# Sparse Matrix Methods for Large-Scale

# Closed-Loop Adaptive Optics

Luc Gilles    lgilles@mtu.edu

Michigan Technological University, ECE Department

January 23, 2004

# Large-Scale Adaptive Optics

- Extreme Adaptive Optics (ExAO), e.g. coronographs, XAOPI (4K actuators),...

- Multi-Conjugate Adaptive Optics (MCAO), e.g. 32m TMT (CfAO, NOAO), 50m Euro50 (consortium), 100m OWL (ESO),...

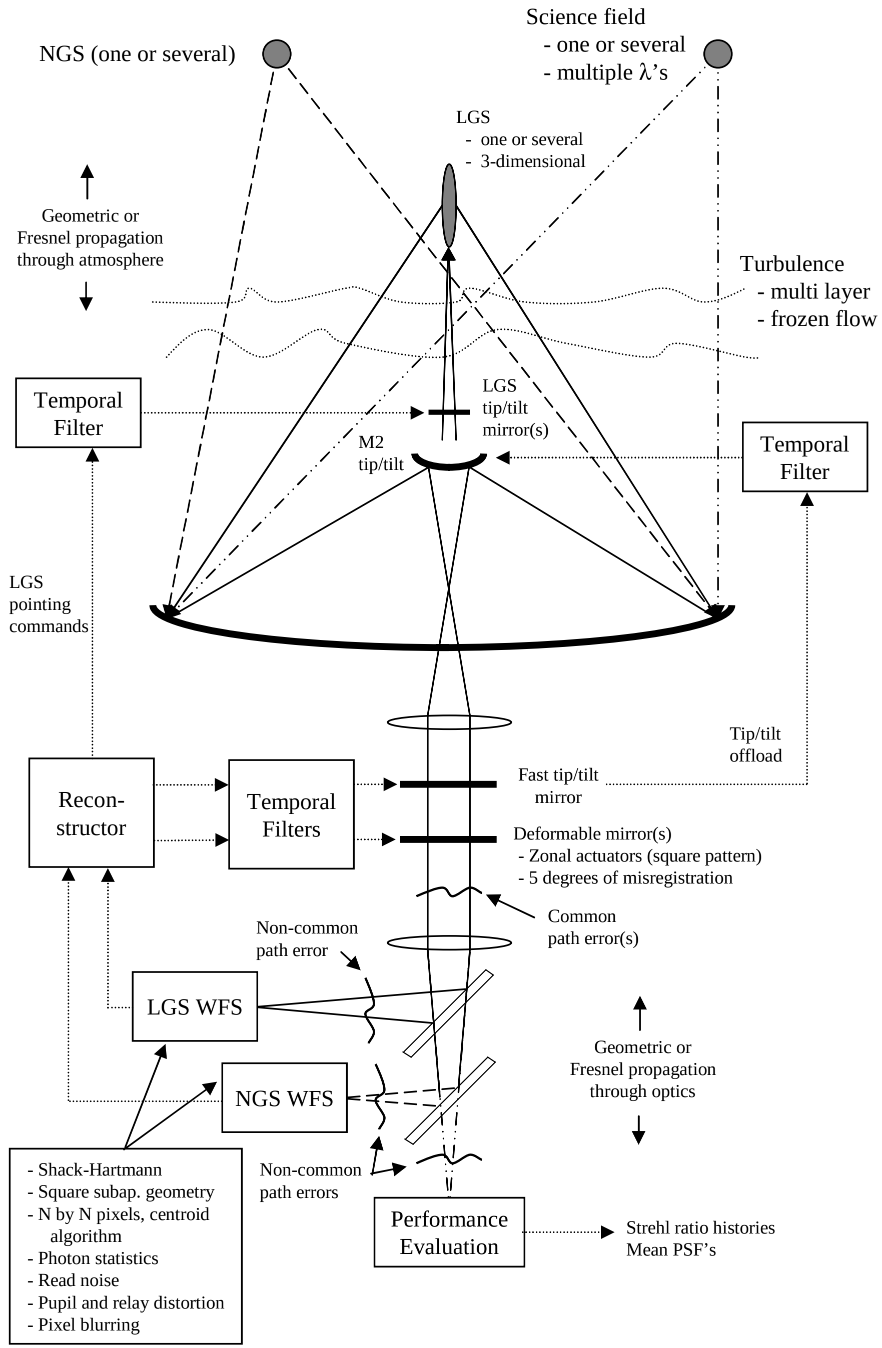
# Basic Model Assumptions

- Geometrical optics propagation through atmos-AO
- Taylor Frozen flow hypothesis
- WFS measurements $\vec{s}(n)$ are linear in the input turbulent OPD profiles $\vec{x}(n)$ and DM OPD profiles $\hat{\vec{a}}(n)$
- DM residual profile $\hat{\vec{e}}_a(n)$ is linear in measurements $\vec{s}(n)$
- Standard type-1 discrete integrator with 2-cycle lag

Note:

- Work on wave optics propagation in progress...
- Formalism valid for both ExAO and MCAO
- MCAO: block structured matrices, concatanated vecs

# Generic System Layout



# Discrete-Space-Time Servo-Loop

- Standard Control Algorithm:

$$n = 0;\ \hat{\vec{a}}(0) = \vec{0};$$

begin cycles

$$[\text{PSF}(n), \text{OTF}(n), \sigma^2(n)] = \text{perfevl}[\,\vec{x}(n), \hat{\vec{a}}(n)\,];$$

$$\vec{s}(n) = \vec{s^o}(n) - G_a \hat{\vec{a}}(n);$$

$$\hat{\vec{e}}_a(n) = R\,\vec{s}(n);$$

$$\hat{\vec{a}}(n+1) = \alpha\,\hat{\vec{a}}(n) + \beta\,\hat{\vec{a}}(n-1) + \gamma\,\hat{\vec{e}}_a(n) + \delta\,\hat{\vec{e}}_a(n-1);$$

$$n = n + 1;$$

end cycles

- $\vec{s^o}(n)$ : open-loop measurements
- $G_a$ : interaction (poke, registration) matrix
- $R$ : DM residual reconstruction matrix

# Simulation-Sensor-Reconstruction Grids

## Simulation Grids

- Turbulence profile, $\vec{x}(n)$
- Aperture-plane simulation OPD's for *performance evaluation* : $\vec{\epsilon}(n) = H_x^{\text{SCFoV,sim}}\,\vec{x}(n) - H_a^{\text{SCFoV,sim}}\,\hat{\vec{a}}(n)$

  Note: typically $8\times$ finer resolution than sensor grids

## Sensor Grids

- for MCAO, $\vec{s}(n)$ resolution typically $D/16, D/32$, or $D/64$
- for ExAO, $\vec{s}(n)$ resolution up to $D/128$ or $D/256$

MichiganTech CfAO

# SSR Grids (cont.)

**Reconstruction Grids**

- DM grids. Sampling typically matches sensor sampling.
- for FE-type reconstructors, i.e. when $R = FE$:

(1) reconstructed turbulence profile $\hat{\vec{x}}(n) = E\,\vec{s}(n)$

(2) reconstructed aperture-plane OPD $H_x^{\text{SCFoV}}\,\hat{\vec{x}}(n)$

$2\times$ finer resolution than DM grids

# Aperture-plane transfer matrices

- Based on bilinear influence functions:

$$\left[H^{\mathrm{SCFoV}}\{\mathrm{iobs},\,\mathrm{ips}\}^{dr}_{dr'}\right]_{k\,l} = h^{(l)}(\vec{r}^{(k)}) = h\left[(\vec{r}^{(k)} - \vec{r'}^{(l)}(z))/dr'\right]$$

where $\vec{r'}^{(l)}(z) = \vec{r}^{(l)} + z\vec{\theta}\{\mathrm{iobs}\}$ is the coordinate of grid point $l$ in the transverse plane at range $z$ and $\vec{r}^{(l)}$ is the corresponding aperture-plane coordinate.

Bilinear splines:

$$h(\vec{r}) = \begin{cases} (1-|x|)\,(1-|y|) & \text{if } |x|<1 \text{ and } |y|<1 \\ 0 & \text{else} \end{cases}$$

# *Aperture-plane transfer matrices (cont.)*

Notation:

- $H_x^{\text{SCFoV,sim}}$ maps simulated profile $\vec{x}(n)$ to aperture-plane simulation grid (perf eval)
- $H_a^{\text{SCFoV,sim}}$ maps DM OPD $\hat{\vec{a}}(n)$ to aperture-plane simulation grid (perf eval)
- For FE-type reconstructors:
  - $H_x^{\text{SCFoV}}$ maps $\hat{\vec{x}}(n) = E\,\vec{s}(n)$ to aperture-plane reconstruction grid
  - $H_a^{\text{SCFoV}\dagger}$ maps $H_x^{\text{SCFoV}}\,\hat{\vec{x}}(n)$ to DM grids

# Control Algorithm: Performance Eval

$n = 0$; $\hat{\vec{a}}(0) = \vec{0}$;

begin cycles

$\Longrightarrow$ $[\text{PSF}(n), \text{OTF}(n), \sigma^2(n)] = \text{perfevl}[\,\vec{x}(n), \hat{\vec{a}}(n)\,]$;

$\vec{s}(n) = \vec{s^o}(n) - G_a \hat{\vec{a}}(n)$;

$\hat{\vec{e}}_a(n) = R\,\vec{s}(n)$;

$\hat{\vec{a}}(n+1) = \alpha\,\hat{\vec{a}}(n) + \beta\,\hat{\vec{a}}(n-1) + \gamma\,\hat{\vec{e}}_a(n) + \delta\,\hat{\vec{e}}_a(n-1)$;

$n = n + 1$;

end cycles

# Performance Evaluation

- $\vec{\epsilon}(n) = H_x^{\text{SCFoV,sim}}\,\vec{x}(n) - H_a^{\text{SCFoV,sim}}\,\hat{\vec{a}}(n)$
- $\sigma^2_{\text{ol}}(n) = \|H_x^{\text{SCFoV,sim}}\,\vec{x}(n)\|^2_{\mathcal{W}^{\text{sim}}}$ , $\sigma^2_{\text{cl}}(n) = \|\vec{\epsilon}(n)\|^2_{\mathcal{W}^{\text{sim}}}$
- $\text{PSF}(n) = |\mathcal{F}\{e^{i\,\vec{\epsilon}(n)}\,W_A\}|^2$ , $\text{OTF}(n) = \mathcal{F}\{\text{PSF}(n)\}$
- $\text{SR}(n)$ , $\text{FWHM}(n)$,...

$\mathcal{W}$ is the FoV averaging matrix (SPSD). Based on bilinear splines, it provides accurate modeling of aperture-plane OPD and edge effects. Includes piston removal.

$$\mathcal{W} = \begin{cases} W = W' - \vec{v}\,\vec{v}^T & \text{if zero SCFoV} \\ \text{Diag}\left(\omega^{(1)}W, \cdots, \omega^{(N_{\text{obs}})}W\right) & \text{else} \end{cases}$$

# Control Algorithm: Sensor Measurements

$n = 0$; $\hat{\vec{a}}(0) = \vec{0}$;

begin cycles

$[\text{PSF}(n), \text{OTF}(n), \sigma^2(n)] = \text{perfevl}[\,\vec{x}(n), \hat{\vec{a}}(n)\,]$;

$\Longrightarrow$ $\vec{s}(n) = \vec{s^o}(n) - G_a \hat{\vec{a}}(n)$;

$\hat{\vec{e}}_a(n) = R\,\vec{s}(n)$;

$\hat{\vec{a}}(n+1) = \alpha\,\hat{\vec{a}}(n) + \beta\,\hat{\vec{a}}(n-1) + \gamma\,\hat{\vec{e}}_a(n) + \delta\,\hat{\vec{e}}_a(n-1)$;

$n = n + 1$;

end cycles

# Sensor Measurements

$$\vec{s}(n) = \vec{s^o}(n) - G_a \hat{\vec{a}}(n)$$

$$\vec{s^o}(n) = G_x^{\text{sim}}\, \vec{x}(n) + \vec{\eta}(n)$$

$$\vec{s}\{\text{iwfs}\}(n) = \sum_{\text{ips}} G_x^{\text{sim}}\{\text{iwfs}, \text{ips}\}\, \vec{x}\{\text{ips}\}(n) + \vec{\eta}\{\text{iwfs}\} - \sum_{\text{idm}} G_a\{\text{iwfs}, \text{idm}\}\, \hat{\vec{a}}\{\text{idm}\}(n)$$

Note: $\boxed{G_x^{\text{sim}} = \Gamma_x^{\text{sim}}\, H_x^{\text{GSFoV}}}$ and $\boxed{G_a = \Gamma_a\, H_a^{\text{GSFoV}}}$, where the $\Gamma_x^{\text{sim}}$ and $\Gamma_a$ map aperture-plane OPD's to WFS measurements

# Turbulence-to-WFS Influence Matrix

$$\left[G_x^{\text{sim}}\{\text{iwfs, ips}\}\right]_{k\,l} = \int d^2\vec{r}\,W_{\text{SA}}^{(k)}\{\text{iwfs}\}(\vec{r})\vec{\alpha}\cdot\vec{\nabla}h^{(l)}(\vec{r})$$

$$= \oint_{\xi_1}^{\xi_2} h^{(l)}(\vec{r}(\xi))\;\vec{\alpha}\cdot\vec{n}(\xi)\,d\xi$$

- $h^{(l)}(\vec{r}) = h\left[(\vec{r} - \vec{r'}^{(l)}(z))/dr'\right]$, $dr'$ = sim grids resolution
- $\vec{r'}^{(l)}(z) = \gamma\{\text{iwfs}\}\,\vec{r}^{(l)} + z^{(l)}\vec{\theta}\{\text{iwfs}\}$ , $\gamma\{\text{iwfs}\} = (1 - z^{(l)}/z_{\text{GS}}\{\text{iwfs}\})$
- $\vec{\alpha} = (\alpha_x, \alpha_y)$ is the measurement direction
- $\vec{n}(\xi) = (dy(\xi)/d\xi\,, -dx(\xi)/d\xi)$

- Same eqs for $G_a\{\text{iwfs, idm}\}$ but with $dr'$ = DM resolutions

# Control Algorithm: Reconstructor

$n = 0$; $\hat{\vec{a}}(0) = \vec{0}$;

begin cycles

$[\text{PSF}(n), \text{OTF}(n), \sigma^2(n)] = \text{perfevl}[\,\vec{x}(n), \hat{\vec{a}}(n)\,]$;

$\vec{s}(n) = \vec{s^o}(n) - G_a \hat{\vec{a}}(n)$;

$\Longrightarrow$ $\hat{\vec{e}}_a(n) = R\,\vec{s}(n)$;

$\hat{\vec{a}}(n+1) = \alpha\,\hat{\vec{a}}(n) + \beta\,\hat{\vec{a}}(n-1) + \gamma\,\hat{\vec{e}}_a(n) + \delta\,\hat{\vec{e}}_a(n-1)$;

$n = n + 1$;

end cycles

MichiganTech CfAO

# Filtered Least-Squares Reconstructor

$$
\begin{aligned}
\hat{\vec{e}}_a(n) &= R\,\vec{s}(n) \\
&= \arg\min_{\vec{e}_a(n)} \left\{ \|\vec{s}(n) - G_a\vec{e}_a(n)\|^2_{C_\eta^{-1}} + \|\vec{e}_a(n)\|^2_{P_a^{\mathsf{GSFoV}}} \right\} \\
&= \left(G_a^T C_\eta^{-1} G_a + P_a^{\mathsf{GSFoV}}\right)^{-1} G_a^T C_\eta^{-1}\,\vec{s}(n)
\end{aligned}
$$

- $P_a^{\mathsf{GSFoV}} = V_a V_a^T$, where $V_a = [\vec{v}_a^{(1)}, \cdots, \vec{v}_a^{(p)}]$. Range$(V_a)$ spans constrained modes, e.g. waffle for SCAO, cancelling tip/tilt for MCAO NGS, cancelling tip/tilt, astigmatism and defocus for MCAO LGS, i.e.

$$
\sum_{\mathsf{idm}} H_a^{\mathsf{GSFoV}}\{\mathsf{iobs},\mathsf{idm}\}\vec{v}_a\{\mathsf{idm}\} = \vec{\mathcal{Z}}_1\{\mathsf{iobs}\} \text{ or } \vec{f}\{\mathsf{iobs}\} \in \mathrm{Null}(\Gamma_a)
$$

Note: $\|\vec{e}_a(n)\|^2_{P_a^{\mathsf{GSFoV}}} = \|V_a\vec{e}_a(n)\|^2$

# Minimum Variance Reconstructor

- Open-loop reconstructor:

- Estimation step:

$$\hat{\vec{e}}_x(n) = E\,\vec{s}(n) = \arg\min_{\vec{e}_x(n)} \left\{ \|\vec{s}(n) - G_x \vec{e}_x(n)\|^2_{C_\eta^{-1}} + \|\vec{e}_x(n)\|^2_{C_x^{-1}} + \|\vec{e}_x(n)\|^2_{P_x^{\mathrm{GSFoV}}} \right\}$$

$$= \left(G_x^T C_\eta^{-1} G_x + C_x^{-1} + P_x^{\mathrm{GSFoV}}\right)^{-1} G_x^T C_\eta^{-1}\,\vec{s}(n)$$

- Fitting step:

$$\hat{\vec{e}}_a(n) = F\,\hat{\vec{e}}_x(n) = \arg\min_{\vec{e}_a(n)} \left\{ \|H_x^{\mathrm{SCFoV}} \hat{\vec{e}}_x(n) - H_a^{\mathrm{SCFoV}} \vec{e}_a(n)\|^2_{\mathcal{W}} + \|\vec{e}_a(n)\|^2_{P_a^{\mathrm{SCFoV}}} \right\}$$

$$= \left(H_a^{\mathrm{SCFoV}\,T} \mathcal{W} H_a^{\mathrm{SCFoV}} + P_a^{\mathrm{SCFoV}}\right)^{-1} H_a^{\mathrm{SCFoV}\,T} \mathcal{W} H_x^{\mathrm{SCFoV}}\,\hat{\vec{e}}_x(n)$$

# Minimum Variance Reconstructor (cont.)

- E-step GSFoV-projector:

$$P_x^{\text{GSFoV}} = V_x V_x^T \,, \quad V_x = [\vec{v}_x^{(1)}, \cdots, \vec{v}_x^{(q)}]$$

$$\sum_{\text{ips}} H_x^{\text{GSFoV}}\{\text{iwfs, ips}\}\vec{v}_x\{\text{ips}\} = \vec{\mathcal{Z}}_1\{\text{iwfs}\} \text{ or } \vec{f}\{\text{iwfs}\} \in \text{Null}(\Gamma_x)$$

- F-step SCFoV-projector:

$$P_a^{\text{SCFoV}} = V_a V_a^T \,, \quad V_a = [\vec{v}_a^{(1)}, \cdots, \vec{v}_a^{(r)}]$$

$$\sum_{\text{idm}} H_a^{\text{SCFoV}}\{\text{iobs, idm}\}\vec{v}_a\{\text{idm}\} = \vec{\mathcal{Z}}_1\{\text{iobs}\} \text{ or } \vec{f}\{\text{iobs}\} \in \text{Null}(\Gamma_a)$$

- MVR's $P_a^{\text{SCFoV}}$ is identical to LSR's $P_a^{\text{SCFoV}}$ if $H_a^{\text{SCFoV}} = H_a^{\text{GSFoV}}$
- $q >> r$, i.e. dim Null($\Gamma_x$) $>>$ dim Null($\Gamma_a$)
- In practive, projectors $P_a^{\text{SCFoV}}$, $P_x^{\text{GSFoV}}$ not known exactly

# Approximate Closed-Loop Reconstructor

- Estimation step:

$$\hat{\vec{e}}_x(n) = E\,\vec{s}(n) = \arg\min_{\vec{e}_x(n)} \left\{ \|\vec{s}(n) - G_x\vec{e}_x(n)\|^2_{C_\eta^{-1}} + \|\vec{e}_x(n)\|^2_{C_{x_d}^{-1}} + \|\vec{e}_x(n)\|^2_{P_x^{\text{GSFoV}}} \right\}$$

$$= \left(G_x^T C_\eta^{-1} G_x + C_{x_d}^{-1} + P_x^{\text{GSFoV}}\right)^{-1} G_x^T C_\eta^{-1}\,\vec{s}(n)$$

where $x_d(n) = x(n) - x(n - n_\tau)$ models an $n_\tau$-cycle lag, and

$$C_{x_d} = \mathcal{F}^{-1}\text{diag}(\langle(2\,\pi\,n_\tau\vec{v}\cdot\vec{\kappa})^2\rangle\|\vec{\kappa}\|^{-11/3})\mathcal{F}$$

If wind speed direction $\vec{v}$ is uniformly distributed in $[0, 2\pi]$,
$\langle(2\,\pi\,n_\tau\vec{v}\cdot\vec{\kappa})^2\rangle = (1/2)(2\pi n_\tau\|\vec{v}\|)^2\|\vec{\kappa}\|^2$

- Fitting step as for MVR

# Control Algorithm: Loop Filtering

$n = 0$; $\hat{\vec{a}}(0) = \vec{0}$;

begin cycles

$[\text{PSF}(n), \text{OTF}(n), \sigma^2(n)] = \text{perfevl}[\,\vec{x}(n), \hat{\vec{a}}(n)\,]$;

$\vec{s}(n) = \vec{s^o}(n) - G_a \hat{\vec{a}}(n)$;

$\hat{\vec{e}}_a(n) = R\,\vec{s}(n)$;

$\Longrightarrow$ $\hat{\vec{a}}(n+1) = \alpha\,\hat{\vec{a}}(n) + \beta\,\hat{\vec{a}}(n-1) + \gamma\,\hat{\vec{e}}_a(n) + \delta\,\hat{\vec{e}}_a(n-1)$;

$n = n + 1$;

end cycles

- Loop filtering can be rewritten $\hat{\vec{a}}(n+1) = g(n) \star \hat{\vec{e}}_a(n)$ with $g(z) = (\gamma z + \delta)/(z^2 - \alpha z - \beta)$. Integrator: $\alpha + \beta = 1$
- For FE-recons, filtering can be applied on $\hat{\vec{e}}_x$ or $\hat{\vec{e}}_a$ since gain coeffs are scalar multiples of the identity matrix

# Pseudo Open-Loop Reconstructor

Inspired by open-loop MVR with loop filtering applied on $\hat{\vec{e}}_x$:

$$\vec{s^o}(n) = \vec{s}(n) + G_a \hat{\vec{a}}(n)$$

$$\hat{\vec{e}}_x(n) = \arg\min_{\vec{e}_x(n)} \Big\{ \| \vec{s^o}(n) - G_x(\hat{\vec{x}}(n) + \vec{e}_x(n)) \|^2_{C_\eta^{-1}} + \| \hat{\vec{x}}(n) + \vec{e}_x(n) \|^2_{C_x^{-1}}$$

$$+ \| \hat{\vec{x}}(n) + \vec{e}_x(n) \|^2_{P_x^{\text{GSFoV}}} \Big\}$$

$$= A^{-1} G_x^T C_\eta^{-1} \left( \vec{s}(n) + \vec{s'}(n) \right) - A^{-1} (C_x^{-1} + P_x^{\text{GSFoV}}) \, \hat{\vec{x}}(n)$$

where

$$\vec{s'}(n) = (G_a F - G_x) \, \hat{\vec{x}}(n) \, , \;\; A = G_x^T C_\eta^{-1} G_x + C_x^{-1} + P_x^{\text{GSFoV}}$$

Note: $A^{-1} G_x^T C_\eta^{-1} = E$ = MVR E-matrix

# POLR (cont.)

Control Algorithm: *Version 1*

$$n = 0;\ \hat{\vec{a}}(0) = \vec{0};\ \hat{\vec{x}}(0) = \vec{0};$$

begin cycles

$$[\mathsf{PSF}(n), \mathsf{OTF}(n), \sigma^2(n)] = \mathsf{perfevl}[\,\vec{x}(n), \hat{\vec{a}}(n)\,];$$

$$\vec{s}(n) = \vec{s^o}(n) - G_a \hat{\vec{a}}(n);$$

$$\vec{s'}(n) = (G_a F - G_x)\hat{\vec{x}}(n);$$

$$\hat{\vec{e}}_x(n) = E\,(\vec{s}(n) + \vec{s'}(n)) - A^{-1}(C_x^{-1} + P_x^{\mathsf{GSFoV}})\hat{\vec{x}}(n);$$

$$\hat{\vec{x}}(n+1) = \alpha\,\hat{\vec{x}}(n) + \beta\,\hat{\vec{x}}(n-1) + \gamma\,\hat{\vec{e}}_x(n) + \delta\,\hat{\vec{e}}_x(n-1);$$

$$\hat{\vec{a}}(n+1) = F\hat{\vec{x}}(n+1);$$

$$n = n + 1;$$

end cycles

# POLR (cont.)

Control Algorithm: *Version 2*

$n = 0$; $\hat{\vec{a}}(0) = \vec{0}$; $\hat{\vec{x}}(0) = \vec{0}$;

begin cycles

$[\text{PSF}(n), \text{OTF}(n), \sigma^2(n)] = \text{perfevl}[\,\vec{x}(n), \hat{\vec{a}}(n)\,]$;

$\vec{s}(n) = \vec{s^o}(n) - G_a \hat{\vec{a}}(n)$;

$\hat{\vec{e}}_x(n) = E\vec{s}(n)$;

$\hat{\vec{x}}(n+1) = [\alpha I + \gamma M]\,\hat{\vec{x}}(n) + [\alpha I + \delta M]\,\hat{\vec{x}}(n-1) + \gamma \hat{\vec{e}}_x(n) + \delta \hat{\vec{e}}_x(n-1)$;

where $M = E(G_a F - G_x) - A^{-1}(C_x^{-1} + P_x^{\text{GSFoV}})$

$\hat{\vec{a}}(n+1) = F\hat{\vec{x}}(n+1)$;

$n = n + 1$;

end cycles

# POLR (cont.)

- POLR is not a conventional matrix multiply reconstructor: requires real-time knowledge of $\hat{\vec{x}}(n)$
- POLR requires knowledge of interation matrix $G_a$
- $E = \left(G_x^T C_\eta^{-1} G_x + C_x^{-1} + P_x^{\text{GSFoV}}\right)^{-1} G_x^T C_\eta^{-1} = A^{-1} G_x^T C_\eta^{-1}$ = open-loop MVR E-matrix
- Computational complexity identical to MVR
- $G_a F - G_x = \Gamma_a H_a^{\text{GSFoV}} H_a^{\text{SCFoV}\dagger} H_x^{\text{SCFoV}} - \Gamma_x H_x^{\text{GSFoV}} \neq 0$

# Transfer Matrix and Loop Stability

- LSR:

$$\hat{\vec{a}}^o(z) = R\vec{s}^o(z)$$
$$\hat{\vec{a}}(z) = T_a(z)\hat{\vec{a}}^o(z)$$
$$T_a(z) = g(z)\,(I + g(z)RG_a)^{-1}$$
$$g(z) = (\gamma z + \delta)/(z^2 - \alpha z - \beta)$$
$$RG_a = I - P_a^{\text{GSFoV}} = I - V_a V_a^T$$

- Given $RG_a = U\text{diag}(\lambda_i)U^{-1}$,

$$T_a(z) = U\text{diag}\,[\,g(z)/(1 + g(z)\lambda_i)\,]\,U^{-1}$$
$$U^{-1}\hat{\vec{a}}(z) = \text{diag}\,[\,g(z)/(1 + g(z)\lambda_i)\,]\,U^{-1}\hat{\vec{a}}^o(z)$$

- Representative filter: $\gamma = 0\,,\delta = \beta = \alpha = 1/2$, yielding $g(z) = 1/[2(z-1)(z+1/2)]$

# *Loop Stability (cont.)*

- Poles $\{z_\star \mid 1 + g(z_\star)\lambda_i = 0\}$ are $z_\star = (1 \pm \sqrt{9 - 8\lambda_i})/4$
- System is stable if and only if all poles are located inside unit circle, i.e. $|z_\star| \leq 1$
- LSR is stable since $\lambda_i = 0, 1$ and thus $z_\star = 1, \pm 1/2$

- MVR - ACLR:
  - E-step:

$$
\begin{aligned}
\hat{\vec{x^o}}(z) &= E\vec{s^o}(z) \\
\hat{\vec{x}}(z) &= T_x(z)\hat{\vec{x^o}}(z) \\
T_x(z) &= g(z)\left(I + g(z)EG_aF\right)^{-1}
\end{aligned}
$$

# Loop Stability (cont.)

- F-step:

$$\hat{\vec{a}}(z) = F\,\hat{\vec{x}}(z) = F\,T_x(z)\,\vec{x}^{\hat{o}}(z)$$

$$F\,T_x(z) = g(z)F\,(I + g(z)EG_aF)^{-1} = g(z)\,(I + g(z)RG_a)^{-1}\,F$$

$$\Longrightarrow \quad \hat{\vec{a}}(z) = T_a(z)\vec{a}^{\hat{o}}(z) \quad \text{(as for LSR)}$$

$$T_a(z) = g(z)\,(I + g(z)FEG_a)^{-1}\ ,\ \vec{a}^{\hat{o}}(z) = F\vec{x}^{\hat{o}}(z)$$

- Since $FEG_a$ has complex eigenvalues $\lambda_i$ that in general won't satisfy the stability criterion $|(\,1 \pm \sqrt{9 - 8\lambda_i}\,)/4| < 1$, MVR-ACLR will be unstable.

- Stability recovered for SCAO by slaving edge actuators and keeping only subaps at least 50% illuminated. Demonstrated in simulations and on the sky @ PALAO.

# Loop Stability (cont.)

- POLR:

$$\begin{aligned} M &= EG_aF - EG_x - A^{-1}(C_x^{-1} + P_x^{\text{GSFoV}}) \\ T_x(z) &= g(z)\,[\,I + g(z)(EG_aF - M)\,]^{-1} \\ \hat{\vec{a}}(z) &= F\,\hat{\vec{x}}(z) = F\,T_x(z)\,\hat{\vec{x^o}}(z) \end{aligned}$$

- $M = 0$ yields MVR. Given $EG_aF - M = U\text{diag}(\lambda_i)U^{-1}$,

$$T_x(z) = U\text{diag}\,[\,g(z)/(1 + g(z)\lambda_i)\,]\,U^{-1}$$

- POLR doesn't have simple $T_a(z)$ s.t. $F\,T_x(z) = T_a(z)\,F$
- Simulations demonstrate that POLR is efficient but becomes unstable with little uncertainty in $G_a$
- Real-time implementation worth considering

MichiganTech CfAO

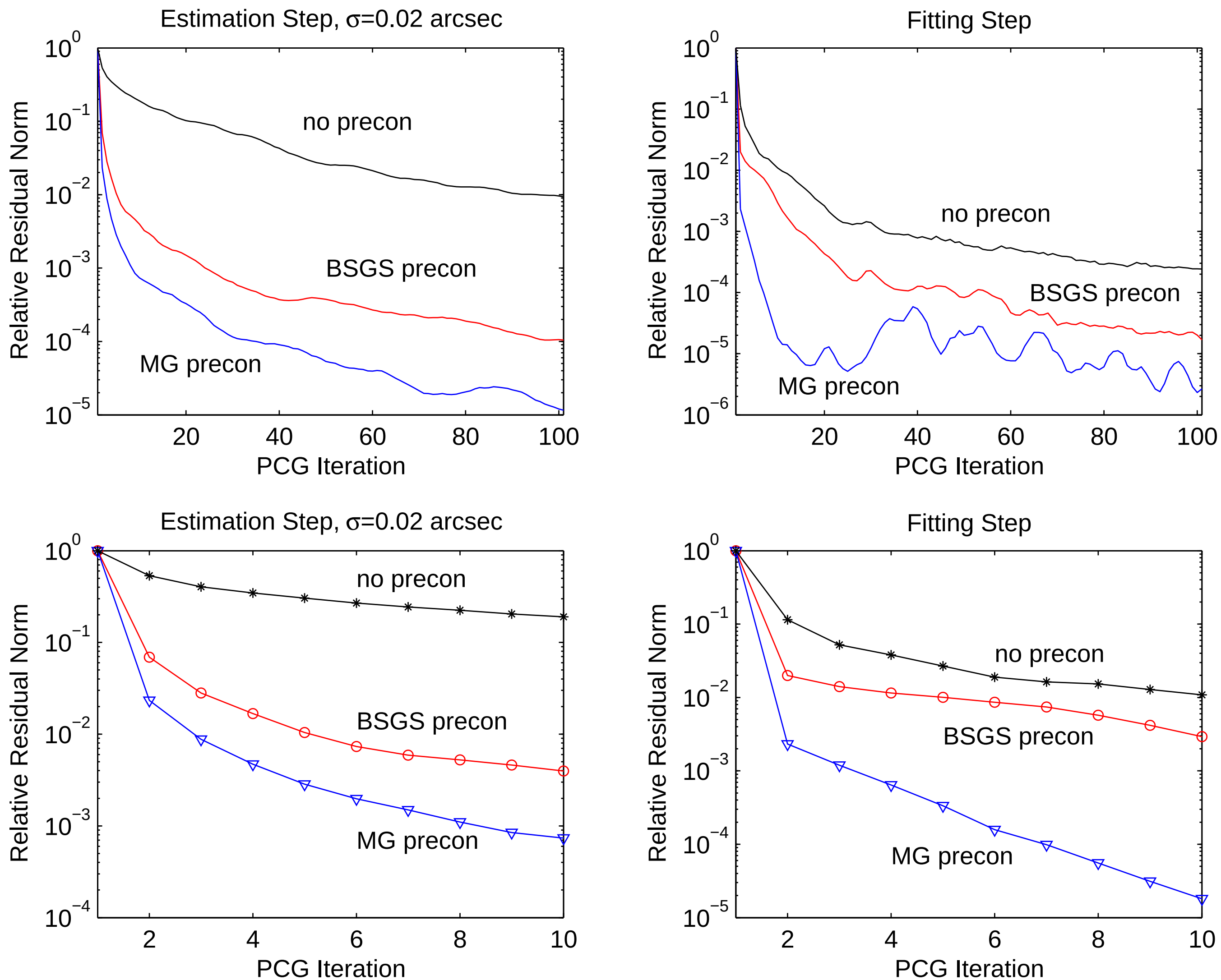
# Sparse MVR solver

- Sparse layer-oriented multigrid-preconditioned conjugate gradient (MGCG) solver for MVR's E- and F-steps:
  - Layer-by-layer OPD estimation
  - 1 BSGS smoothing iteration for layer decoupling
  - LGS tip/tilt uncertainty leads to spatially colored measurement noise handled by rank-1 projector
  - LGS focus uncertainty can be handled similarly with an additional rank-1 projector
  - E- and F-systems solved using sparse plus low-rank inversion lemma (Sherman-Morrison)

# Sample 32m problem description

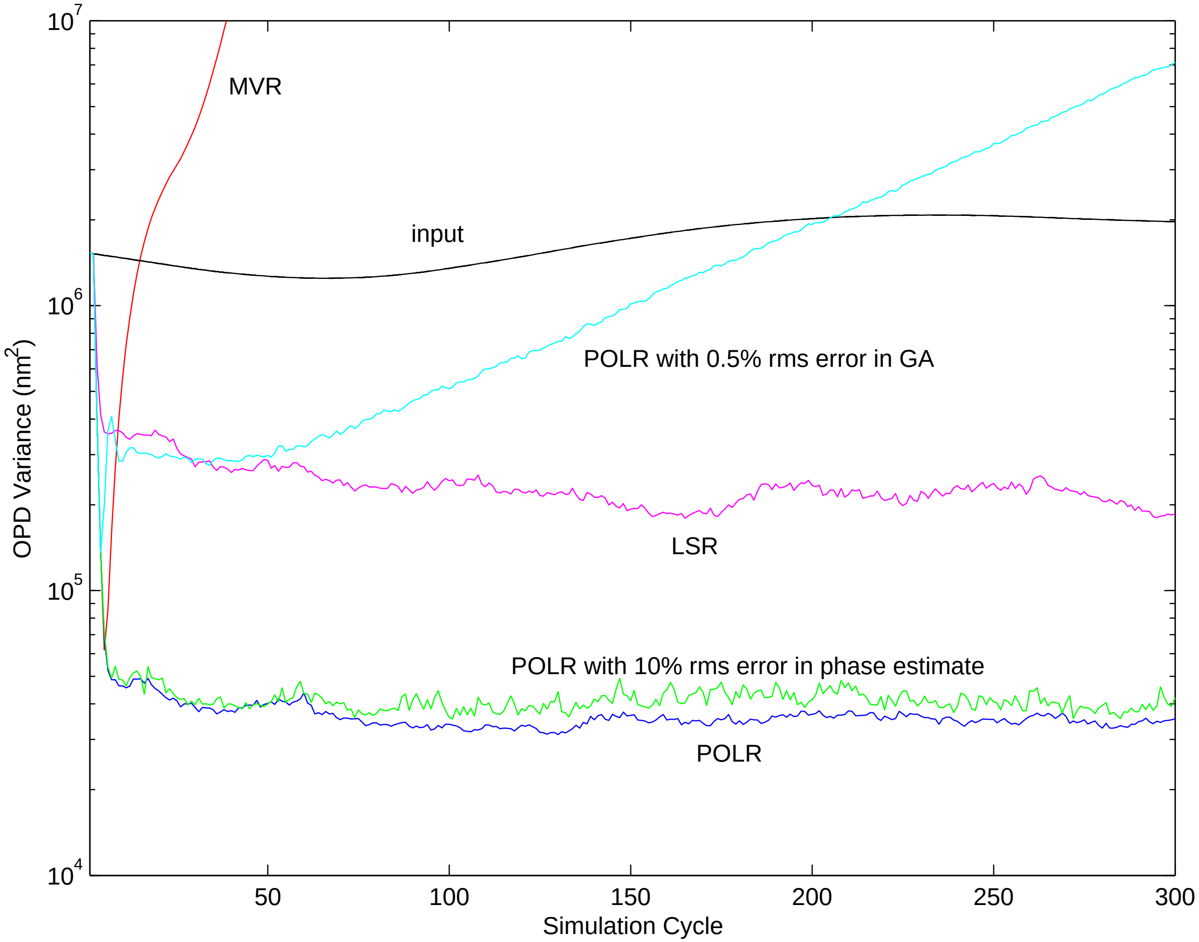
- Cerro Pachon 6 layer median turbulence profile ranging from $0$m to $15,460$m scaled to yield $r_0 = 16$cm @ $0.5\mu$m. $\theta_0 = 2.65''$. $64$m phase screens. Reconstructed PS1-PS4 oversample $2\times$ subaps; recons PS5-PS6 match subaps resolution. $70,942$ grid points reconstructed
- 5 LGS's at the four edges and center of a 1″ GSFoV; $64 \times 64$ subapertures each; $33,320$ measurements
- 4 NGS's at mid distance between the four LGS's delimiting the GSFoV $(8$ tip/tilt measurements)
- $3 \times 3$ 1″ SCFoV; Simpson weights
- 3 DM's. DM's 1-2 in Fried config, DM3 $(10,309$m) undersamples $2\times$ subaps. Total of $8,449$ actu

# 32m open-loop convergence results

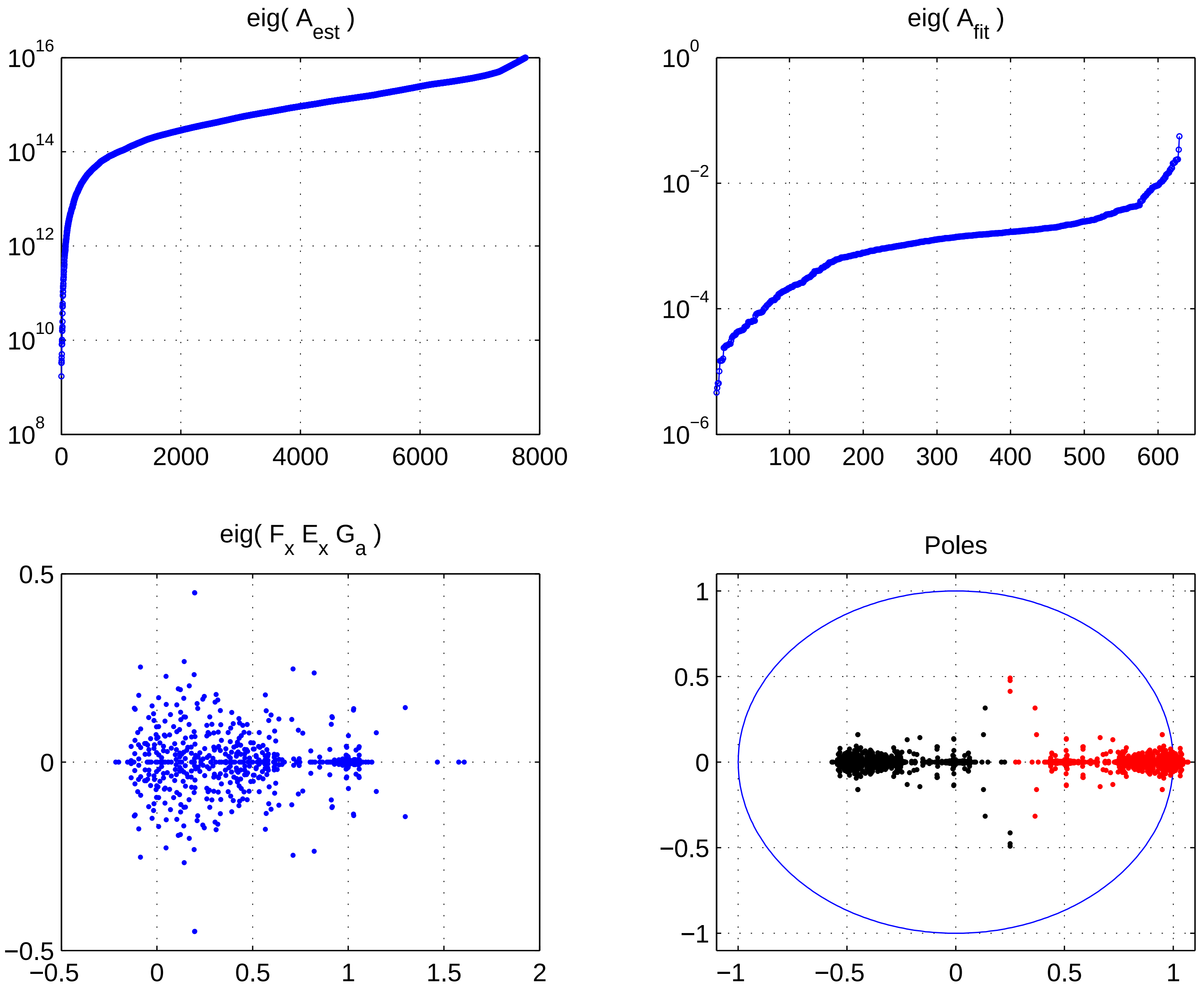


# 8m closed-loop simulation results

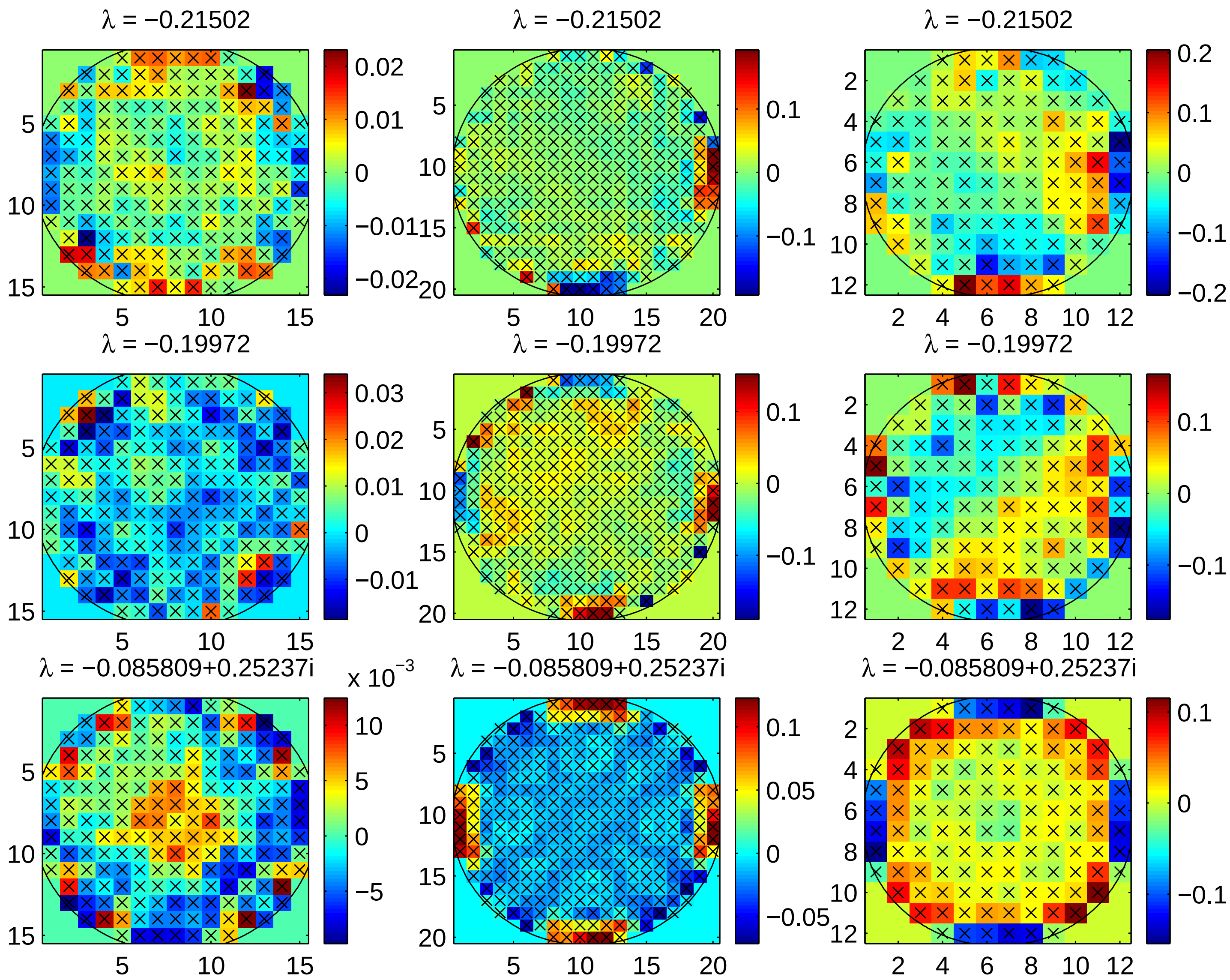
- Same geometry and fields as for 32m but with $16 \times 16$ LGS WFS subaps. $7,826$ reconstructed phase screen grid points; $629$ actu and $2,248$ measurements.



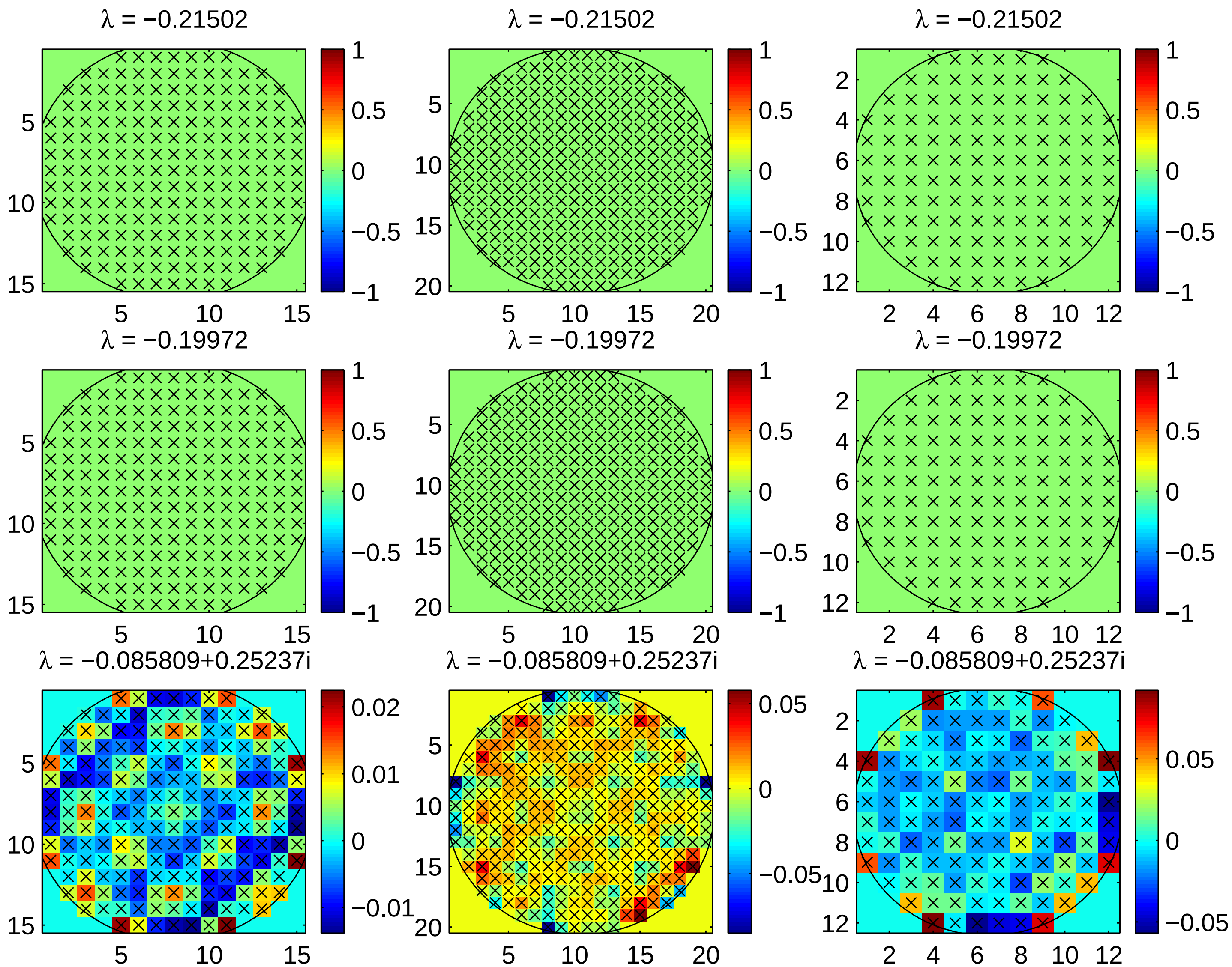
# 8m closed-loop poles



# Real part of sample offending modes



# *Imag part of sample offending modes*



# Constrained MVR

- Minimizes $< \sigma^2(n) > = < \|\vec{\epsilon}(n)\|^2_{\mathcal{W}} >$ subject to 2 linear constraints:
  - $\boxed{RG_a = P_a}$ where $P_a^2 = P_a$ and $P_a^T = M P_a M^{-1}$ with $M = H_a^{\text{SCFoV}\,T} \mathcal{W} H_a^{\text{SCFoV}}$
  - $\boxed{(I - P_a)R = 0}$

$\Longrightarrow \quad R = R_1 + R_2$ with

$$
\begin{aligned}
R_1 &= F E_f \\
E_f &= \left( G_x^T C_{\eta_f}^{-1} G_x + C_{x_f}^{-1} \right)^{-1} G_x^T C_{\eta_f}^{-1} \\
\vec{\eta}_f(z) &= \vec{\eta}(z)\, g(z)\,, \quad \vec{x}_f(z) = \vec{x}\, g(z) \\
R_2 &= (I - R_1 G_a)\, G_a^{\dagger} \\
G_a^{\dagger} &= \left( G_a^T\, C_{s_f}^{-1}\, G_a \right)^{-1} G_a^T\, C_{s_f}^{-1} \\
C_{s_f} &= G_x\, C_{x_f} G_x^T + C_{\eta_f}
\end{aligned}
$$

# Constrained MVR (cont.)

- Obviously stable since $\lambda_i = 0, 1$
- Is there a sparse efficient way to compute it ?
  - $C_{x_f}^{-1} \simeq \mathcal{F}^{-1}$ diag$( \|\vec{\kappa}\|^{11/3} \sqrt{1 + \|\vec{v}\|^2 \|\vec{\kappa}\|^2 / \nu_c} )\mathcal{F}$
  - $C_{\eta_f} \simeq C_\eta$
  - $G_a^\dagger$ in $R_2$ most troublesome
- Optimized range of projector $P_a$ also troublesome