Linear Parameter Varying Control for Laser Tomographic Adaptive Optics

Jesse Cranney[a,b], Jose De Dona[b], Ryan McCloy[c], Visa Korkiakoski[d], and Francois Rigaut[d]

[a]SERC Limited, Canberra, Australia
[b]University of Newcastle, Callaghan, Australia
[c]University of New South Wales, Sydney, Australia
[d]Australian National University, Canberra, Australia

ABSTRACT

Linear control schemes such as LQG have proven to be effective at predicting and correcting for atmospheric turbulence in Adaptive Optics (AO) systems. One drawback of these linear schemes is that the control matrices involved must be recomputed when the AO configuration changes in order to maintain a high performance. We explore one such case in LTAO mode, where the scientific target moves continuously along some known trajectory. We propose a Linear Parameter Varying (LPV) solution to this problem which shows promising results in Yao simulations for a 40x40 SHWFS 6 NGS case. The LPV solution provides a greater flexibility in the useful operating region of LTAO, incurring only a modest increase in computational demand.

Acknowledgments

The authors would like to acknowledge the support of the Cooperative Research Centre for Space Environment Management (SERC Limited) through the Australian Government's Cooperative Research Centre Programme.

1. INTRODUCTION

1.1 Predictive LQG Control Scheme

We discuss here the Linear Quadratic Gaussian (LQG) controller with the so-called *Near-Markov* turbulence model [1, 2]. For the fixed target AO mode, the control law is designed to minimise the residual phase integrated vertically in the direction of the target (or targets, for MCAO/MOAO). For the formulations presented, the state of the turbulence is estimated on a zonal layer-oriented basis under the frozen flow hypothesis along with the near-field approximation for light propagation through each layer of turbulence. Natural Guide Stars (NGS) are used (i.e., the cone effect is ignored), however the extension to Laser Guide Stars (LGS) appears straightforward.

The dynamics of the turbulence are modeled in discrete time as a linear function of the previous state plus additive zero-mean white noise:

$$\phi_{k+1} = A\phi_k + v_k \tag{1}$$

where A provides (see, e.g., [1, 2]) the minimum variance estimate of ϕ_{k+1} given ϕ_k and is equal to:

$$A = \left\langle \phi_{k+1}\phi_k^\mathsf{T} \right\rangle \left\langle \phi_k\phi_k^\mathsf{T} \right\rangle^{-1}$$

where $\langle x \rangle$ denotes the expected value of the random variable, x. The variable ϕ_k in the previous expressions is composed of the spatially discretised samples of the turbulence phases of each layer at time k, vertically concatenated in a single vector. In a zonal basis under the frozen flow assumption, the covariance matrices can be found by evaluating the Von Karman covariance function for each layer with its argument equal to the

Further author information (for correspondence):
Jesse Cranney: E-mail: jesse.cranney@uon.edu.au, Telephone: +61 403 144 392

distance between zonal estimation points (with an appropriate shift due to wind-shift for the $\langle \phi_{k+1}\phi_k^\mathsf{T}\rangle$ matrix). That is:

$$\langle \phi_{k+1}\phi_k^\mathsf{T}\rangle_{i,j} = \begin{cases} C_\ell\left(\sqrt{(x_i + v_{x,i}T - x_j)^2 + (y_i + v_{y,i}T - y_j)^2}\right) & \text{layer}(i) = \text{layer}(j) = \ell \\ 0 & \text{otherwise} \end{cases}$$

and

$$\langle \phi_k\phi_k^\mathsf{T}\rangle_{i,j} = \begin{cases} C_\ell\left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right) & \text{layer}(i) = \text{layer}(j) = \ell \\ 0 & \text{otherwise} \end{cases}$$

The function layer(i) in the previous expressions returns the layer number that the i^th index in the concatenated vector ϕ_k belongs to. $C_\ell(r)$ is the covariance between two points in the ℓ^th layer separated by a distance r. This can be found directly from the following expression [3]:

$$C_\ell(r) = \left(\frac{L_{0,\ell}}{r_{0,\ell}}\right)^{5/3} \frac{\Gamma(11/6)}{2^{5/6}\pi^{8/3}} \left[\frac{24}{5}\Gamma\left(\frac{6}{5}\right)\right]^{5/6} \left(\frac{2\pi r}{L_{0,\ell}}\right)^{5/6} K_{5/6}\left(\frac{2\pi r}{L_{0,\ell}}\right)$$

where $r_{0,\ell}$ and $L_{0,\ell}$ are the Fried parameter and outer scale of the turbulence in layer ℓ respectively, and $\Gamma(\cdot)$ and $K_{5/6}(\cdot)$ are the Gamma function and modified Bessel function of the second kind, order $5/6$, respectively.

The process noise v_k has stationary covariance and, under the standard assumption that v_k and ϕ_k are uncorrelated, is given by:

$$\langle v_k v_k^\mathsf{T}\rangle = \langle \phi_k\phi_k^T\rangle - A\langle \phi_k\phi_k^T\rangle A^\mathsf{T}$$

The measurement of the turbuelence is performed by M different Wave-front Sensors (WFS), one for each guide-star. The wave-front phase from atmospheric turbulence and DMs in each of these directions is integrated and the resulting pupil-plane wave-front is converted to discrete measurements over the pupil, for example, to slopes in the case of a Shack-Hartmann WFS. This process is closely modelled as a linear function of the turbulence state, as well as the correction applied by the DM(s), along with additive zero-mean white noise with known covariance based on the WFS measurement error. The overall output equation is:

$$s_k = C\phi_{k-1} + Du_{k-2} + w_k \tag{2}$$

where s_k contains the measurements (slopes) which can be used from time $t \geq kT$ (T is the sampling period of the control loop), u_k is the command vector applied over time $t \in [kT, (k+1)T)$, w_k is the measurement noise with covariance $\langle w_k w_k^\mathsf{T}\rangle$, C is a linear mapping from turbulence—defined at zonal grid points in the discrete layers of the atmosphere—to the WFS slopes, and D is a linear mapping from the DM commands to the WFS slopes (often called the global *interaction matrix*).

It is standard to invoke the *separation theorem* (see [4]) when considering the estimation of ϕ_k and the choice of command u_k. Such an approach allows one to arrive at the Kalman Filter recursions for the estimation process, and the Linear Quadratic Regulator (LQR) for the control law, which combined in closed-loop give the Linear Quadratic Gaussian (LQG) control law. The closed-loop update equations for the LQG control for the AO application discussed here are:

$$\hat{\phi}_{k+1|k+1} = (A - LC)\hat{\phi}_{k|k} + Ls_{k+1} - LDu_{k-1} \tag{3}$$

$$u_{k+1} = K\hat{\phi}_{k+1|k+1} \tag{4}$$

where L is the observer gain, which in the case of the steady-state Kalman Filter is found from the positive-definite solution, P, of the Discrete Algebraic Ricatti Equation (DARE):

$$P = APA^\mathsf{T} + \langle v_k v_k^\mathsf{T}\rangle - APC^\mathsf{T}\left(\langle w_k w_k^\mathsf{T}\rangle + CPC^\mathsf{T}\right)^{-1} CPA^\mathsf{T}$$

and $L = APC^T\left(\langle w_k w_k^\mathsf{T}\rangle + CPC^\mathsf{T}\right)^{-1}$. The K term is the *control matrix* and for LQR this is found to be the minimising solution of a quadratic cost function which penalises residual turbulence in the direction(s) of the scientific object(s) of interest. For the Laser Tomographic AO case (LTAO), there is only one direction of interest and the single ground-layer conjugated DM aims at correcting the turbulence only in that direction. The corresponding cost function becomes:

$$J(u_k) = \left\| P^{\alpha,\beta}\phi_{k+1} - Q^{\alpha,\beta}u_k \right\|_W^2 \tag{5}$$

where $\|x\|_W^2 = x^\mathsf{T}Wx$ is the squared 2-norm of the vector x, weighted by the matrix W (W is a design choice that allows to give more importance to different regions across the pupil). $P^{\alpha,\beta}$ is a projection matrix, which integrates the turbulence phase contribution from each layer in the direction (α,β) onto the telescope pupil. The $Q^{\alpha,\beta}$ matrix is found in a similar manner, except that it is the projection from the DMs (rather than the turbulence) in the direction α,β. In the special case of the pupil-conjugated DM (for LTAO, for example), the $Q^{\alpha,\beta}$ matrix is the same regardless of correction direction.

Without actuator constraints, the optimal solution to Eqn (5) is given by:

$$u_k = ({Q^{\alpha,\beta}}^\mathsf{T}WQ^{\alpha,\beta})^{-1}{Q^{\alpha,\beta}}^\mathsf{T}WP^{\alpha,\beta}\phi_{k+1}$$

if ${Q^{\alpha,\beta}}^\mathsf{T}WQ^{\alpha,\beta}$ is invertible.

Invoking the separation theorem (see [4]), we then use the estimation $\hat{\phi}_{k+1|k}$ instead of ϕ_{k+1}, or equivalently $\hat{\phi}_{k+1|k} = A\hat{\phi}_{k|k}$ (this latter equality results from the standard derivation of the Kalman Filter, Eqn (3)). The resulting control law is thus:

$$u_k = \underbrace{({Q^{\alpha,\beta}}^\mathsf{T}WQ^{\alpha,\beta})^{-1}{Q^{\alpha,\beta}}^\mathsf{T}WP^{\alpha,\beta}A}_{K(\alpha,\beta)}\hat{\phi}_{k|k} \tag{6}$$

As a practical note, to improve the conditioning of the matrix being inverted, one can increase the resolution of the pupil spatial sampling, or add a regularising term to the matrix before inverting, e.g., ${Q^{\alpha,\beta}}^\mathsf{T}WQ^{\alpha,\beta} + \mu\mathbf{I}$ with $\mu > 0$. The former option tends to improve the expected performance of the control law, while the latter option allows to include a cost-related term to the magnitude of the command vector, which is useful, for example, when there are redundant actuators. Note that neither of these changes will have any effect on the dimensions of the resulting control matrix.

It is also important to note at this point that the estimation strategy (Eqn (3)) does not depend on the direction of the science target (α,β) since that choice only relates to the $K(\alpha,\beta)$ matrix (present in Eqn (6)).

1.2 Linear Parameter Varying Approach

The linear parameter varying (LPV) paradigm was introduced in the PhD thesis of Shamma [5] to provide a framework for the control design approach of gain-scheduling and involves stitching together a group of linear controllers to produce a nonlinear controller for a nonlinear system. These linear controllers are combined in real-time (e.g., via switching) based on available measurements of the varying parameter (see also [6, 7]).

Notice that in our setup the closed-loop system comprising Eqns (1), (2), and (6) is already and LPV system, where the two-dimensional parameter is $\rho \triangleq (\alpha,\beta)$. Furthermore, Eqns (1) and (2) are Linear Time Invariant (LTI), reducing the LPV component to only Eqn (6), which we can rewrite as:

$$u_k = K(\rho)\hat{\phi}_{k|k}$$

For our application, the two-dimensional parameter ρ is bounded in a polytopic region (which corresponds to the region of values of $\rho = (\alpha,\beta)$ belonging to the desired FoV). For the purpose of this paper, we consider such regions to be given by bounding rectangles expressed as $\mathcal{P} = \{\underline{\rho}_i \leq \rho_i \leq \bar{\rho}_i\}$ for $i = 1,2$. A standard approach (see, e.g., [8]) is to embed the above LPV system into a *convex polytopic parameterisation*, whereby the LPV system matrices are expressed as a convex combination of their values at the vertices. We first start by expressing the parameter ρ as a convex combination of its vertex values:

$$\rho = \sum_{i=1}^{4} a_i \rho_i$$

where the convex combination parameters satisfy:

$$\sum_{i=1}^{4} a_i = 1 \tag{7}$$

$$0 \leq a_i \leq 1 \ \forall i \tag{8}$$

Finally, the LPV matrix $K(\rho)$ can be approximated by the convex polytopic parameterisation (in the case when K is an affine function of ρ this representation is exact):

$$\tilde{K}(\rho) \triangleq \sum_{i=1}^{n} a_i K(\rho_i) \approx K(\rho)$$

2. LPV FOR LTAO

Here we discuss LPV specifically in the context of variable target direction LTAO. Firstly, we derive an algorithm which gives a useful operation over a small subset of the total FoV. Secondly, we propose an extension to that approach that instead switches between sub-regions of the FoV, which shows useful performance over the entire FoV without any notable increase in computational demand (compared to the first approach).

2.1 4-Vertex Case

In the context of LTAO, LPV is attractive in that it offers a convenient way of varying the direction of correction of the LTAO setup as often as required without recomputing any control matrices online or changing any hardware, at the cost of a modest increase in online computations. The estimation strategy is unaffected by the LPV approach proposed, so Eqn (3) remains unchanged. The modification is in the control law (Eqn (4)), which for the case of a 4-vertex polytopic region is instead given as the following convex combination:

$$u_k = \sum_{i=1}^{4} a_i u_k^i \tag{9}$$

where u_k^i is the optimal command for correcting in the direction defined by values corresponding to the i^{th} vertex of the convex polytope. For a concrete example in the LTAO case, we consider a desired field of view defined as being a square of width 2θ arcsec around the origin. This region is defined by the four vertices of a square with coordinates:

$$v = (\alpha, \beta) = \{(-\theta, -\theta),\ (-\theta, \theta),\ (\theta, -\theta),\ (\theta, \theta)\}$$

It is clear that for each of these four directions, it is possible to find an appropriate control law (see Eqns (4) & (6)) by evaluating $K_i \triangleq K(\alpha_i, \beta_i)$ at the i^{th} vertex in v. The goal of this LPV approach is to provide a suitable control law for values within the region defined by the vertices v without explicitly recomputing the corresponding control matrix for each direction[*]. Specifically, the method used here is a convex combination of the optimal control laws corresponding to each direction, where the weighting coefficients a_k^i are smoothly calculated based on the distance to the four vertices. The general multi-linear interpolation proposed in Section 1.2 is, in this case:

$$a_i = \left(1 - \frac{|\alpha_i - \alpha|}{2\theta}\right)\left(1 - \frac{|\beta_i - \beta|}{2\theta}\right) \tag{10}$$

[*] At the telescope scales considered here, the time taken to recompute the optimal control law for a given direction is in the order of a few seconds on a single GPU, whereas the AO sampling period is in the order of milliseconds. This implies that recomputing the control law *online* is not practical within current technological limitations.

which can be easily verified as satisfying the requirements of a convex combination (Eqns (7) & (8)) when $(\alpha,\beta) \in [-\theta,\theta] \times [-\theta,\theta]$. Substituting Eqns (4) & (10) into (9), we get:

$$u_k = \underbrace{\left(\sum_{i=1}^{4} \left(1 - \frac{|\alpha_i - \alpha|}{2\theta}\right) \left(1 - \frac{|\beta_i - \beta|}{2\theta}\right) K_i \right)}_{\tilde{K}(\alpha,\beta)} \hat{\phi}_k = \tilde{K}(\alpha,\beta)\hat{\phi}_k \tag{11}$$

As $\theta \to 0$, it is clear that all four vertices tend toward the origin, and in the limiting case, $\tilde{K}(\alpha,\beta) = K(0,0)$. As θ grows, the differences between the resulting control laws at each vertex are more pronounced and, because the dependency of K with (α,β) is non-linear, the difference between the optimal value $K(\alpha,\beta)$ and the convex combination also becomes significant. This indicates (and is verified in simulation) that there exists a direct trade-off between performance away from the vertices and the total area enclosed by the vertices.

Note that in the 2-dimensional case discussed here, the LPV approach can be considered as a bi-linear interpolation of the control matrix between the exact solutions found at the vertices v.

2.2 N^2-Vertex Case

For the previous case, the performance is shown in the simulations of Section 3 to be best when the square defined by v is smallest. This suggests that rather than increasing the size of the square over which the convex combination is computed, one would expect better performance by smoothly switching between small squares, where the set of small squares spans a large region. This is achieved by defining the extended vertex vector $\tilde{v}$ as all vertices of a $N \times N$ square grid with pitch 2θ. That is:

$$\tilde{v} = (\alpha,\beta) = \{(2\theta(i-(N-1)/2), 2\theta(j-(N-1)/2)),\ \forall (i,j) \in [0,1,\ldots,N-1] \times [0,1,\ldots,N-1]\}$$

Now, the control matrix K_i is evaluated at each vertex $\tilde{v}_i$, and the command vector is calculated using the unique four vertices that enclose the desired direction (α,β):

$$u_k = \sum_{i=1}^{N^2} a_i K_i \hat{\phi}_k \tag{12}$$

where

$$a_i = \begin{cases} \left(1 - \frac{|\alpha_i - \alpha|}{2\theta}\right)\left(1 - \frac{|\beta_i - \beta|}{2\theta}\right) & \text{if } \left(\frac{|\alpha_i - \alpha|}{2\theta} < 1\right) \wedge \left(\frac{|\beta_i - \beta|}{2\theta} < 1\right) \\ 0 & \text{otherwise} \end{cases}$$

Note that the sum in Eqn (12) has the same convex-combination properties as Eqn (11), and apart from finding the active vertices, there are no additional computations required in the implementation.

3. SIMULATIONS

All simulations presented are performed in Yao [9], where the simulation parameters are shown in Table 1.

3.1 4-Vertex Case

For the case of a single square region of the FoV, the LPV LTAO performance is expected to decrease as the region enclosed by the square increases. As such, we analyse the performance of the entire closed-loop system in a numerical end-to-end simulation for an increasing region enclosed by the vertices. Figure 1 shows for each case (which are different only by the area of the region they attempt to operate in) the averaged on-target performance over 10000 samples, and Figure 2 shows the instantaneous performance at a given sample during the sequence. The object moves smoothly along the trajectory shown according to the following function:

$$\begin{aligned} \alpha &= R\cos(\theta/4)\cos(\theta) \\ \beta &= R\cos(\theta/4)\sin(\theta) \\ \theta &= 2\pi k/1000 \end{aligned}$$

Parameter	Value
#WFS = #GS	6
#subapertures/WFS	40 × 40 (1240 active)
actuators/DM	41 × 41 (1313 active)
guide star position	17” radius (uniformly separated by 60°)
wavelength λ	650nm
Fried parameter r_0	0.13m
diameter	8m
control period T	2ms
control delay	2 frames
GS Flux	187-320 photons/subap/frame
turbulence altitudes	$[0, 3, 6, 13]$km
turbulence wind-speed	$[15\angle 0°, 20\angle 60°, 25\angle 120°, 20\angle 180°]$m/s
turbulence weight	$[0.4, 0.3, 0.2, 0.1]$

Table 1. Simulation Parameters

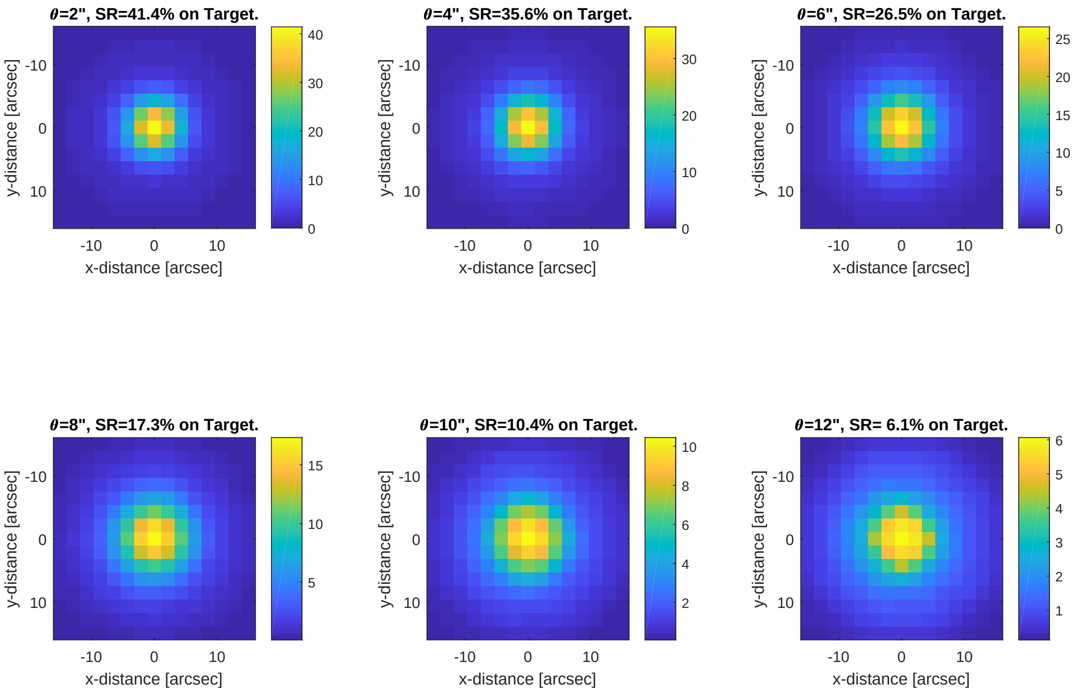

Figure 1. 4-vertex case, on-target SR averaged over 10000 samples for target within region of width 2θ arcsec.

where R is the width of the useful FoV and k is the sample number (corresponding to time $t = kT$).

It can be appreciated in Figures 1 and 2 that the highest on-target SR is, as expected, achieved when the area covered is smallest and it decreases monotonically with the growth of the area. Notice that in Figure 2 there is an offset between the desired target direction and the peak SR location that is negligible for small areas but becomes noticeable as the area coverage increases. This is as expected, since the optimal control gain $K(\alpha, \beta)$ is a non-linear function of (α, β) and hence it does not necessarily coincide with the convex-combination of the optimal values computed at the vertices. This provides motivation for increasing the number of vertices, thereby decreasing the area of every sub-region, as studied in the next section.

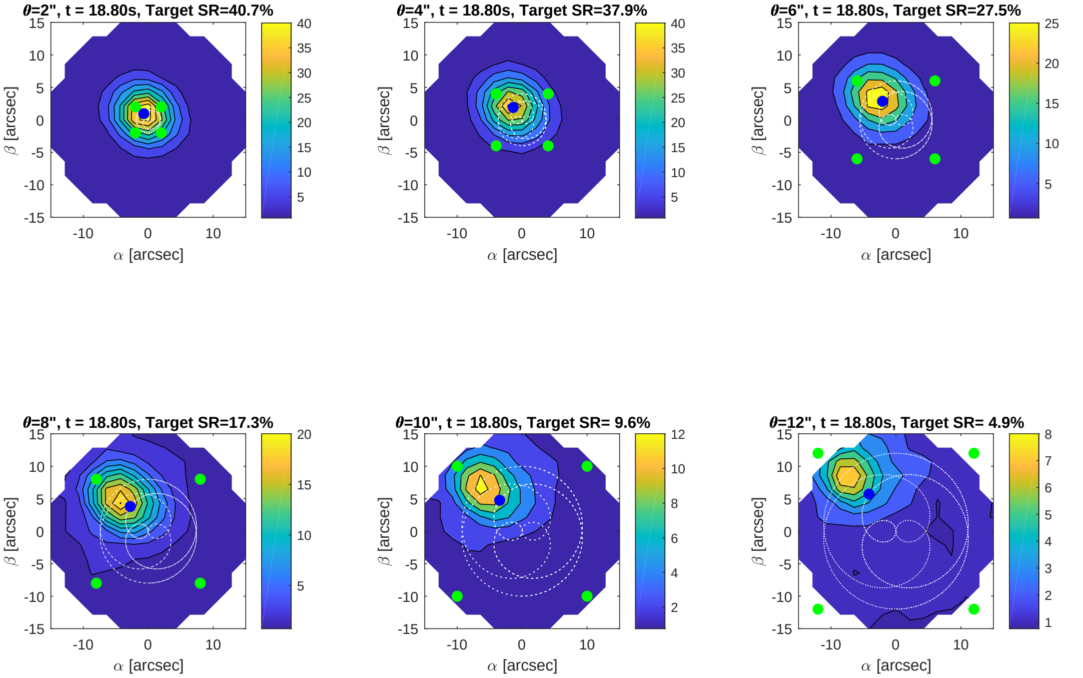

Figure 2. 4-vertex case, wide-field SR at sample $k = 9400$. Dotted line indicates the trajectory which has a period of 4000 samples (every 8 seconds). Green dots indicate vertices where control law is explicitly calculated.

3.2 N^2-Vertex Case

For the extended FoV case with more vertices, it is possible to achieve higher performance across the entire FoV. The resolution of the square grid can be increased without incurring any notable increase in online computational burden, however, the amount of memory that needs to be accessed in real time[†] grows approximately linearly with the number of vertices.

Figure 3 shows the averaged on-target performance over 10000 samples, and Figure 4 shows the instantaneous performance at a given sample during the sequence. The object moves smoothly over the entire FoV (as opposed to the reduced FoV in the 4-Vertex case) according to the following function:

$$\begin{aligned}\alpha &= 15\cos(\theta/4)\cos(\theta)\\ \beta &= 15\cos(\theta/4)\sin(\theta)\\ \theta &= 2\pi k/1000\end{aligned}$$

It can be seen in Figures 3 and 4 that the achieved peak SR values increase with the number of vertices, N^2, utilised in the convex polytopic parameterisation (equivalently, with the number of sub-regions that the FoV has been subdivided into). Also, notice in Figure 4, that for $N = 8$ and 16 the location of the peak SR coincides with the desired target direction.

[†] If the trajectory is known in advance, then the relevant control matrices can be scheduled to reduce the amount of memory required.

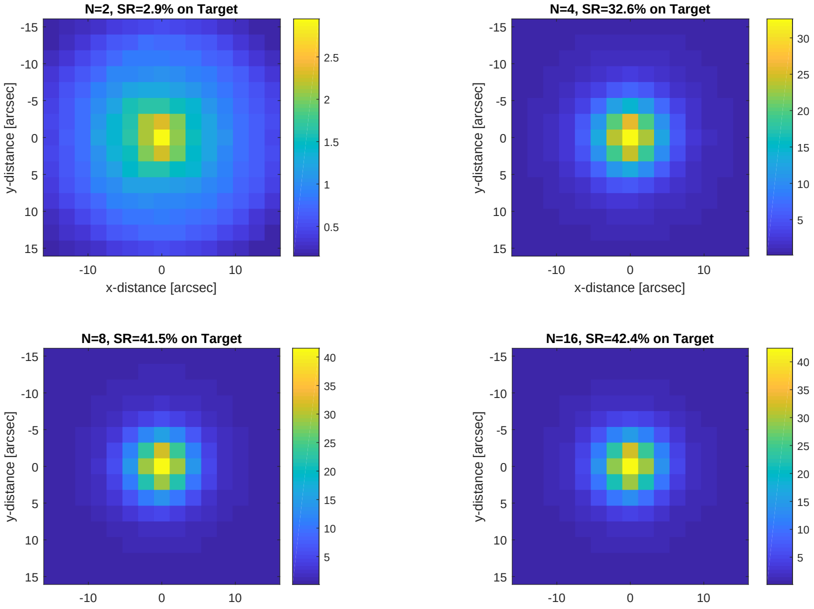

Figure 3. N^2-vertex case, on-target SR averaged over 10000 samples for target within region of width 15 arcsec.

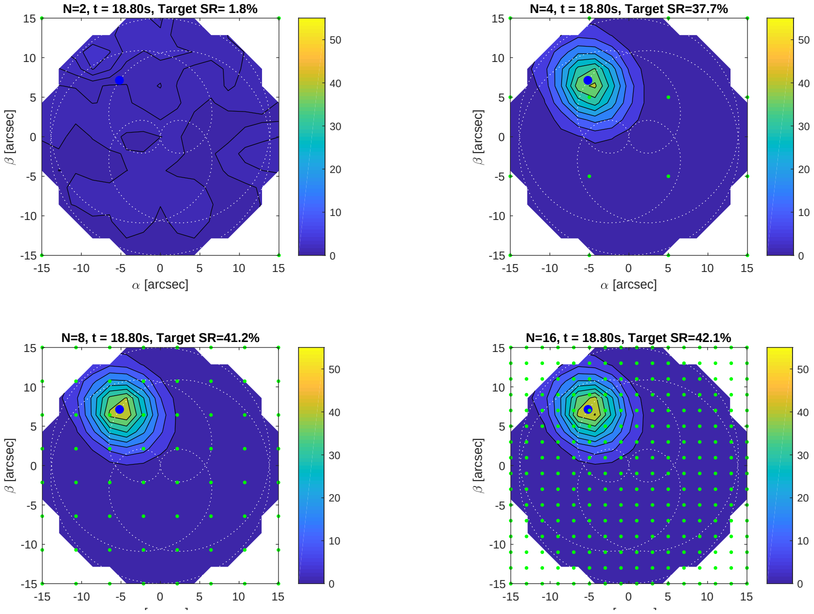

Figure 4. N^2-vertex case, wide-field SR at sample $k = 9400$. Dotted line indicates the trajectory which has a period of 4000 samples (every 8 seconds). Green dots indicate vertices where control law is explicitly calculated.

4. CONCLUSION

In this work we have introduced a Linear Parameter Varying (LPV) control scheme for AO and presented simulation results for the LTAO case, where the target object is moving continually throughout the FoV. Simulations show promising results for the proposed 4-vertex approach when the required FoV is small and the N^2-vertex case for LTAO over the full field of view.

References

[1] P. Piatrou and M. Roggermann. “Performance study of Kalman filter controller for multiconjugate adaptive optics.” In: *Applied Optics, 46(9):1446-55.* (2007).

[2] D. T. Gavel and D. M. Wiberg. “Towards Strehl-optimizing adaptive optics controllers”. In: *Proc. SPIE 4839, 972-982* (2002).

[3] Assemat. F, Wilson. R, and Gendron. E. “Method for simulating infinitely long and non stationary phase screens with optimized memory storage”. In: *Opt. Express 14, 988-999* (2007).

[4] Karl Åström. *Introduction to Stochastic Control Theory.* ACADEMIC PRESS New York and London, 1970.

[5] Jeff Shamma. “Analysis and Design of Gain Scheduled Control Systems”. PhD thesis. Department of Mechanical Engineering, Massachusetts Institute of Technology, 1988.

[6] Jeff Shamma. “An overview of LPV systems”. In: *Control of Linear Parameter Varying Systems with Applications.* Springer-Verlag, New York, 2012.

[7] F. Bruzelius. “Linear Parameter Varying Systems”. PhD thesis. Department of Signal and Systems, Chalmers University of Technology, Goteborg Sweden, 2004.

[8] Ryan McCloy, José De Doná, and María Seron. “Set theoretic approach to fault-tolerant control of linear parameter-varying systems with sensor reintegration”. In: *International Journal of Control* (2017), pp. 1–17.

[9] Rigaut. F. *Yao.* 2018. URL: http://frigaut.github.io/yao/.