

Distributed control in adaptive optics Deformable mirror and turbulence modeling

Rogier Ellenbroek^a, Michel Verhaegen^a, Niek Doelman^b, Roger Hamelinck^c, Nick Rosielle^c and Maarten Steinbuch^c

^aDelft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands;

^bTNO Science and Industry, Stieltjesweg 1, 2628 CK Delft, The Netherlands;

^cTechnische Universiteit Eindhoven, Den Dolech 2, 5600 MB, The Netherlands

ABSTRACT

Future large optical telescopes require adaptive optics (AO) systems whose deformable mirrors (DM) have ever more degrees of freedom. This paper describes advances that are made in a project aimed to design a new AO system that is extendible to meet tomorrow's specifications. Advances on the mechanical design are reported in a companion paper [6272-75], whereas this paper discusses the controller design aspects.

The numerical complexity of controller designs often used for AO scales with the fourth power in the diameter of the telescope's primary mirror. For future large telescopes this will undoubtedly become a critical aspect. This paper demonstrates the feasibility of solving this issue with a distributed controller design. A distributed framework will be introduced in which each actuator has a separate processor that can communicate with a few direct neighbors. First, the DM will be modeled and shown to be compatible with the framework. Then, adaptive turbulence models that fit the framework will be shown to adequately capture the spatio-temporal behavior of the atmospheric disturbance, constituting a first step towards a distributed optimal control. Finally, the wavefront reconstruction step is fitted into the distributed framework such that the computational complexity for each processor increases only linearly with the telescope diameter.

1. INTRODUCTION

Over the years, adaptive optics (AO) has grown from wild ideas to a proven technology that is indispensable for any large telescope. Inspired by its success, ideas have sprouted for larger and larger telescopes to look at even more distant and fainter stellar objects. However, the performance of a large telescope depends greatly upon the number of degrees of freedom in the deformable mirror (DM) of the AO system. This number should grow with the area of the primary mirror and thus quadratic in the telescope diameter.

Standard control approaches for AO consist of operations that typically scale with the number of actuators squared, which means that the total computational complexity of the controller increases with the telescope diameter to the fourth power. However, as the temporal controller bandwidth should be related to the number of actuators per area, which is more or less constant, this may not be decreased for larger telescopes. More efficient implementations will therefore be required for the control algorithms themselves. Many results towards realizing this are already available,^{1,2} but few methods implemented for AO fully exploit all available wavefront information. Moreover, the computational load of a controller that does attempt this³ also scales with the telescope diameter to the fourth power and no efficient implementations for such a controller exists yet.

This paper reports on work that has been done in a joint project aimed at designing a new AO system that has an extendible design, which means that the same design should be applicable when the number of actuators is increased. Advances in the design of the DM are discussed in,⁴ while this paper focusses on the controller design, for which a fully distributed control approach will be introduced. The goal of this controller is to yield optimal performance in terms of rms wavefront error while distributing calculations over a geometric grid of locally connected processors that each control one actuator. Feasibility of such a design depends on the available locality in the AO system itself, which will be investigated. Moreover, approaches for solving several arising problems will be discussed.

Further author information: Send correspondence to R. Ellenbroek. E-mail: r.m.l.ellenbroek@dcsc.tudelft.nl, Telephone: +31 (0)152783371

Controller design for AO systems is not a trivial task. Classically, controller design consists of an integrator structure⁵ in which the measured gradient vector \vec{S}_k from the wavefront sensor (WFS) at time instant k is first reconstructed to a wavefront error vector \vec{e}_k . This reconstruction step can be written as a matrix product of the reconstruction matrix $G^\#$, which is the pseudo-inverse of the geometry matrix and the slope vector \vec{S}_k (section 5):

$$\vec{e}_k = G^\# \vec{S}_k. \quad (1)$$

The resulting wavefront is then mapped back into an update vector \vec{d}_k for the actuator commands \vec{u}_k as

$$\vec{d}_k = B^{-1} \vec{e}_k, \quad (2)$$

where B is the mirror influence function matrix,⁵ which contains the influence of a unit actuator command for each actuator to wavefront phase at all actuator locations in its columns. The update \vec{d}_k is subsequently used to update the command vector \vec{u}_k as

$$\vec{u}_{k+1} = (1 - \beta) \vec{u}_k + \alpha \vec{d}_k \quad (3)$$

where α is the integrator gain and β the integrator leak factor that are tuned such that the closed loop is stable and the mean square wavefront error $\vec{e}_k^T \vec{e}_k$ is minimized.

This strategy only performs well if neither the DM, nor the wavefront sensor shows any detrimental dynamics or delays. In practice this is never the case, as there is always the delay of the wavefront sensor that integrates photons over time and the calculation time needed by a computer to calculate the command vector. By using the classical integrator, one in fact assumes that the atmospheric disturbance can be well predicted by the last measurement sample.⁶ The faster the behavior of the atmospheric disturbance w.r.t. the sampling frequency, the less this holds.

Therefore, many different control schemes have been devised that try to tackle these shortcomings.⁶⁻⁸ Often, the atmospheric disturbance is assumed to show a frozen flow characteristic, which makes it well predictable.⁸ On the other hand, according to some this is only due to the low-pass filtering effect of the Shack-Hartmann wavefront sensor⁹ (SHWFS). However, prediction models have scarcely been incorporated into a controller design. Some problems are that these models are based on strong assumptions, such as Kolmogorov statistics and Brownian motion,⁹ completely frozen flow¹⁰ or need external measurements such as wind speed and direction.¹¹ On the other hand, methods do exist that adapt their wavefront reconstruction step to measured wavefront statistics¹² in closed loop.

In a recent paper,³ an optimal controller design is demonstrated which comprehends a turbulence model that is based on open loop wavefront sensor data and doesn't make any presumptions on its behavior. However, the resulting controller matrices are fully dense and do not seem to possess structure that can be exploited for distributed implementations. In order to overcome this, an attempt is made in this paper to fit each of the controller's components into a distributed control framework: the mapping of a desired DM shape into a set of DM actuator command signals, the wavefront prediction and its reconstruction.

This distributed control framework will first be introduced in section 2, after which each of the components will be discussed. In section 3 the DM will be modeled both statically and dynamically. In section 4 a distributed turbulence model will be evaluated and finally in section 5 the distributed wavefront reconstruction problem will be examined.

2. DISTRIBUTED CONTROL DESIGN FRAMEWORK

In this section, the concept of distributed control is formulated in more detail. The distributed controller framework consists of a grid of processors called Distributed Processing Units (DPU) that each control one actuator of the deformable mirror. These DPU's have thus the same regular geometric grid layout as the actuators. Here it is assumed that this is a hexagonal grid, but all of the following can easily be transformed into a square grid layout. It is further assumed that the SHWFS measurement grid has the same structure as the actuator grid and is aligned with this such that each spot lies in the geometric center of four neighboring actuators (figure 1). This corresponds to the often used Fried geometry, with actuators on the phase point locations. Wavefront slope information extracted from each spot is known only to the four surrounding DPU's.

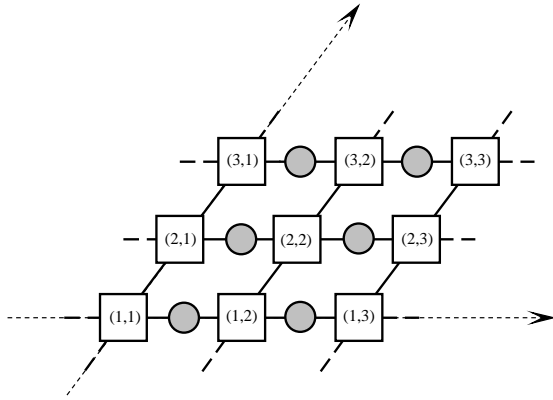


Figure 1. The distributed framework in a hexagonal layout. DPU's are represented by rectangles containing their geometric coordinates. They can communicate via the solid communication lines. The grey filled circles represent the SHWFS measurement spots.

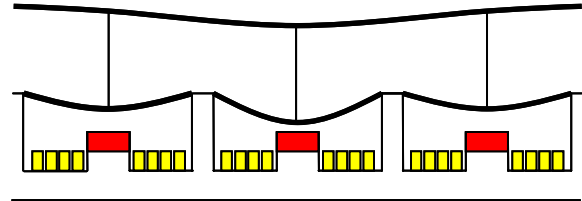


Figure 2. General design of the deformable mirror as described in detail in.⁴ The layered design consists of a deformable reflective surface, connection rods to the actuators in the actuator plate and a stiff support structure from top to bottom respectively.

Further, the controllers have the ability to communicate with four directly surrounding neighbors. The controller located at coordinate (i, j) can communicate to $(i - 1, j - 1), (i - 1, j + 1), (i + 1, j + 1)$ and $(i + 1, j - 1)$. If communication is required to other, more remote DPU's, this requires multiple sequential steps that take time and should be avoided as much as possible.

Locality. In this paper, the system property *locality* is used. Let this be defined as the relative accuracy with which it is possible to describe the behavior of the system using only local information when compared to using all information. Local information is information known to either a single DPU or to its direct neighbors. Although the number of direct neighbors may be chosen arbitrarily by defining a maximum geometric inter-actuator distance, it does *not* scale with the system dimensions. This links locality directly to *scalability*: if a system has a high locality then its behavior can be well described using a limited set of local information that does not grow with the dimensions of the system.

Calculations and notation. In the sequel, when computations are said to fit within the distributed framework, this means that they can be performed either directly by a local DPU or by first receiving information from neighbors to which it has a direct communication link, i.e. its *connections*. Let a vector of which each element is only known to a single corresponding DPU be denoted by \tilde{v} . For often used matrix-vector products $\tilde{w} = M\tilde{v}$, this implies that M must have a specific sparse structure: the i^{th} row of M may only contain non-zero elements at positions that correspond to the connections of DPU i .

However, when a computation requires not only information from the DPU's connections, but also from a limited set (i.e. whose size is independent of the system dimensions) of other neighboring DPU's, this can also fit into the distributed framework. This just yields a fixed number of sequential communication steps.

3. MIRROR MODELING

The DM for which the distributed controller is designed,⁴ is of the continuous face-sheet type. This section comprehends the DM modeling that is relevant for controller design and will be used to answer the following questions:

1. What is the locality of the mirror's influence functions?
2. Will the dynamics of the DM be relevant for the controller design?
3. How much damping will be required in the actuators?

The general mirror design is depicted in figure 2 and consists of a thin deformable reflective face-sheet that is supported by electromagnetic push-pull actuators. They are connected to the reflective face-sheet with thin rods,

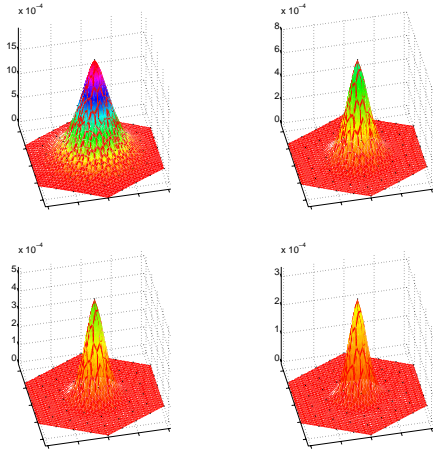


Figure 3. Influence functions in [m/N] for the central actuator in a mirror with a hexagonal actuator grid with 61 actuators with a stiffness of 100, 500, 1000 and 2000 [N/m] from left to right and top to bottom respectively.

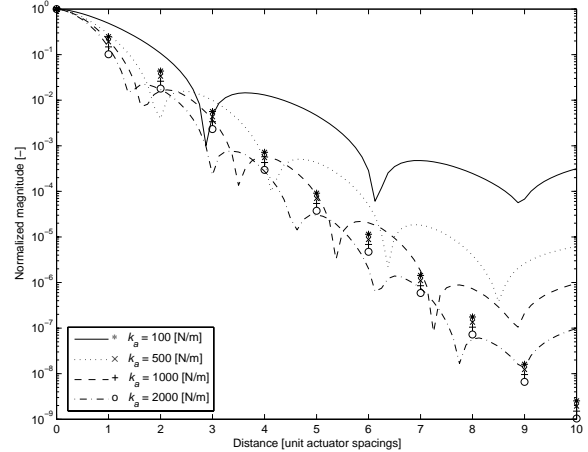


Figure 4. Lines: normalized influence function maximum magnitude versus actuator spacing unit distances. Marks: normalized magnitude of the corresponding inverse matrix.

thus fixing only the out of plane deformation and are attached to a stiff support structure that gives the mirror a flat reference.

The following subsections comprehend the modeling and identification of a single actuator which is subsequently used to create both static as well as dynamic models of the complete DM. The implications of the result for a distributed controller design will be evaluated.

3.1. Actuator modeling and identification

The actuators consist of a closed magnetic circuit in which a permanent magnet provides a static magnetic force on a ferromagnetic core which is suspended in a membrane. By applying a current through the coil which is situated around the magnet, this force is influenced, providing movement of the ferromagnetic core. This movement is transferred via a rod imposing the out-of plane displacements in the reflective deformable membrane. In the actuator design a match is made between the negative stiffness of the magnet and the positive stiffness of the membrane suspension. Although this is non-linear w.r.t. the membrane deflection, the eventual operating range of the actuator is small enough to justify a linear model.

The actuator can be straightforwardly modeled as a linear mass-spring-damper system. The moving mass consists of the actuator membrane and the rod, the spring is the net mechanical actuator stiffness and the damping can be either due to air flow inside the housing or due to electronics (e.g. back-EMF of the coil being dissipated in a resistor). The actuator has been built and a set of input-output data was obtained by measuring the membrane deflection while feeding the coil a white noise current signal. From this data, the following mass m_a , stiffness k_a and damping b_a values were estimated via frequency domain identification:

$$m_a = 9.9 \cdot 10^{-3} [g], \quad b_a = 1.4 \cdot 10^{-3} [Ns/m], \quad k_a = 330 [N/m]. \quad (4)$$

3.2. Modeling the deformable face-sheet

Although the face-sheet has only a slight thickness, it still has a considerable out-of-plane stiffness when and should be modeled as a thin plate.¹³ The thin rods connecting the actuators to the plate will be assumed to exert point-forces, which allows the use of the analytical solution to the biharmonic equation.¹⁴ For a circular plate of radius r_{plate} with clamped edge conditions (no deflection or out-of-plane rotation at the edge) the plate deformation can be calculated as:

$$h(F, z, \zeta) = \frac{F}{16\pi R} \left\{ (1 - r^2)(1 - \rho^2) + [r^2 + \rho^2 - 2r\rho \cos(\phi - \psi)] \ln \frac{r^2 + \rho^2 - 2r\rho \cos(\phi - \psi)}{1 + r^2\rho^2 - 2r\rho \cos(\phi - \psi)} \right\}, \quad (5)$$

where $h(F, z, \zeta)$ expresses the plate deflection at position z due to a perpendicular point-force F at position ζ . Positions z and ζ are defined in the complex plane as $z = r e^{j\phi}$ and $\zeta = \rho e^{j\psi}$ respectively, where r and ρ are normalized w.r.t r_{plate} . Further, R is the flexural rigidity of the plate, which is defined as:

$$R = \frac{Et^3}{12(1 - \mu^2)r_{plate}^2}, \quad (6)$$

where E is the Young's modulus of the plate material, μ its Poisson ratio and t its thickness. The equation is linear in the force F and because of the small deflections ($O(\mu m)$) it is sufficiently accurate to use superposition of deflections due to individual forces in case of multiple simultaneous point-forces.

Let the deflection of the plate be given as a vector \vec{h} , defined on a limited set of discrete points \mathcal{D} . This set comprehends all points at which a force acts on the plate. These forces are both the supporting actuator forces as well as lumped inertia forces of the thin plate itself, that are spread over a fine grid. The forces are contained in the vector \vec{F} , which can be related to the deflection \vec{h} via a stiffness matrix K with elements $K_{i,j}$:

$$K\vec{F} = \vec{h}, \quad K_{i,j} = \frac{\partial h(F, z_i, \zeta_j)}{\partial F}, \quad (7)$$

where both i and j enumerate all points in \mathcal{D} . A force equilibrium is now sought that satisfies

$$(\overline{M}_p + \overline{M}_a)\ddot{\vec{h}} + (\overline{B}_p + \overline{B}_a)\dot{\vec{h}} + \vec{F} + \overline{C}_a\vec{h} = \vec{F}_a \quad (8)$$

where $\overline{C}_a, \overline{M}_p, \overline{M}_a, \overline{B}_p$ and \overline{B}_a are all diagonal matrices containing the actuator stiffness, plate inertia, actuator inertia, plate air-damping and actuator damping terms respectively. \vec{F}_a is a vector containing the actuator forces. Using (7), a standard dynamical system form is obtained with mass, damping and stiffness matrices $\overline{M}_a + \overline{M}_p$, $\overline{B}_a + \overline{B}_p$ and $K^{-1} + \overline{C}_a$ respectively. Note that if a certain grid point does not have an external force, mass, stiffness or damping, the entry in the corresponding matrix or vector is zero.

The static case: influence functions. For the static case, all time-derivative terms in (8) are zero and the deflection \vec{h} can be directly expressed in terms of the forces \vec{F}_a , from which the mirror influence matrix B containing the DM influence functions in its columns can be derived:

$$B = (K^{-1} + \overline{C}_a)^{-1}. \quad (9)$$

The influence function of a mirror's central actuator has been plotted for four values of k_a in figure 3. It can be observed that this stiffness significantly affects the width of the influence function. This can be seen even more clearly from figure 4, in which the lines represent the normalized maximum absolute DM deflection on a circle with certain radius around the poked actuator. For $k_a = 1000$ [N/m], that will be aimed for in the mechanical design, the deflection of the 4th neighboring actuator already remains below one thousandth of the central deflection. For an actuator range in the order of [μm], this is a deformation in the order of [nm], which can be disregarded without affecting the optical quality.

However, calculation of the actuator commands corresponding to a desired DM shape involves the inverse of the influence matrix B . The question relevant for distributed controller design is thus whether the locality is retained in the inverse mapping $B^{-1}\tilde{\phi}$. This can be observed to be the case from the markers in figure 4, which show the normalized magnitude of the elements of the i^{th} row of B^{-1} that correspond to phase points at a certain distance – in unit actuator spacings – from actuator i . This magnitude decreases also very fast with the distance and is only slightly influenced by k_a . Moreover, the same decrease over the distance holds for larger grids, which means that the number of neighbors needed to calculate the actuator commands stays constant for increasing system dimensions, confirming that also the inverse mapping has a high locality.

The dynamic case: eigenmodes and damping. A system property that is also very important for controller design is the location of its eigenfrequencies. As for any system, the DM will be much easier to control below its first eigenfrequency, which is even more true for this system because many eigenfrequencies will be clustered together in a small range. The eigenfrequencies can be straightforwardly calculated using the dynamical model and are significantly affected by k_a , as shown for the lowest eigenfrequency by the thin line in figure 5. For

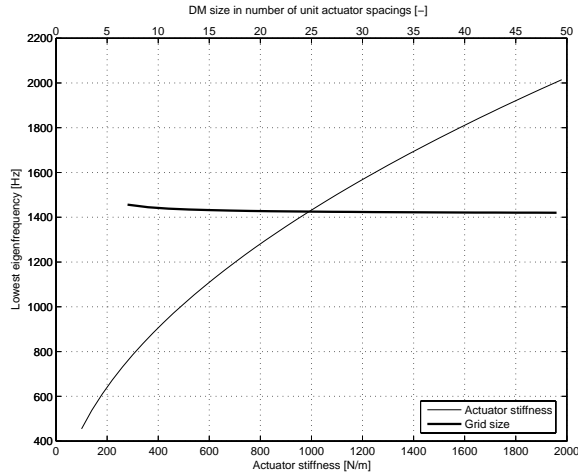


Figure 5. Influence of both the actuator stiffness as well as the DM size on its lowest eigenfrequency. The first corresponds to the bottom axis, while the second to the top axis.

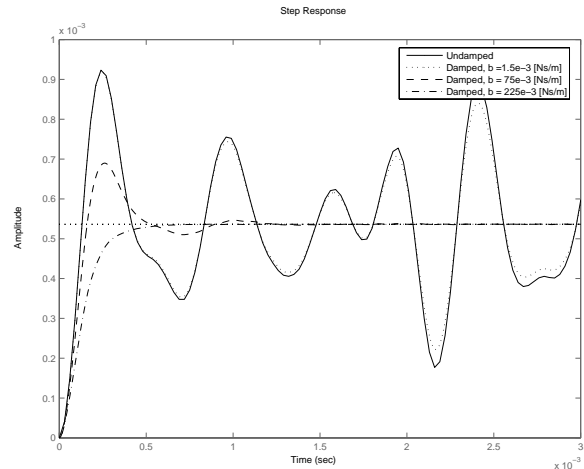


Figure 6. Impulse responses of a mirror with 61 actuators in a hexagonal grid with various actuator damping coefficients.

$k_a = 1000$ [N/m] this eigenfrequency lies around 1400 [Hz], which should be enough to achieve the desired temporal control bandwidth of approximately 200 [Hz].

Further, since both plate mass as well as actuator stiffness are added, the lowest eigenfrequency should not decrease much when the DM design is extended to larger sizes. The value of the lowest eigenfrequency is plotted against the DM size in number of unit actuator spacings for $k_a = 1000$ [N/m] as the thick line in figure 5. As can be seen, the frequency decreases slightly, but this is only the decaying effect of the clamped plate boundary. It can be safely concluded that the eigenfrequencies of the DM will not affect the control system performance as long as they are damped sufficiently.

Since the DM is to be controlled over a slow SHWFS for feedback, high-frequent oscillations can never be suppressed. Therefore, the system should have a suitable amount of electromagnetic or air damping to have a fast, but well damped step-response. The step response of a single actuator in a hexagonal 61 actuator DM has been plotted in figure 6 for the undamped and for several damped cases. The undamped system has a highly oscillating step response, requiring over 100 times more actuator damping than that estimated in section 3.1 to suppress. On the other hand, with this amount of damping, the rise and settling times are very good: approximately 0.15 [ms] and 0.5 [ms] respectively. Dynamics of the DM will then not pose any trouble for a distributed control framework.

4. TURBULENCE MODELING

In this section, the locality of the atmospheric disturbance will be studied by means of a local turbulence model. Characterization and modeling of atmospheric turbulence has a long history, e.g. Fried¹⁵ in 1965. Many approaches towards realistic simulation models have since then been recorded. Models based on Kolmogorov statistics such as^{16,17} work well, but apart from pure frozen flow behavior it appears difficult to model the evolution of the disturbance over time. Models exploiting the fractal nature of turbulence¹¹ seem better at this, and even speculate on the existence of good linear predictor models.

For controller design it is common to model only the static or slowly time-varying statistical properties of the disturbance¹² in the sense of a covariance matrix. Recently, results of an optimal \mathcal{H}_2 controller for AO were shown⁶ to yield a much improved performance. This design included a turbulence model based only on open loop SHWFS measurement data. Alas, this turbulence model is not easily fitted into the distributed framework as the obtained matrices of the turbulence model are dense and seem to have little structure. On the other hand, continuing on results by^{11,18} the remainder of this section discusses a moving average predictor filter that is designed to fit within the distributed framework and can in the same framework be made adaptive.

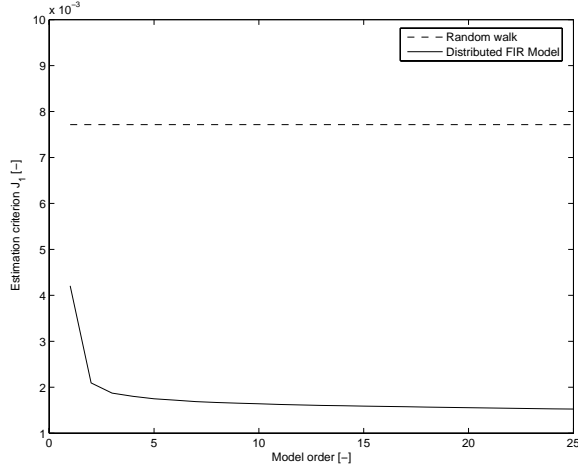


Figure 7. Prediction criterion J_1 evaluated for the random walk predictor and distributed FIR predictors of orders 1, 5 and 25.

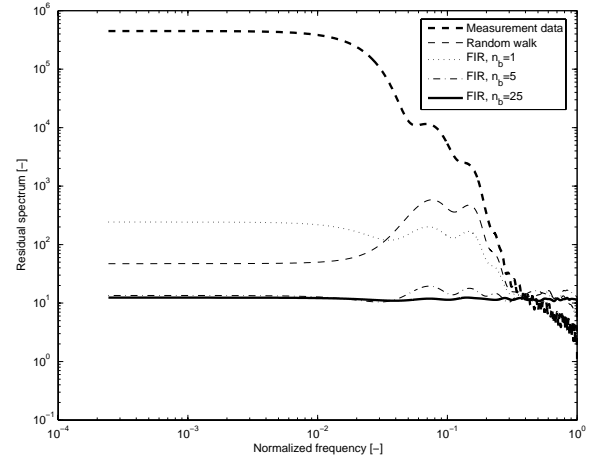


Figure 8. Average of the per-phase point temporal frequency spectra of \tilde{d}_k evaluated for the random walk predictor and distributed FIR predictors of orders 1, 5 and 25.

4.1. A distributed moving average filter

Let a prediction of the wavefront at time $k + 1$ be given by an n_b^{th} order FIR filter

$$\hat{\phi}_{k+1} = B_1 \tilde{\phi}_k + B_2 \tilde{\phi}_{k-1} + \dots + B_{n_b} \tilde{\phi}_{k-n_b+1}, \quad (10)$$

where matrices B_i with $i = 1, 2 \dots n_b$ are the filter coefficient matrices and $\tilde{\phi}_j$ previous wavefront measurements. Further, all matrices B_i are equally parameterized such that the products $B_i \tilde{\phi}_j$ fit within the distributed framework (section 2). This can be rewritten to a more standard form where the unknown filter parameters are contained in one vector $\tilde{\theta}$ via the following steps. First note that a sparse matrix B_i can be written as the series product

$$B_i = L_B \bar{B}_i R_B, \quad (11)$$

where L_B and R_B are selection matrices containing only ones and zeros and \bar{B}_i is a diagonal matrix containing all possibly non-zero elements of B_i . Subsequently, use the Khatri-Rao matrix identity¹⁹ to rewrite

$$\text{vec}(L_B \bar{B}_i R_B \tilde{\phi}_j) = (\tilde{\phi}_j^T R_B^T \odot L_B) \tilde{b}_i = L_B \text{diag}(R_B \tilde{\phi}_j) \tilde{b}_i, \quad (12)$$

where $\tilde{b}_i = \text{vecd}(\bar{B}_i)$ and dvec is an operator that stacks all diagonal matrix elements into a vector.¹⁹ The filter equation thus becomes

$$\hat{\phi}_{k+1} = L_B \text{diag}(R_B \tilde{\phi}_k) \tilde{b}_1 + \dots + L_B \text{diag}(R_B \tilde{\phi}_{k-n_b+1}) \tilde{b}_{n_b} = \Phi_k \tilde{\theta}, \quad (13)$$

where

$$\Phi_k = [L_B \text{diag}(R_B \tilde{\phi}_k) \quad \dots \quad L_B \text{diag}(R_B \tilde{\phi}_{k-n_b+1})], \quad \tilde{\theta} = [\tilde{b}_1^T \quad \dots \quad \tilde{b}_{n_b}^T]^T. \quad (14)$$

Identification. Let the optimal value for $\tilde{\theta}$ be defined to minimize the expected squared prediction error $\epsilon = \langle \tilde{e}_k^T \tilde{e}_k \rangle$, where $\langle \cdot \rangle$ denotes the expected value operator and $\tilde{e}_k = \hat{\phi}_k - \tilde{\phi}_k$ the prediction error. The optimization problem thus becomes:

$$\tilde{\theta} = \arg \min_{\tilde{\theta}} \langle (\tilde{\theta}^T \Phi_k^T - \tilde{\phi}_k^T) (\Phi_k \tilde{\theta} - \tilde{\phi}_k) \rangle. \quad (15)$$

Let the expected value be approximated with a finite sum over n samples s.t.

$$\tilde{\theta} = \arg \min_{\tilde{\theta}} \sum_{k=1}^n (\tilde{\theta}^T \Phi_k^T - \tilde{\phi}_k^T) (\Phi_k \tilde{\theta} - \tilde{\phi}_k) = \arg \min_{\tilde{\theta}} \sum_{k=1}^n \tilde{\theta}^T \Phi_k^T \Phi_k \tilde{\theta} - 2 \tilde{\phi}_k^T \Phi_k \tilde{\theta} + \tilde{\phi}_k^T \tilde{\phi}_k, \quad (16)$$

which yields a linear system of equations that can be straightforwardly solved for $\tilde{\theta}$:

$$\mathbf{R}\tilde{\theta} = \tilde{v}, \quad \text{where} \quad \mathbf{R} = \sum_{k=1}^n \Phi_k^T \Phi_k, \quad \tilde{v} = \sum_{k=1}^n \Phi_k^T \tilde{\phi}_k. \quad (17)$$

Calculation of the solution now involves the inverse of \mathbf{R} , which exists if the measured signal $\tilde{\phi}_k$ is sufficiently exciting. In practice this will be the case if the number of samples n is chosen large enough.

This FIR filtering approach has been applied to measurements from an AO breadboard which contains a plan parallel circular plate turbulence simulator resembling a spatial Kolmogorov distribution with a $D/r_0 = 5$. In open loop, 10000 samples were collected using a 127-spot hexagonal SHWFS. The filter order was varied over the range $n_b = 1 \dots 25$ and the filter coefficient matrices B_i were parameterized such that the product $B_i \tilde{\phi}_j$ can be evaluated using information from only the six phase points surrounding phase point (x, y) , indexed as $(x-1, y)$, $(x-1, y+1)$, $(x, y+1)$, $(x+1, y)$, $(x+1, y-1)$, $(x, y-1)$. The first half of the data-set was used for identification, the second for validation of the identified model.

Validation. Results will be analyzed using the criteria J_1 , J_2 and S_1 :

$$J_1 = \frac{\sum_{k=1}^n \tilde{e}_k^T \tilde{e}_k}{\sum_{k=1}^n \tilde{\phi}_k^T \tilde{\phi}_k}, \quad J_2 = n \frac{\tilde{e}_k^T \tilde{e}_k}{\sum_{k=1}^n \tilde{\phi}_k^T \tilde{\phi}_k}, \quad S_1(f) = \frac{1}{n_p} \sum_{p=1}^{n_p} \mathcal{E}_p(f), \quad (18)$$

where n_p is the number of phase points and \mathcal{E}_p is the temporal spectrum of the value of the phase point with index p at frequency f . Results for J_1 are shown against the filter order in figure 7, where the value of this criterion is also shown for the *random walk* predictor, for which $n_b = 1$ and $B_1 = I$. Several observations can be made from this figure. Firstly, note that the local FIR predictor can better the random walk predictor more than a factor 4. Further, observe that J_1 is only slightly improved by increasing the filter order above 5. This is also clear in figure 8, which shows that the average temporal power-spectrum $S_1(f)$ obtained using a fifth order local FIR filter is already almost white. As the FIR filter is both local in time and in space, this suggests that the atmospheric disturbance has a high locality. However, calculation of the static predictor filter requires inversion of a full matrix, which is both computationally intensive and can only be done after a large set of measurement data has been collected. Therefore, the next subsection discusses possibilities for making this filter adaptive.

4.2. Adaptive distributed predictor

Adaptive linear filters have been around for a long time and their properties thoroughly studied.²⁰ The filter coefficients of an adaptive filter are not fixed, but are updated each sample to either converge to their globally optimal values or to track changes in the statistical properties of the input signal. This section discusses two distributed adaptive filters that are of the first kind: steepest descent and LMS. Note that all calculations of the filter update equations can be performed within the distributed framework.

Steepest descent. As will be shown in more detail in the next section, the method of steepest descent actually constitutes a method for solving systems of linear equations.²¹ For the case of adaptive filtering, it is applied to the system of equations in (17). For each sample, the filter coefficient vector is updated as²⁰

$$\tilde{\theta}^{(k+1)} = (I - \alpha \mathbf{R}_k) \tilde{\theta}^{(k)} + \alpha \tilde{v}_k \quad (19)$$

where

$$\mathbf{R}_k = \mathbf{R}_{k-1} + \Phi_k^T \Phi_k, \quad \tilde{v}_k = \tilde{v}_{k-1} + \Phi_k^T \tilde{\phi}_k. \quad (20)$$

As both \mathbf{R}_k and \tilde{v}_k converge to the best estimates for the covariance matrix \mathbf{R} and correlation vector \tilde{v} respectively for $k \rightarrow \infty$, the adaptive filter should converge to the static solution as long as the step size α is chosen properly.²⁰

LMS. The LMS or Least Mean Squares algorithm is also a popular adaptive algorithm that knows many applications because of its low computational complexity. It has the same update law as the method of steepest descent, but here the estimates of \mathbf{R} and \tilde{v} are simply taken as the most recent set of measurement values:

$$\mathbf{R}_k = \Phi_k^T \Phi_k, \quad \tilde{v}_k = \Phi_k^T \tilde{\phi}_k. \quad (21)$$

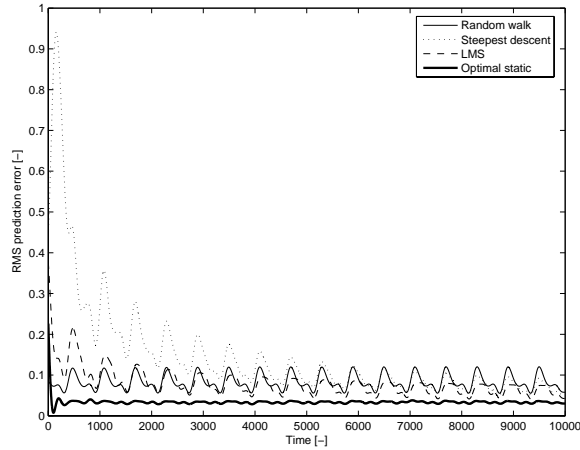


Figure 9. Performance criterion J_2 from (18) of the two adaptive filters, together with the results obtained using the random walk predictor and the optimal static predictor. All filters are of 15th order and results have been smoothed to make them more clear.

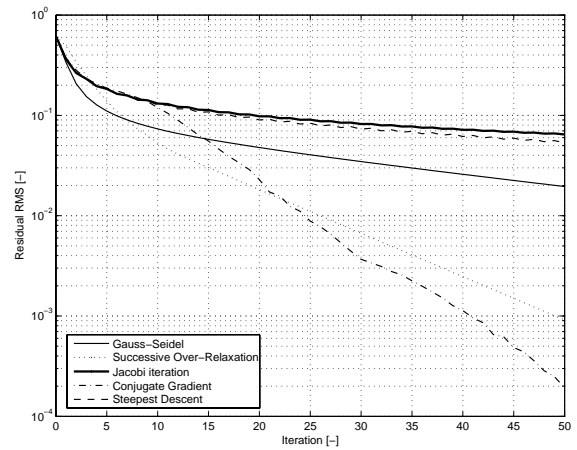


Figure 10. RMS values of the residuals obtained after each iteration of various iterative reconstructors. Results are averaged over 20 wavefronts artificially generated using the method described in²² over a hexagonal phase grid with 17 points on its diagonal.

Results for both algorithms are shown in terms of J_2 in figure 9 together with the corresponding results from the optimal static predictor and the random walk predictor. Although the LMS filter shows a lot of variance and the steepest descent algorithm converges slowly, both adaptive algorithms eventually better the performance of the random walk predictor. Further improvement of the adaptive filtering performance is a subject for further research.

5. WAVEFRONT RECONSTRUCTION

The last, but probably most important component of the controller is the wavefront reconstructor that reconstructs the wavefront phase from the gradients measured by a SHWFS. Let the phase vector be denoted $\vec{\phi}$ and the measured slope vector $\vec{S} = [\vec{S}_x^T \ \vec{S}_y^T]^T$ which consists of the gradients \vec{S}_x and \vec{S}_y corresponding to the x and y directions respectively. The relation between \vec{S} and $\vec{\phi}$ can be expressed using a matrix G as

$$\vec{S} = G\vec{\phi}. \quad (22)$$

The geometry matrix G can be chosen in numerous ways, but here the Fried geometry⁵ will be used that can be easily adapted to a hexagonal grid.³ Thus, the wavefront reconstruction consists of solving a set of linear equations that arise from the least squares fitting problem of $\vec{\phi}$:

$$G^T G \vec{\phi} = G^T \vec{S}. \quad (23)$$

Note here that the matrix $G^T G$ is singular due to the two unobservable modes of the SHWFS,⁵ which means that a pseudo-inverse²¹ must be employed to solve the system directly. However, calculation of the solution for $\vec{\phi}$ would require $O(n_p^2)$ calculations even when the matrix $(G^T G)^{-1} G^T$ is known, because this matrix is dense, whereas G is sparse. This also requires a lot of memory when large scale AO systems are considered and has led to iterative¹ and hierarchic² approaches that are not only computationally more efficient ($O(n_p)$), but also lack the need to store the dense matrix.

However, although these methods yield efficient solutions that may even be partially parallelized, they do not fit within the distributed framework. Thus, two options can now be considered: do the reconstruction by a dedicated external processor or find a reconstruction procedure that does fit into the framework. Because of the structure of the matrices G and $G^T G$, the second option seems interesting to explore using well known iterative

solver methods such as Jacobi iteration and successive overrelaxation.²¹ Let the reconstruction problem of (23) be written more succinctly as

$$A\vec{\phi} = \vec{b}. \quad (24)$$

Iterative solvers in general use the notion that the solution to this problem is also the minimum of the quadratic function

$$f(\vec{\phi}) = \frac{1}{2}\vec{\phi}^T A\vec{\phi} + b^T\vec{\phi} + c, \quad (25)$$

which can – in the case that this is a convex function – be found by gradient searches. Two basic iterative methods that work in a very straightforward way are the method of steepest descent and Jacobi iteration, which have the following update law:

$$\vec{\phi}^{(k+1)} = (I - \overline{Q}A)\vec{\phi}^{(k)} + \overline{Q}\vec{b}, \quad (26)$$

where \overline{Q} is a diagonal matrix that for Jacobi iteration equals the inverse of the diagonal of A and for steepest descent $\overline{Q} = \alpha I$ with α the step size. Details of these procedures are beyond the scope of this paper, but note that in both cases the sparse structure of A and thus $G^T G$ is left intact. Due to the definition of the Fried geometry, the structure of $G^T G$ is such that the product $\tilde{w} = (G^T G)\tilde{v}$ is always a two-step procedure within the distributed framework. Furthermore, calculation of $\tilde{b} = G^T \tilde{S}$ also fits the framework: calculation of each element of \tilde{b} requires information from only the four measurement spots surrounding the corresponding phase point.

On the other hand, the convergence of both algorithms is very slow. This can be observed in figure 10, where the convergence in terms of residual rms of both methods has been plotted together with that of Gauss-Seidel, successive overrelaxation and conjugate gradient.²¹ Although the conjugate gradient method has the fastest convergence, it does not fit within the distributed framework, which proves to be not entirely the case for the second-fastest algorithm: successive over-relaxation (SOR). SOR is a variant on the Gauss-Seidel algorithm that converges fast for the reconstruction problem and can be partly parallelized.²³ Its update rule can be expressed as

$$(I + \omega\overline{D}^{-1}L)\vec{\phi}^{(k+1)} = ((1 - \omega)I - \omega\overline{D}^{-1}U)\vec{\phi}^{(k)} + \omega\overline{D}^{-1}\vec{b}, \quad (27)$$

where the matrix A has been split into three parts s.t. $A = L + \overline{D} + U$ with \overline{D} its diagonal and L and U its lower and upper triangular parts respectively. The scalar over-relaxation parameter ω can be tuned to optimize the convergence speed,²¹ which is beyond the scope of this paper. The update can be observed to involve two sequential steps: calculation of the right-hand side and solving for $\vec{\phi}$:

$$\vec{x}_k = ((1 - \omega)I - \omega\overline{D}^{-1}U)\vec{\phi}^{(k)} + \omega\overline{D}^{-1}\vec{b} \quad (28)$$

$$(I + \omega\overline{D}^{-1}L)\vec{\phi}^{(k+1)} = \vec{x}_k \quad (29)$$

Due to the structure of A , the first step fits exactly within the distributed framework, whereas the second step constitutes a forward substitution problem for which this does not hold. But again due to the structure of A when the points in the phase grid are suitably ordered, this forward substitution problem can be written as a series of $n_f \approx n_w/2$ elimination steps – where n_w is the number of phase points over the diagonal of the grid. When the vectors $\vec{\phi}^{(k+1)}$ and \vec{b} are stacked, the update law can be written as

$$\begin{bmatrix} \vec{\phi}^{(k+1)} \\ \vec{b} \end{bmatrix} = \begin{bmatrix} M_{n_f} & 0 \\ 0 & I \end{bmatrix} \cdots \begin{bmatrix} M_1 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} F & \omega\overline{D}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} \vec{\phi}^{(k)} \\ \vec{b} \end{bmatrix}, \quad F = (1 - \omega)I - \omega\overline{D}^{-1}U. \quad (30)$$

where each elimination matrix M_k requires exactly two steps in the distributed framework. This means that one iteration of SOR can be performed with $O(n_w)$ sequential steps. As the number of SOR iterations required to obtain acceptable reconstruction accuracy scales also with $O(n_w)$, the total number of sequential steps required is of order $O(n_w^2)$. However, this procedure yields a lot of idle time for the DPU's which – as will be shown – can be significantly reduced by means of simple scheduling rules.

The scheduling rules can be best understood by regarding individual off-diagonal elements in the matrices as

operations that involve inter-DPU communication. These operations cannot be performed until the required information is available at the other DPU. When looking closely at the full sequence corresponding to n_w iterations of SOR, it shows that not all operations are performed as soon as they could be. By applying the following scheduling rules over and over again, diagonal matrices are formed at the end of the sequence that can be directly combined with any previous step and subsequently removed. An operation can be advanced from step i to $i - 1$ whenever

1. the value needed by DPU d from DPU s is not modified at step $i - 1$
2. and the old value at DPU d is not needed by any other DPU in step i .

The length of the remaining series so reduces to $O(n_w)$, making it linear in the telescope diameter. This is then also the case for the required processing speed of the DPU's and the inter-DPU communication bandwidth. This seems acceptable, as the reconstruction process is fundamentally a spatial integration procedure. When information from geometrically remote gradients is discarded, the proper phase value will not be obtained. Only after at least n_w sequential communication steps can gradient information have propagated completely from one side of the grid to the other.

On the other hand, it seems probably that the reconstruction procedure will require significantly fewer iterations when starting from a good initial guess supplied by a predictor filter (section 4). But as this will introduce the two unobservable modes into the solution – which are neither generated nor damped by the iterative reconstruction procedure – this remains a subject for further research.

6. CONCLUSIONS AND FUTURE WORK

It can be concluded that distributed control is a very interesting and feasible approach for controlling large scale AO systems. It has been shown that important components in the control loop can be very well described using only the information that is available locally. In section 3 this has been shown for the DM and the mapping of a desired wavefront shape into a set of DM actuator signals. Moreover, it was shown that it can be considered a static element on the condition that sufficient damping is present. Further, a distributed predictor filter was described in section 4. Application on measurement data showed the promising result that using only the local wavefront history a very good prediction can be obtained. Moreover, a simple modification to make the filter adaptive was shown to give respectable performance and to fit within the distributed framework.

Finally, in section 5 a distributed approach to the wavefront reconstruction problem was discussed. Although the complexity of this problem does scale with the dimensions of the system, it has been shown to grow linearly in the telescope diameter. For practical implementations this implies that the single DPU processing power as well as the local communication bandwidth have to increase linearly with the telescope diameter. It is a subject for further research to further attenuate this scaling.

Next steps will be the closing of the control loop in simulations, design of the driving electronics for the DM and experimental validation on a breadboard.

Acknowledgements

The DM is designed in a joint cooperation between the Delft University of Technology, TNO Science and Industry and the Technische Universiteit Eindhoven. The work done is supported by Senter Novem, the Dutch Innovative Research Project (IOP) Precision Technology.

REFERENCES

1. L. Gilles, "Order- n sparse minimum-variance open-loop reconstructor for extreme adaptive optics," *Optics Letters* **28**, pp. 1927–1929, October 2003.
2. D. G. MacMartin, "Local, hierarchic, and iterative reconstructors for adaptive optics," *Journal of the Optical Society of America* **20**, pp. 1084–1093, June 2003.
3. K. Hinnen, M. Verhaegen, and N. Doelman, "Adaptive optics h²-optimal control design applied on an experimental setup," in *Proceedings of the SPIE conference on astronomical telescopes and instrumentation - advances in adaptive optics*, SPIE, May 2006.

4. R.F.M.M.Hamelinck, N. Rosielle, and N. Doelman, "Large adaptive deformable mirror: design and first prototypes," in *Proceedings of the 5th annual SPIE conference on Optics and Photonics*, SPIE, (San Diego), August 2005.
5. J. Hardy, *Adaptive optics for Astronomical Telescopes*, Oxford University Press, New York, 1998.
6. K. Hinnen, M. Verhaegen, and N. Doelman, "H²-optimal control of an adaptive optics system: part ii, closed-loop controller design," in *Proceedings of SPIE*, M. L.-H. Robert K. Tyson, ed., **5903**, SPIE, August 2005.
7. A. A. Abdullah and P. A. Ioannou, "Decentralized reconfigurable control for large-scale systems with application to a segmented telescope test-bed," in *Proc. of the 42nd IEEE Conference on Decision and Control*, (Maui, Hawaii, USA), December 2003.
8. Y.-T. Liu and S. Gibson, "Adaptive optics with adaptive filtering and control," in *Proceedings of the American Control Conference (ACC)*, pp. 3176–3179, (Boston, Massachusetts, USA), June 30 – July 2 2004.
9. D. G. Pérez, L. Zunino, and M. Garavaglia, "Modeling turbulent wave-front phase as a fractional brownian motion - a new approach," *Journal of the optical society of America A* **21**, pp. 1962–1969, October 2004.
10. N. Imbert, P. Mouyon, and G. Monteseny, "State representation and simulation of 2d turbulent wind," in *AIAA Modeling and Simulation Technologies Conference*, (Denver, CO), August, 14-17 2000.
11. C. Schwartz, G. Baum, and E. N. Ribak, "Turbulence-degraded wave fronts as fractal surfaces," *Journal of the Optical Society of America A* **11**, pp. 444–, January 1994.
12. B. L. Ellerbroek and T. Rhoadarmer, "Real time adaptive optimization of wavefront reconstruction algorithms for closed loop adaptive optical systems," in *Proceedings of the SPIE Conference on Adaptive Optical System Technologies*, pp. 1174–1185, SPIE, (Kona, Hawaii), March 1998.
13. R.F.M.M.Hamelinck, N. Rosielle, J. Kappelhof, B. Snijders, and M. Steinbuch, "Large adaptive deformable membrane mirror with high actuator density," in *Proceedings of the SPIE conference on Astronomical telescopes and instrumentation*, SPIE, (Glasgow, UK), June 2004.
14. M. Loktev, D. W. Monteiro, and G. Vdovin, "Comparison study of the performance of piston, thin plate and membrane mirrors for correction of turbulence-induced phase distortions," *Optics Communications* **192**, pp. 91–99, May 2001.
15. D. Fried, "Statistics of a geometric representation of wavefront distortion," *Journal of the Optical Society of America* **55**, pp. 1427–1435, November 1965.
16. A. Glindemann, R. Lane, and J. Dainty, "Simulation of time-evolving speckle patterns using kolmogorov statistics," *Journal of modern optics* **40**, pp. 2381–2388, 1993.
17. C. M. Harding, R. A. Johnston, and R. G. Lane, "Fast simulation of a kolmogorov phase screen," *Applied optics* **38**, pp. 2161–2170, April 1999.
18. N. Doelman, K. Hinnen, F. Stoffelen, and M. Verhaegen, "Optimal control strategy to reduce the temporal wavefront error in ao systems," in *Proceedings of the SPIE conference*, June 2004.
19. J. W. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Transactions on circuits and systems* **25**, pp. 772–781, September 1978.
20. M. H. Hayes, *Statistical digital signal processing and modeling*, John Wiley & Sons, New York, 1996.
21. G. H. Golub and C. F. V. Loan, *Matrix computations*, Johns Hopkins University Press, Baltimore, MD, USA, 3rd ed., 1996.
22. N. Roddier, "Atmospheric wavefront simulation using zernike polynomials," *Optical Engineering* **29**, pp. 1174–1180, October 1990.
23. R. D. da Cunha and T. Hopkins, "Parallel overrelaxation algorithms for systems of linear equations," in *Transputing '91 V.1*, P. H. Welch, D. Stiles, T. L. Kunii, and A. Bakkers, eds., pp. 159–169, IOS Press, 1991.