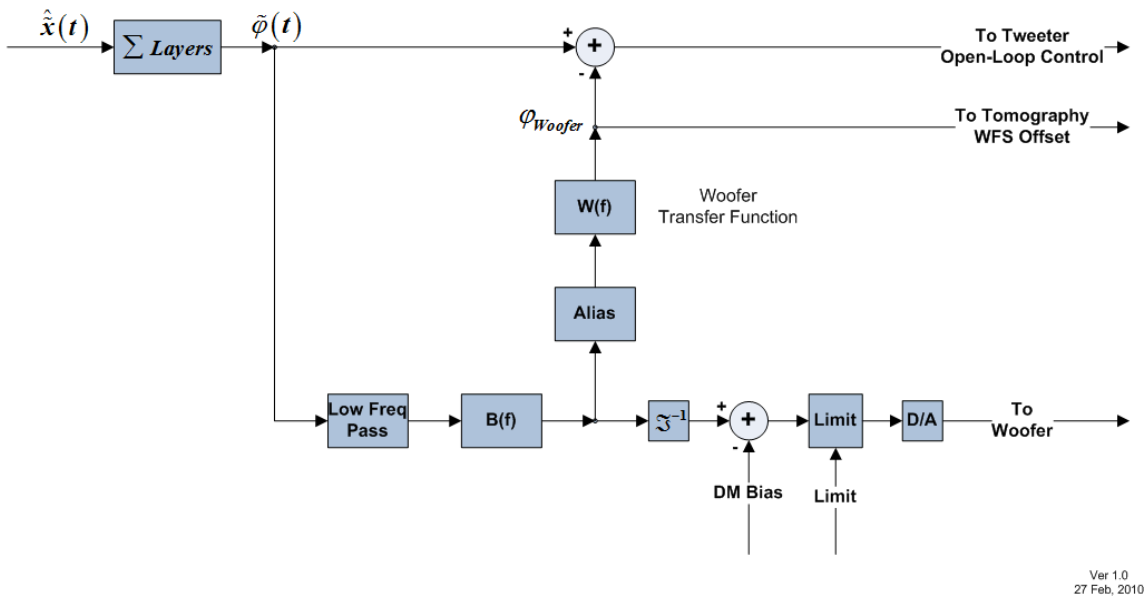

Figure 7. Block diagram of woofer / tweeter split

Tweeter command generation follows a more complicated set of operations because of the need to control these DMs open loop to high accuracy. The method outlined below is considerably more accurate than assuming a linear superposition fit [(4)].

The voltages that need to be applied to MEMS actuators depend on both the top plate displacement and the force that needs to be exerted on the top plate by the actuator. To close approximation, the relationship between the displacements and forces is linear (equation 12), albeit the forces are cross-coupled to neighboring actuators by the partial-differential plate equation.

The first step is to calculate these forces using the Fourier domain equivalent to the plate-equation's bi-Laplacian operator:

$$\tilde{f}_p(\mathbf{f}) = \tilde{\Phi}(\mathbf{f})\tilde{\phi}_{tw}(\mathbf{f}) \tag{43}$$

The second step is to transform both $\tilde{\phi}_{tw}(\mathbf{f})$ and $\tilde{f}_p(\mathbf{f})$ to the spatial domain

$$\begin{aligned} \phi_{tw} &= \mathbf{F}^{-1}\tilde{\phi}_{tw} \\ \mathbf{f}_p &= \mathbf{F}^{-1}\tilde{\mathbf{f}}_p \end{aligned} \tag{44}$$

then, for each actuator individually, use a nonlinear function (implemented as a lookup table or splines, see the Real Time Controller Design Document for details) to determine the actuator voltages:

$$V_j = v\left(f_p\left(\mathbf{u}_j\right), \phi_{tw}\left(\mathbf{u}_j\right)\right) \tag{45}$$

2.6 Pseudo Open Loop Control

The term "pseudo open-loop control" is meant to convey that a closed-loop controller is operated, conceptually, as an open loop controller, using a signal that consists of the residual measured in the sensor added to the known command to the actuator. The main advantage of this approach, if implemented properly, is that it results in a stable control loop by definition. A potential second "advantage" is that the minimum-variance formulation can use the open-loop a-priori and measurement covariance functions instead of the closed-loop versions.

Stability is assured so long as the signal that is added back in to the residual is in fact the value of the actuator, which is not necessarily the same as the value of the control. The "uncontrollable" modes must be well understood and accounted for, or the loop will simply drive them unstably. The equivalent concept in control theory is called the "separation principle" where state estimation is separate from control generation, and stability is assured so long as the *conditional mean* of the state is used to form the feedback control signal. Then stability is the same as if the actual state itself was used in the feedback [(5)].

The estimation algorithms presented above (sections 2.3 and 2.4) are designed to produce the conditional mean value of the state, $\mathbf{x}(t)$, the delta-refractive-index over the atmospheric volume above the telescope. The controls then applied to the deformable mirrors then proceed from this state estimate. In the case of the tweeters, these are open loop. In the case of the woofer, the control is closed loop. The important step in the woofer closed loop operation is to use $\tilde{\varphi}_w(\mathbf{f})$, the actual shape on the woofer, in the summing block of Figure 4, instead of $\tilde{\phi}_{tw}(\mathbf{f})$ from (36).

The second "advantage," that of using open-loop a-priori and measurement covariance functions is not really an advantage, but an approximation. The reason is this does not take into account that previous iterations of the controller over time have changed the covariance properties of the state estimate. Thus, at a particular point in time, adding the residual wavefront to the DM wavefront, which is determined from the state estimate, does not have the same statistical properties as the true state.

We advocate the use of closed-loop steady state covariances (expressed as power spectral densities $\tilde{\mathbf{S}}_x$, $\tilde{\mathbf{S}}_y$, and $\tilde{\mathbf{S}}_{n_y}$) computed off-line using the solution to the steady-state Riccati equations. The resulting controller will then, in steady state, be stable, and the state estimate will be convergent to the minimum variance estimate. Only during a brief start up period after closing the loop will the estimate not be the minimum variance estimate.

The Riccati equations are simplified by the use of the Fourier domain. Each spatial frequency as seen on the ground layer has its own independent Riccati equation with a state vector of dimension equal to the number of layers and a measurement vector of

dimension equal to the number of wavefront sensors. However, again, this calculation is offline and is used to create $\tilde{\mathbf{S}}_x$, $\tilde{\mathbf{S}}_y$, and $\tilde{\mathbf{S}}_{n_y}$ as parameters to load into the RTC.

2.7 Split Tomography

The term "split tomography" was coined by Brent Ellerbroek and Luc Gilles of the TMT/NFIRAOS project to explain a simplification of the means of including tip/tilt/focus/astigmatism information from sensors pointing at natural guidestars as an adjunct to the high-order wavefront sensor data derived from laser guide stars.

In the NGAO system architecture, the point-and-shoot guidestars are treated independently of the tomography. The tomography constellation of 4 laser guide stars is concentrated in a ~10 arcsecond diameter field of regard, where as the point-and-shoot guidestars are positionable over a 2 arcminute diameter field of regard. In the most probable of on-sky configurations, the point-and-shoot guidestars are outside of the tomography field of view. This precludes our use of the tip/tilt stars as direct sources of information for tomography via ray-tracing, since they are probing atmosphere that is outside the RTC's model volume (this restriction in model volume was one of the cost-saving measures in the build-to-cost replan).

Instead, we propose to use statistical correlation methods (with correlations determined either empirically or via simulation) to relate the tip/tilt/focus/astigmatism information to the required on-axis science tip/tilt correction and the 8 higher order blind modes introduced because of the lack of tip/tilt information from the 4 laser guidestars.

The two tip/tilt sensors and one tip/tilt/focus/astigmatism sensor produce 12 measurement values. These 12 values are to be combined linearly to produce the science path tip/tilt estimate, and the estimates of tip/tilts on the 4 tomography guidestar paths, a total of 10 values. Thus a 10 x 12 linear estimation matrix is used. The equation is

$$\mathbf{x}_{tt} = \left\langle \mathbf{x}_{tt}\mathbf{x}_{ttfa}^T \right\rangle \left\langle \mathbf{x}_{ttfa}\mathbf{x}_{ttfa}^T \right\rangle^{-1} \mathbf{x}_{ttfa} = \mathbf{E}_{tt}\mathbf{x}_{ttfa} \tag{46}$$

where $\mathbf{x}_{ttfa}$ is the vector of measurements and $\mathbf{x}_{tt}$ is the vector of unknown tilts. The tilt estimation matrix $\mathbf{E}_{tt}$ is determined off-line and loaded into the RTC as a parameter.

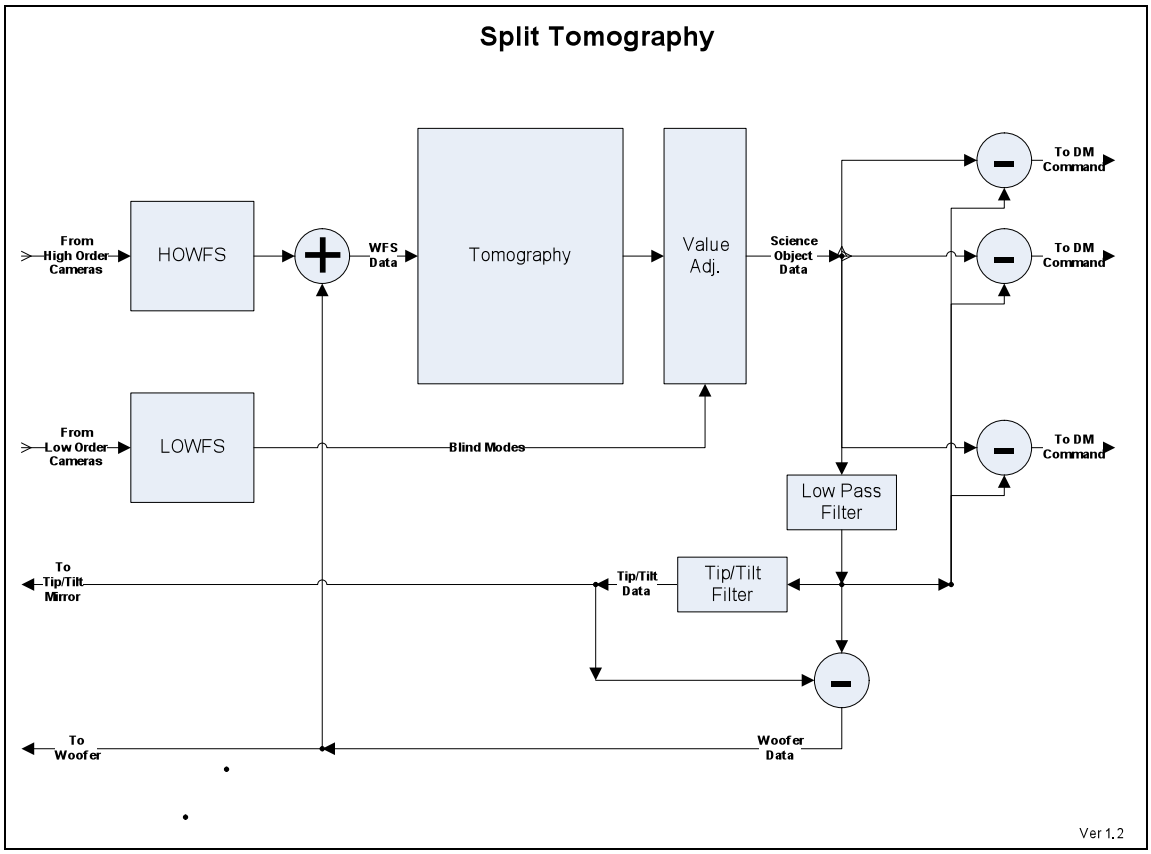

Figure 8. Split Tomography block diagram, showing also the woofer/tweeter split

3. Algorithm Interpretation

3.1 Spatial Domain Interpretation

The massive-parallel computing architecture is based on a systolic arrangement of small compute elements, all operating in parallel, and with data communication paths to their immediate neighbors. The compute elements perform computations locally, and all simultaneously. Data can be transmitted between neighbors in a given direction in one simultaneous shift operation.

With this arrangement of processors, it is possible to map the physical space on to the array directly. At various stages in the algorithm, processors represent:

- 3-D spatial sample points in the atmospheric volume
- 2-D spatial sample points on the aperture associated with each wavefront sensor
- 2-D spatial sample points on the aperture associated with each deformable mirror
- 2-D Fourier domain sample points in each layer of atmosphere
- 2-D Fourier domain sample points on the aperture associated with each wavefront sensor
- 2-D Fourier domain sample points on the aperture associated with each deformable mirror

The ability to do local processing and to communicate with local neighbors enables a large class of computations to be performed in fast parallel operations. The NGAO real-time operations have all been successfully mapped to this architecture to take advantage of parallelism.

Figure 1 shows the flow of data through the various stages of the real-time control processing. The tomography HOWFS and the point-and-shoot HOWFS each have dedicated processors implementing

- Image processing
- Centroiding
- Solution to the gradient Equation (4)

Each HOWFS processor has a 2-dimensional array of compute elements, mapped to the grid of subapertures. Each compute element connects to its neighbor in (x,y) space. At the start, each compute element contains the pixel raw values associated with the 4x4 grid of pixels in the subaperture region of interest. At the end, each compute element contains the reconstructed phase as that position in the grid. Since the solver for the gradient equation is implemented in the Fourier domain (Equations (21) and (22)), we defer its description to section 3.2. The image processing and centroiding however are straightforward to implement in the spatial domain.

The parameters required in each compute element of the wavefront reconstructor include those listed in Table 1.

Table 1 Fixed parameter storage in a wavefront sensor compute element

Parameter	Number of values	Description
Aperture mask	1	Whether or not the subaperture is illuminated
Dark subtraction	4	On 2x2 pixel grid
Flat field	4	On 2x2 pixel grid
Centroider weights	8	x and y centroid weights
Nonlinearity coefficients	~10	If needed
Interpolator weights	4 complex	For distortion compensation
Fourier kernel coefficients	$4N_1$ complex	x, y, forward and inverse transform
Gradient operator	2 complex	**G**, equation (20)
Divergence operator	2 complex	$\mathbf{G}^H$, equation (20)
A-priori covariance	1 complex	$\mathbf{S}_y$
Noise covariance	1 real	⍰⍰
Preconditioner matrix	1 complex	**C**, equation (23)

Each compute element is required to store the following data:

Table 2 Variable data storage in a wavefront sensor compute element

Datum	Number of values	Description
Pixel data numbers	16	4x4 Hartmann region
Slopes	2	x and y slopes
Slope Fourier coefficients	2 complex	x and y Fourier coefficients
Phase Fourier coefficients	1 complex	Reconstructed phase values

At the end of wavefront phase reconstruction, phase data from the wavefront sensors are aggregated as input to the tomography engine.

In the tomography engine, the compute elements are assigned to discrete samples of the atmospheric volume, divided into layers and then within each layer into lateral samples. The resulting volume element samples are called *voxels*. Each element of this three-dimensional array has connection with its immediate neighbor in x,y, and z directions. The array is thus a microcosm of the atmosphere as sampled on to the voxels, each voxel having its associated compute element. In the end, each voxel will contain an estimate of the delta-index of refraction value for its associated x,y,z location within the volume.

At the start, the layers also are affiliated with wavefront sensors, one high order wavefront sensor per layer. (The number of layers is equal one more than the number of

high-order wavefront sensors. This is to maintain (2) as an underdetermined equation, while avoiding introducing unnecessary additional blind modes, see (6)). Each layer of compute elements starts with the corresponding wavefront sensor's phase data, one (Fourier) sample per compute element. The iterative tomographic reconstruction algorithm (33 then proceeds, mostly in the Fourier domain, so we defer the description of the Fourier processing to section 3.2.

The algorithm (33 requires masking by the illuminated aperture twice per iteration. The operation $\mathbf{FP}^T\mathbf{PF}^{-1}$ means, inverse-transform from the Fourier to the spatial domain, multiply in the spatial domain by the aperture mask, then forward-transform back into the Fourier domain. The masking operation is done elementwise, in parallel on each compute element. The forward and inverse Fourier transforms are computed using the algorithm described in Appendix H of the RTC Design Document [(2)]. The transform operation involves shifting the data around the grid of compute elements of the layer (all layers operating simultaneously), multiplying and accumulating the data as it goes by until that compute-element's associated transform value is constructed. Every voxel at every layer is acccumlating its final value at every step, but it takes $2N_1$ shift steps to complete the operation, where N_1 is the number of subapertures across one dimension (x then y direction) of the layer.

In the spatial domain, the operation **A** represents forward propagation of the light from the guidestars through the layers, each ray accumulating delta-index from each layer, and depositing the result at the appropriately assigned layer. $\mathbf{A}^T$ represents the back-propagation of update values from each wavefront sensor, deposited and accumulated at each layer along rays back to the guidestars. It is necessary to laterally shift and interpolate data as rays propagate forward or backward so as to follow the rays in the laser guide star cone beams, with apex at the laser guidestar position. Here is a summary:

Forward Propagation (Equations (1), (26)):

- *Lateral Shift*: For the *k*'th guidestar, shift layer *l*'s data transversely by $\vec{\theta}_k z_l$ where z_l is the altitude of the *l*'th layer
- *Lateral Interpolate:* At each layer *l*, interpolate the data to resample it on a grid with spacing scaled by $1-$, where z_{gs} is the altitude of the guidestar.
- *Z-step*: Sum the layers, storing the result at the layer associated with WFS *k*.

Back propagation (Equation (28)):

- *Z-step:* Distribute WFS *k*'s data to all the layers
- *Lateral Interpolate*: At each layer *l*, interpolate the data to resample it on a grid with spacing scaled by -1.
- *Lateral Shift*: Shift layer *l*'s data transversely by $-\vec{\theta}_k z_l$. Accumulate this result as k=0,1,…M

These operations are coded on the systolic array such that all spatial positions are processed simultaneously. Lateral interpolation requires shifting in the neighboring 4 elements for bilinear interpolation, then shifting the result over to the correct grid location, taking up to $2N_1$ steps. Lateral shift is done elementwise (in the Fourier domain). Z-shift requires L sequential steps as data are transmitted layer to layer until all compute

elements have seen all the data associated with its (x,y) location. The K guidestars can be processed together, but then K data values must be transmitted on each Z-step. Equivalently, K propagations of L sequential steps can be done.

The parameters required in each compute element of the tomography engine include those listed in Table 3.

Table 3. Fixed parameter storage in a tomography engine compute element

Parameter	Number of values	Description
Aperture mask	1 real	Associated with WFS
Measurement covariance	1 complex	□□□ Associated with WFS
Precompensator weights	M complex	**Q**, associated with WFS
Interpolator weights and position locators	4 complex, 2 integer	For cone beam scaling at layer
Lateral shift factors	M complex	For positioning guidestar rays in layer, □□□□2□□□
A-priori covariance	1 complex	□□ Associated with layer
Fourier kernel coefficients	$4N_1$ complex	x, y, forward and inverse transform
Iteration gain	1 real	□ Iteration step gain
Feedback gain	1 real	Time step gain
Integrator gain	1 real	Time step integrator bleed

Each tomography engine compute element is required to store the following data:

Table 4 Variable data storage in a wavefront sensor compute element

Datum	Number of values	Description
WFS phase	1 complex	**y**
Woofer phase	1 complex	□
Measurement residual	1 complex	e
Temporary volume	1 complex	For intermediate calculations of (32)
Preconditioned measurement	1 complex	w
Volume delta-index	1 complex	x

Deformable mirror command generation was described in section 2.5. The initial determination of on-axis phase correction is accomplished within the tomography engine with the steps:

- Z-step to sum all the layers.
- Distort the wavefront to map on to regular woofer actuator grid
- Fit the woofer phase (equation (38)) with actuator influence functions
- Determine the resulting shape of the woofer (equation (41))
- Distort to the reference grid and subtract from the phase correction to form the tweeter displacement
- Distort to the tweeter actuator grid
- Calculate the plate forces on the tweeter, equation (43)
- Transform the woofer commands to the spatial domain
- Transform the tweeter displacements and tweeter forces to the spatial domain

Table 5 Fixed parameter storage in a tomography engine compute element on a layer associated with a deformable mirror.

Parameter	Number of values	Description
Interpolator weights	4 complex	For distortion compensation (Woofer)
Interpolator weights	4 complex	For distortion compensation (Tweeter)
Woofer compensator function	1 complex	**B**
Woofer influence function	1 complex	**W**
Bilaplacian Operator	1 complex	**Φ**

Table 6. Variable data storage in a tomography engine compute element on a layer associated with a deformable mirror.

Parameter	Number of values	Description
Displacement	1 complex 1 real	DM shape (Woofer and Tweeter)
Force	1 complex 1 real	(Tweeter)
Actuator commands	1 real	(Woofer)

We now ship the woofer actuator commands directly to the woofer driver.

The tweeter displacements and forces are transmitted to a DM processor (a GPU) in order to do fast nonlinear lookup table operations described in equation (45).

3.2 Fourier Domain Interpretation

We maintain the estimated atmosphere and the measured values from the WFSs in the Fourier domain.

The wavefront reconstruction algorithm (21), (22) requires multiplication by operators $\widetilde{\mathbf{G}}$, and $\widetilde{\mathbf{G}}^H$. These are given as 2-element vectors for each spatial frequency (20). In the case of multiplying by $\widetilde{\mathbf{G}}^H$ as in (21), the x and y components of a slope like object are multiplied respectively by the x and y components of $\widetilde{\mathbf{G}}^H$ and summed to form a single valued phase-like object. In the case of multiplying by $\widetilde{\mathbf{G}}$, the single valued phase-like object is multiplied by each element of $\widetilde{\mathbf{G}}$ to form a 2-vector slope-like object. The $\mathbf{FP}^T\mathbf{PF}^{-1}$ operation (multiplication by the pupil mask in the spatial domain) takes place on both the x and y components of the slope-like object. So, in reality, this masking operation takes place twice per iteration.

The preconditioner in the wavefront reconstruction algorithm is a complex scalar.

The tomography processing algorithm requires a preconditioner, $\widetilde{\mathbf{Q}}$, which maps the *M*-vector of wavefront sensor phase residuals to an *M*-vector of preconditioned wavefront sensor phase residuals. *M* is the number of high-order wavefront sensors, so the $\widetilde{\mathbf{Q}}$ is an *M* x *M* matrix multiply for each spatial frequency. Each compute element in the systolic array accomplishes this matrix multiply with *M* Z-shifts, accumulating the dot product of the wavefront phase data as it goes by with that of a pre-stored *M* vector in each processing element ("precompensator weights" in Table 3).

The forward and back-propagation operations $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{A}}^H$ take place in the spatial frequency domain as described earlier:

Forward Propagation (Equations (1), (26)):

- *Lateral Shift*: For the *k*'th guidestar, shift layer *l*'s data transversely by multiplying by the complex factor $\exp\left\{ i2\pi\mathbf{f}\cdot\theta_k z_l \right\}$ where z_l is the altitude of the *l*'th layer
- *Lateral Interpolate:* At each layer *l*, interpolate the data to resample it on a spatial frequency grid with frequency spacing scaled by $\alpha(z_l) = 1 - z_l / z_{gs}$, where z_{gs} is the altitude of the guidestar.
- *Z-step*: Sum the layers, storing the result at the layer associated with WFS *k*.

Back propagation (Equation (28)):

- *Z-step:* Distribute WFS *k*'s data to all the layers
- *Lateral Interpolate*: At each layer *l*, interpolate the data to resample it on a spatial frequency grid with spacing frequency scaled by -1.
- *Lateral Shift*: Shift layer *l*'s data transversely by multiplying by the complex factor $\exp\left\{ -i2\pi\mathbf{f}\cdot\theta_k z_l \right\}$. Accumulate this result as *k*=0,1,…,*M*

4. Requirements Flow Down

In this section we describe how the algorithm design meets NGAO requirements.

4.1 Speed of Calculations

The RTC must complete its calculations at sufficient speed to keep up with the rapidly changing atmospheric turbulence, providing a timely correction to the wavefront that will keep the science images diffraction-limited.

The bandwidth and latency requirements are given in FR-1406, which is traceable to an analysis in [7] for the exoplanets science case. Here is a summary of all the earlier published error budgets [8]

Science Case	High Order Bandwidth Allocation	Tip/Tilt Bandwidth Allocation	r_0 , cm	wind speed, v, m/sec	τ_0, ms	Notes	Pub date
Io	15 nm	0.57 mas	16 cm	8 m/s	5.93 ms	NGS mode	6/7/07
Kuiper Belt	17 nm	1.09 mas	16 cm	8 m/s	5.93 ms	LGS mode	8/7/07
ExoJupiter	21 nm	0.58 mas	16 cm	8 m/s	5.93 ms	LGS mode	7/6/07
Extended Groth	21 nm	1.45 mas	16 cm	8 m/s	5.93 ms	LGS mode	7/6/07
Galactic Center	30 nm	0.49 mas	16 cm	8 m/s	5.93 ms	LGS mode	6/7/07
KAON644	59 nm	1.3 mas	14.7 cm	9.5 m/s	4.4 ms	LGS mode	3/10/09

Since KAON644 represents post build-to-cost (March 09) analysis, it is being used as the definitive source, although the impact on science cases other than exoplanet is unclear.

Here is a summary of the post build-to-cost impacts on the RTC requirements [9]:

- Seven instead of nine LGS WFS and UTTs.
- Tomography only needs to be performed with the central “3+1” LGS asterism over a ~ 40” diameter science field for the imager and a ~ 5” diameter science field for the IFU.
- Three independent point-and-shoot AO control loops must be supported.
- Only one LGS WFS subaperture scale needs to be supported. This should be assumed to be 64 subapertures across the pupil.
- Only two NGS WFS subaperture scales need to be supported.
- No multi-IFU science needs to be supported.
- A fast tip-tilt mechanism for the tweeter DM is no longer required.

The Mauna Kea ridge model [10] provides the baseline design wind speed (average seeing) of 9.5 m/s.

4.1.1 High-order controller

The Greenwood frequency, often referred to in the error budget spreadsheets, is a concept tied to a single-pole feedback control (11). Since the NGAO system is a hybrid architecture, woofer in closed loop, tweeter in open loop, the most stressing bandwidth and latency requirements are on the open loop control. In open loop control, time delay determines the bandwidth performance according to (1)

$$\sigma_{BW}^2 = \left(\tau/\tau_0\right)^{5/3}$$

Where

$$\tau_0 = \left(6.88\right)^{-3/5}\left(r_0/v\right)$$

and v is the wind speed. The relationship to Greenwood's frequency, f_g is

$$\tau_0 = 0.134/f_g$$

The following table summarizes the assumed conditions:

r_0	v	τ_0	f_g
14.7 cm	9.5 m/s	4.4 ms	30.43 Hz

Latency time suballocations to the RTC high-order controller are as follows:

Table 7 Latency allocations for high order wavefront control

Suballocation category	Delay, microseconds	Flowdown requirement
Stare (1/2 frame)	250	
Readout	500	
WFS compute	300	
Tomography	450	
DM Cmd Gen	200	
DM Hold (1/2 frame)	250	
Latency contingency	100	
Stare (1/2 frame)	250	
Total lag	2050 μs	3075 μs
HO Bandwidth error	42.1 nm	59 nm

With 2.05 ms delay, the bandwidth error is 42.1 nm, compared to the error budget maximum of 59 nm.

4.1.2 Tip/Tilt Controller – Science Path

For tip/tilt, the allocation for bandwidth error is 1.3 milli-arcseconds. Tip/tilt control in the NGAO architecture is closed loop, since the LOWFS detect the light after it has been corrected by the tip/tilt stage. The closed loop tip/tilt error is [12] (formula 4.2-14)

=(/)

where f_c is the single-pole controller law crossover frequency (frequency where the closed loop transfer function, $f / (f+f_c)$, reaches -3db), and f_T is Tyler's tip/tilt frequency, given by [12] (formula 4.2-16):

$$\square\square=0.0811\square 0\square 16\square\square 0$$

The following table summarizes the assumed conditions:

r_0	v	f_s	f_c	f_T
14.7 cm	9.5 m/s	1 khz	50 hz	2.59 Hz

- f_s is the control loop sample rate
- The controller is single-pole integral control. The loop consists of an integrator, the time-latency excluding stare time and actuator hold time, and 2 zero-order holds to model the one-sample camera stare time and the one-sample hold time of the tip/tilt drive.

Latency time suballocations to the RTC tip/tilt controller are as follows:

Table 8. Latency allocations for tip/tilt control

Suballocation category	Delay, microseconds	Flowdown requirement
Stare (1/2 cycle)	500	
Readout	500	
Compute TT	100	
TT set voltage	30	
Hold (1/2 cycle)	500	
Latency contingency	100	
Total lag	1730 μs	3000 μs
HO Bandwidth error	0.54 mas	1.3 mas

The closed-loop controller achieves 50 hz bandwidth with a phase margin of 60 degrees at a sample rate of 1 kHz. The 3 ms total lag allowance is rather arbitrary, but is given as an example that achieves 1.3 mas tip/tilt error variance with a 20 hz bandwidth and a phase margin of 61 degrees at a sample rate of 1 kHz. Using Rich Dekany's suggested simplified formula of $\tau_{lag} = 0.134/f_T$ we get a lag tolerance of approximately 5 ms.

Because of the time lags, the controller will have some overshoot at frequencies beyond the -3 db amplitude point. The described controller will have about 6 db amplitude amplification at 100 hz. A lead-lag compensator will suppress this somewhat, therefore lead-lag compensation capability is a requirement on the two channels of tip/tilt control for the science fast steering stage.

4.1.3 Tip/Tilt Controllers – HOWFS Paths

This sections will be added the week of 12/1/09.

4.1.4 Tip/Tilt Controllers – LOWFS Paths

This sections will be added the week of 12/1/09.

4.2 Accuracy of Calculations

The calculation accuracy requirement is given in FR-1434 for high-order wavefront control (5 nm rms) and FR-1435 for tip/tilt control (1 mas rms). The numerical precision requirement is given in FR-1408 (1 nm rms).

Both the wavefront reconstruction and tomography steps require iteration. Upon convergence, the wavefronts at the output of the wavefront reconstruction and the output of the tomography engine, have achieved the match to input data to the accuracy of the machine word length. The word length is selected so that digitization error is less than other physical limits, such as wavefront measurement error due to photon noise. The current error budget has this set at a few nanometers, and given that starting wavefronts are several microns rms, the 18 bit word length baseline for use in the RTC was deemed sufficient.

The Hartmann centroiding error depends on the photon-counting and read-noise properties of the wavefront sensor. The digitization error, or photon noise, whichever is dominant, carries over to the error in the reconstructed wavefront. The 12-bit camera pixel data is sufficient to process the 1000 detected photo-electrons per Hartmann spot per frame to a level well below its inherent 33:1 signal to noise ratio set by photon noise.

The tomography algorithm is designed to converge to a minimum variance solution. Again, the digitization word length is chosen to represent numbers well below the expected residual tomography wavefront error. The residual error is expected to be in the 10 nm regime, and since the whole wavefront is on the order of 10's of microns rms, a signal to noise of at least 1000:1 (10 bits) needs to be maintained in all calculations, in particular the subtraction in the iteration loop. 18 bit calculations are the baseline.

DM command generation requires the use of a non-linear functional lookup. Because nonlinearities change the bit accuracy between input to the table and output, the D/A converter word length is designed to maintain the resulting wavefront control accuracy to below the assigned error budget number. At the fasted part of the MEMS nonlinear response curve, the wavefront is changing at about 100 nm/V, so ~100mv out of a total of 240 V accuracy is needed on the D/A, or about 12 bits.

The nonlinear model for open loop DM control is accurate to about 30 nm wavefront (15 nm surface), so a signal to noise on the order of 1000:1 should be maintained during the process of non-linear function fit. The baseline design is to use the GPU to implement 100-coefficient spline functions to map mirror displacements and forces to MEMS drive voltages. Accuracy of this technique well exceeds the 30 nm inherent accuracy of the model. The model is established with a calibration process performed on the MEMS DM. The calibration visits 400 force and displacement points in a 20x20 array, [(4)] recording the voltages at each of these points. The resulting voltage surface is quite smooth and is accurately modeled by the 100-coefficient spline method. The GPU must now complete this operation for all the actuators on the DM within the margin of the latency time allocated to it.

4.3 Interface Data

4.3.1 Cameras

The wavefront sensor input data rate requirement is stated in FR-1437. The RTC will need to be able to accept the 256 x 256 array of camera pixels at 12 bits per pixel from each of the 7 high-order wavefront sensors within a frame cycle time of 500 ms and meet the latency requirements in Table 8. All wavefront sensor input data streams are handled in parallel by separate processors, so the latency and frame rates are not affected by the number of HOWFS.

4.3.2 DMs

The DM data rate requirement is stated in FR-1456. DM output rates to the 20 x 20 woofer, 64 x 64 tweeter, and the 7 32 x 32 LOWFS DMs must be sufficient to meet the latency requirements in Table 8.These are met with a 1.6 MB/sec interface to the woofer, a 16.4 MB/sec interface to the tweeter, and 8.2 MB/sec interfaces to the LOWFS DMs.

4.3.3 Diagnostic and Telemetry Data

The listing of required diagnostic and telemetry data streams is given in FR-1416. The purposes of these data streams are to:

- Provide immediate information to the AO operator, via the supervisory control
- Provide a means of recording all dynamic data at full frame rate for purposes of
  a) system commissioning and error diagnostics
  b) later evaluation of system conditions as they relate to science data postprocessing.

4.3.4 Offloading

Offloading is mentioned in requirements FR-1405 and FR-1414. The RTC will provide offload information to the supervisory control for subsequent dispatch to other subsystems within NGAO or to the telescope operating system. The RTC provides the offload information at a rate much slower (set by the supervisory system) than the full frame rate of the real-time control. Since this is subsampled data, steps need to be taken in the RTC to prevent aliasing.

Subsampling and anti-aliasing is achieved as follows. Within the RTC, all full-frame-rate telemetry streams pass out of the systolic array on dedicated paths to the high-speed RAID disk. As this data is transmitted, compute elements are assigned to also low-pass filter it, using a one-pole digital filter that is implemented in a few lines of code:

$$\bar{x} = \beta \bar{x} + (1 - \beta) x$$

This calculation is a scalar operation that takes place at every compute element simultaneously whenever a given dynamic variable (one of those listed in FR-1416) is completed updating once per frame. The compute element stores the state of the low-pass filter and provides it to the supervisory control either on request or at a low periodic rate. The parameter β is a positive constant less than one, chosen appropriate to the offload update rate. E.g. if the offload cycle is every $N_o >> 1$ frame cycles, then

$$\beta = e^{-1/2N_0}$$

will put the pole of the low-pass filter at the offloading Nyquist rate.

It is the responsibility of the supervisory control to do any further processing of this slow-rate data and to distribute it to the appropriate subsystem.

4.4 Calibration

Calibration data requirements are given in FR-1417. The above telemetry data, or alternatively, the low-pass filtered offload data can be used as a data stream to aid in calibration as outlined in FR-1417.

4.5 Parametric Data

The need for parametric data input to the RTC is mentioned in FR-1414. All parameters listed in Table 1, Table 3, and Table 5 are required by the RTC algorithms to operate. The supervisory control system provides this data either directly through its interface or file locators so that the RTC can load moderate amounts of parametric data directly from disk.

The RTC parameters (in Table 1, Table 3, and Table 5) are in a form that is suited to the RTC basic algorithm operations. Some of these items depend indirectly on the more familiar seeing parameters such as r_0, wind speed, C_n^2, etc. It is the responsibility of the supervisory control system to orchestrate the translation to RTC form given the "familiar" descriptions of operating configuration and seeing parameters.

The RTC will immediately incorporate new parameters when they are loaded, even when the RTC is cycling in a real time control state, synchronized to a frame cycle so as to minimize mid-calculation disruptions.

Additional information will eventually be posted here (~late December, 2009) providing instructions for how to compute the RTC parameter set (Table 1, Table 3, Table 5) given basic or fundamental parameters like r_0, wind speed, C_n^2, etc. Some of the off-line calculations are moderately extensive, such as determining the steady-state covariances, and will require a substantially powerful off-line computer to keep up with changing seeing conditions. Some further analysis is needed to calculate just how powerful a computer is needed, trading off against the consequence of erroneous seeing assumptions in the RTC algorithms. In the NGAO environment, we expect the laser guidestar signals to be bright enough that the system will be generally operating in a high signal-to-noise condition. In this case, the sensitivity of AO performance to a-priori covariance information will be low.

5. References

1. Hardy, J.W. *Adaptive Optics for Astronomical Telescopes.* New York : Oxford University Press, 1998.
2. Reinig, M., and Gavel, D. *NGAO Real Time Controller Design Document.* 2009. KAON xxx.
3. Gavel, D. *Stability of Closed Loop Tomography.* UC Santa Cruz : Laboratory for Adaptive Optics Report, 2004.
4. Morzinski, K. and Gavel, D., *The open-loop control of MEMS: modeling and experimental results,* Proceedings of the SPIE 6467, 2007.
5. Brezinski, C. *Computational Aspects of Linear Control (Numerical Methods and Algorithms).* New York : Springer, 2002.
6. Gavel, D. *Noise Propagator for Laser Tomography AO.* Keck Next Generation Adaptive Optics Project : KAON 621, 2009.
7. Dekany, R. *Build-to-Cost Architecture Performance Analysis.* Keck Next Generation Adaptive Optics Project : KAON 644, 2009.
8. Wavefront Error Budget. *Keck NGAO Twiki.* [Online] http://www.oir.caltech.edu/twiki_oir/bin/view/Keck/NGAO/WavefrontErrorBudget.
9. Wizinowich, P. *Design Changes in Support of Build-to-Cost.* Keck Next Generation Adaptive Optics Project : KAON 642, 2009.
10. Dekany, R. *Mauna Kea Ridge (MKR) Turbulence Models.* Keck Next Generation Adaptive Optics Project : KAON 503, 2007.
11. Greenwood, D. Bandwidth specification for adaptive optics system. *Journal of the Optical Society of America.* 1977, Vol. 67.
12. Wizinowich, P. *Design of the Keck Adaptive Optics System.* Keck Observatory : KAON 208, 2000.

Appendix A: Table of symbols used in this document

The following is a brief description of each mathematical symbol used in the algorithm descriptions

Table 9 Table of Symbols used in this Document.

Symbol	Meaning	1st use equation
a_w	Actuator commands to the woofer DM	(38)
$\mathbf{A}$	Forward propagation operator in tomography	(2)
$\mathbf{A}^T$	Back-propagation operator in tomography	(28)
$\mathbf{B}$	Woofer DM control spatial filter (designed to compensate for the woofer actuator response function)	(39)
$\mathbf{c}$	Divergence of Hartmann slopes	(21)
$\mathbf{C}$	Preconditioner for phase estimation from slopes (approximate inverse of Laplacian)	(23)
c	Hartmann sensor centroid readings	(11)
$c_{jj'}$	Interpolation weights (for correcting spatial grid distortion)	(16
$d_j(V_j)$	Displacement of the woofer actuator in response to a voltage	(13)
$\mathbf{e}$	Wavefront phase prediction residual (tomography)	(32)
$\mathbf{E}$	Low-order blind mode reconstructor (including science path tip/tilt) from LOWFS data	(46)
$\mathbf{f}$	Spatial frequency	(6)
f	Index in the spatial frequency sample grid	(6)
$\mathbf{F}$	Fourier transform operator	(7)
$\mathbf{F}^{-1}$	Inverse Fourier transform operator	(7)
f_e	MEMS actuator electrostatic force	(12)
f_p	MEMS Plate force	(12)
f_s	MEMS actuator spring force	(12)
f_x, f_y	Spatial frequency (x and y components)	(20)
f_w	Nyquist spatial frequency of the woofer DM	(37)

Symbol	Meaning	1st use equation
$\mathbf{G}$	Gradient operator	(4)
$\mathbf{G}^T$, $\widetilde{\mathbf{G}}^H$	Divergence operator (over-tilde denotes frequency domain version; H denotes Hermetian transpose)	(18)
I	Detected photo-counts in a pixel of the WFS	(15)
j	Index in the spatial domain sample grid	(3)
$\mathbf{K}$	Conditional mean operator (tomography)	(31)
L	Number of layers in the model atmospheric volume	Section 2.4
M	Number of high-order wavefront sensors used for tomography	Section 2.4
N	Number of sample grid points on a single wavefront sensor or tomography volume layer	(7)
N_1	Number of sample grid points in one dimension across a wavefront sensor or tomography volume layer. $N = N_1 \times N_1$	Section 3.1
$\mathbf{P}$	Selects data within the aperture	(5)
$\mathbf{PP}^T$	Sets all data outside the aperture to zero	
$\mathbf{Q}$	Preconditioner for tomography	(32)
$\mathbf{r}$	Equation error during iterations to find the preconditioned wavefronts	(32)
$r_j(\mathbf{u}-\mathbf{u}_j)$	Woofer spatial response function	(13)
$R_k(\mathbf{u})$	Spatial grid distortion map	(14)
s	Hartmann slope measurement	(3)
$\mathbf{s}$	All Hartmann slopes, lined up in a vector in raster scan order and grouped by wavefront sensor	(4)
$\mathbf{S}_y$	A-priori covariance of wavefront phase	(18)
$\mathbf{u}$	Spatial position in the spatial domain	(1)
V	MEMS actuator applied voltage	(12)
$v(.,.)$	MEMS nonlinear voltage lookup table; force and displacement as inputs	(45)

Symbol	Meaning	1st use equation
$\mathbf{w}$	Preconditioned wavefront error for back-propagation (tomography)	(32)
w	Centroid weights	(15)
$\mathbf{W}$	Woofer frequency response function	(38)
$\mathbf{x}$	All delta-index values for every sample point in the volume, lined up in a vector	(2)
$\mathbf{y}$	All wavefront phase measurements, lined up in a vector in raster scan order and grouped by wavefront sensor	(2)
$\mathbf{z}$	Spatial position in altitude	(1)
[illegible]	Cone beam transverse scaling factor	(1)
σ_n^2	Error variance of the Hartmann sensor slopes	(18)
ξ	MEMS actuator displacement	(12)
[illegible]	Index of refraction	(1)
ϕ	Desired wavefront correction	(36)
ϕ_w	Desired shape of woofer DM	(37)
$\mathbf{\Phi}$	BiLaplacian operator	(43)
φ_W	Actual shape of the woofer DM	(32)
[illegible]	Off-axis angle of guide star k	(1)
[illegible]	Wavelength of guidestar light	(1)