SC42025 Filtering & Identification

Practical assignment

Report

Author (alphabetic):	*Student numbers:*
Iurie Coroli	4552245
Kwabena Ofori	4031601

January 24, 2019

TU Delft

Contents

Abstract

High resolution imaging in the visible spectrum from ground-based telescopes is seriously hampered by atmospheric turbulence. Adaptive optics (AO) is the system that corrects in real-time atmospheric aberrations as displayed in Figure 1. Incoming light is split in two beams and directed toward a wavefront sensor (WFS) and the instrument camera. The information provided by the sensors is processed to estimate a future wavefront and converted into voltage for the actuators located under a deformable mirror (DM). The DM flattens the wavefront in order to retrieve the original image quality as if there was no atmosphere.

Modeling atmospheric turbulence plays key role in AO system performance. In this report, three methods of data-driven turbulence modelling are studied. In Chapter 1 a random walk model of turbulence is studied, and control of deformable mirror based on this model is derived.
In Chapter 2 so-called frozen-flow model is studied, which takes into account spatio-temporal correlation of the turbulent wavefront. Based on this model a Kalman observer is designed, and from the designed observer optimal input is computed.
In Chapter 3 subspace identification technique N4SID is applied in order to obtain the model of turbulent wavefront. Obtained model is the used in H2-optimal control to minimize the residual wavefront covariance.
In Chapter 4 the performance of the three methods is compared.
In writing this report the book [1] was extensively used.

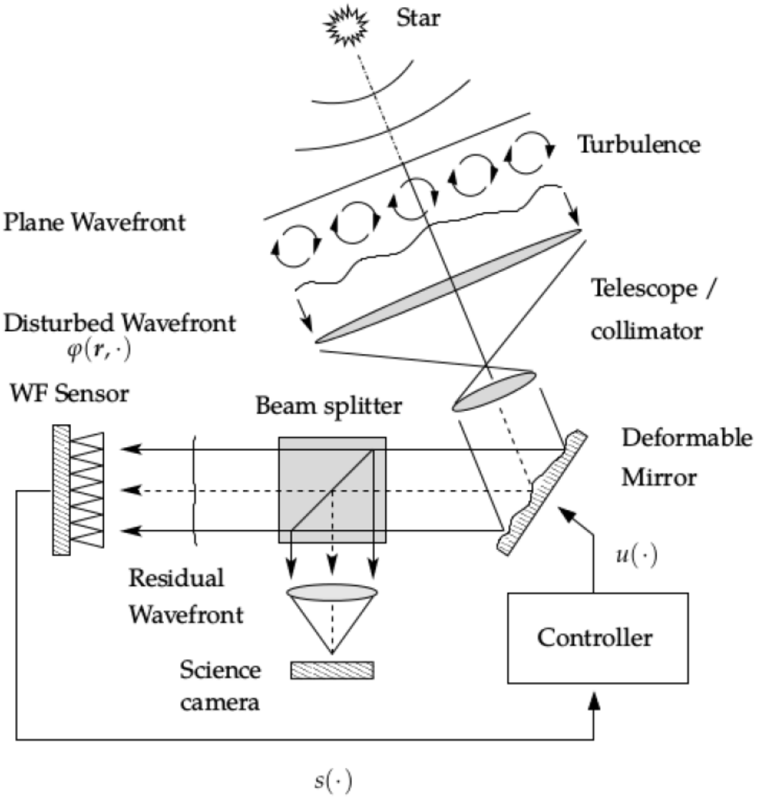

Figure 1: Schematic of adaptive optics[2]

1 | Static wavefront reconstruction

In this chapter, static wavefront reconstruction is studied. First, the wavefront at time instance k, $\phi(k)$, is reconstructed from the slopes measurement, $s(k)$. The following model is assumed:

$$s(k) = G\phi(k) + e(k),$$

where $s(k)$ is measurement of Shack-Hartmann sensor. $G \in R^{2p^2 \times (p+1)^2}$ contains +1, -1 entries and corresponds to finite-difference operation. $e(k)$ is zero-mean white Gaussian noise with covariance $\sigma_e^2 I$. It is also known that the noise covariance is Σ_e Least squares approach will be used to solve this problem.

1.1 Static reconstruction

First, a cost function J has to be defined. One possible way is to minimize the squared 2-norm of the residual $e(k)$ and hence to define cost function as

$$J = \left\| L^{-1}s(k) - L^{-1}G\phi(k) \right\|_2^2, \tag{1.1}$$

where $L = \sigma_e I$. However, in this way the solution is not unique, since G is not invertible. An additional constraint has to be added to the cost function, namely some function of the squared 2-norm of the estimated wavefront. The known covariance of the wavefront $E[\phi(k)\phi(k)^T] = C_\phi(0)$ should also be used. This unknown function can be composed by introducing a second "dummy" equation:

$$\phi(k) = C_\phi^{1/2}\mu(k),$$

where $\mu(k)$ $(0, I)$. Then if $\lambda\|\mu\|_2^2 = \lambda\left\|C_\phi^{-1/2}\phi(k)\right\|_2^2$ is added to the Equation (1.1), the new cost function will provide a unique solution of the least squares problem:

$$J = \left\| L^{-1}s(k) - L^{-1}G\phi(k) \right\|_2^2 + \lambda\left\| C_\phi(0)^{-1/2}\phi(k) \right\|_2^2. \tag{1.2}$$

Here, λ represents a weighting factor between the first term (strict data fit) and the second term (norm of $\phi(k)$ up to a multiplicative factor). The parameter λ indicates the reliability of the measurement of $C_\phi(0)$. The solution can be found by explicitly writing the norms in the cost function and taking the gradient with respect to ϕ:

$$\begin{aligned} J &= \left\| L^{-1}s(k) - L^{-1}G\phi(k) \right\|_2^2 + \lambda\left\| C_\phi(0)^{-1/2}\phi(k) \right\|_2^2 \\ &= (s(k)^T L^{-T} - \phi(k)^T G^T L^{-T})(L^{-1}s(k) - L^{-1}G\phi(k)) + \lambda\phi(k)^T C^{-T/2}C^{-1/2}\phi(k) \\ &= s(k)^T W_e s(k) - s(k)^T W_e G\phi(k) - \phi(k)^T G^T W_e s(k) + \phi(k)^T (G^T W_e G + \lambda W_\phi)\phi(k) \end{aligned}$$

$$\nabla_\phi J = 2(G^T W_e G + \lambda W_\phi)\phi(k) - 2G^T W_e s(k) = 0 \tag{1.3}$$

Here, $W_e = (LL^T)^{-1}$ and $W_\phi = (C_\phi(0)^{1/2}C_\phi(0)^{T/2})^{-1}$. From (1.3) the expression for the optimal estimator can be derived:

$$\hat{\phi} = (G^T W_e G + \lambda W_\phi)^{-1} G^T W_e s(k) \tag{1.4}$$

Exactly the same result could be derived if the cost function is reformulated to match the usual weighted least squares format:

$$\min_\phi \left\| \underbrace{\begin{bmatrix} -\frac{1}{\sigma_e}G \\ \sqrt{\lambda}C^{-1/2} \end{bmatrix} \phi(k)}_{L^{-1}Fx(k)} + \underbrace{\begin{bmatrix} \frac{1}{\sigma_e} \\ 0 \end{bmatrix} s(k)}_{L^{-1}y(k)} \right\|_2^2$$

$$\hat{\phi} = (F^T L^{-T} L^{-1} F)^{-1} F^T L^{-T} L^{-1} s(k)$$

Writing the matrices of the redefined problem yields Equation (1.4).

1.2 Optimal control calculation

The current goal is to control the deformable mirror in order to improve the quality of the received image. By controlling the actuators of the deformable mirror, a wavefront is induced and is denoted by $\phi_{DM}(k)$. A one-step delay is assumed between the time at which we apply the control inputs and the time at which the wavefront is induced, i.e.

$$\phi_{DM}(k) = Hu(k-1) \tag{1.5}$$

A random walk propagation of the atmosphere is also assumed, i.e.

$$\hat{\phi}(k+1|k) = \phi(k) \tag{1.6}$$

The residual wavefront is defined as

$$\epsilon(k) = \phi(k) - \phi_{DM}(k), \tag{1.7}$$

and the two following assumptions are made: $E[\epsilon(k)] = 0$ and $E[\epsilon(k)\epsilon(k)^T] = C_\phi(0)$. In order to obtain the highest image resolution, minimization of the variance of the residual wavefront is desired. For this purpose the optimal input command u(k) is determined from a least-squares problem as a function of estimated residual $\hat{\epsilon}(k)$, the past command $u(k-1)$ and the matrix H. In order to reformulate this problem in least-squares format, substitute Equation (1.6) and Equation (1.5) into Equation (1.7). As a result, the following expression for the residual estimate can be written:

$$\hat{\epsilon}(k) = \hat{\phi}(k+1|k) - Hu(k-1) \tag{1.8}$$

Furthermore, express $\hat{\phi}(k+1|k)$ from the expression for $\hat{\epsilon}(k+1)$:

$$\begin{aligned} \hat{\epsilon}(k+1) &= \hat{\phi}(k+1|k) - \phi_{DM}(k+1) \\ \hat{\phi}(k+1|k) &= \hat{\epsilon}(k+1) + Hu(k) \end{aligned}$$

Substitution of this relation into Equation (1.8) and rearranging, yields:

$$\hat{\epsilon}(k+1) = (\hat{\epsilon}(k) + Hu(k-1)) - Hu(k)$$

Finally, an assumption is made that $E[\hat{\epsilon}(k+1)\hat{\epsilon}(k+1)^T] = E[\epsilon(k+1)\epsilon(k+1)^T] = E[\epsilon(k)\epsilon(k)^T] = C_\phi(0)$. This yields the following weighted least squares problem:

$$\begin{aligned} &\min_{u(k)} \hat{e}(k+1)\hat{e}(k+1)^T \\ &\text{where } \hat{e}(k+1) = \underbrace{C_\phi(0)^{-1/2}(\hat{\epsilon}(k) + Hu(k-1))}_{L^{-1}y(k)} - \underbrace{C_\phi(0)^{-1/2}Hu(k)}_{L^{-1}Fx(k)} \end{aligned}$$

The minimizing input is given by the following equation:

$$\hat{u}(k) = (H^T W_\phi H)^{-1} H^T W_\phi [\hat{\epsilon}(k) + Hu(k-1)], \tag{1.9}$$

where $W_\phi = (C_\phi(0)^{1/2} C_\phi(0)^{T/2})^{-1}$.

By expanding the brackets in Equation (1.9) and substituting Equation (1.8) the following relation is obtained:

$$\delta u(k) = (H^T W_\phi H)^{-1} H^T W_\phi \hat{\epsilon}(k)$$

Now the Equation (1.4) can be used. By looking at the structure of adaptive optics control-loop it can be seen that when the loop is closed, the wavefront sensor does not measure $\phi(k)$, but instead it measures the difference between the incoming wavefront and the wavefront induced by the deformable mirror, i.e. $\epsilon(k)$. Hence Equation (1.4) can be used in order to estimate the residual wavefront. Substitution of Equation (1.4) for $\hat{\epsilon}(k)$ yields:

$$\delta u(k) = \underbrace{(H^T W_\phi H)^{-1} H^T W_\phi (G^T W_e G + \lambda W_\phi)^{-1} G^T W_e}_{M} s(k) \tag{1.10}$$

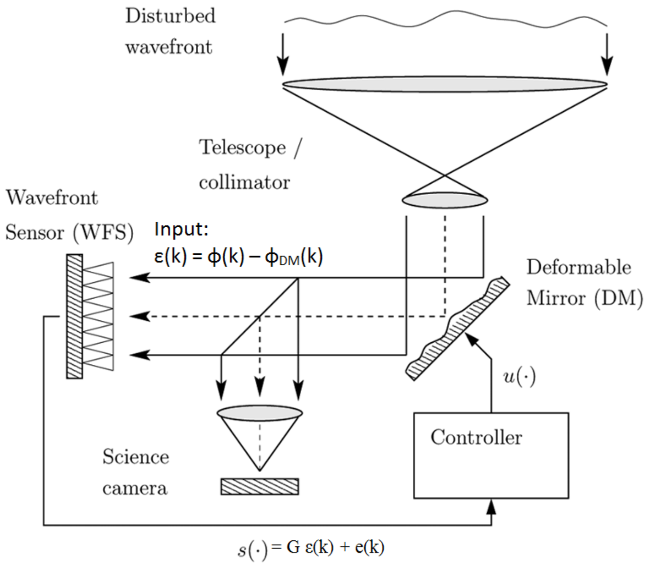

Figure 1.1: Adaptive optics closed loop: Shack-Hartmann sensor now measures $\epsilon(k)$.

This result can be used in order to simulate the closed loop behaviour of the adaptive optics system. The mirror is assumed to be initially flat. The incoming wavefront affected by the atmospheric turbulence is given for a 1000 time-samples. Control action is calculated in a loop described by the following pseudo-code:

```
1. Calculate $\epsilon(k)$
   $\epsilon(k) = \phi(k) - Hu(k-1)$
2. Calculate $s(k)$
   $s(k) = \epsilon(k) + e(k)$
3. Calculate $u(k)$ from Equation (1.10)
   $u(k) = Ms(k) + u(k-1)$
```

Before computing the variance of the residual wavefront, it is also necessary to remove its mean value. The important quantity is how much the wavefront varies around its mean, while mean itself is not of interest.
The number of elementary operations is crucial for online computation of optimal input. For this method, M matrix is fixed and pre-computed and has size m^2x2p^2, $s(k)$ is a column vector with $2p^2$ entries. Hence, computation of optimal input as $Ms(k) + u(k-1)$ requires $4p^2m^2$ elementary operations, which in the given case equals to 7056.

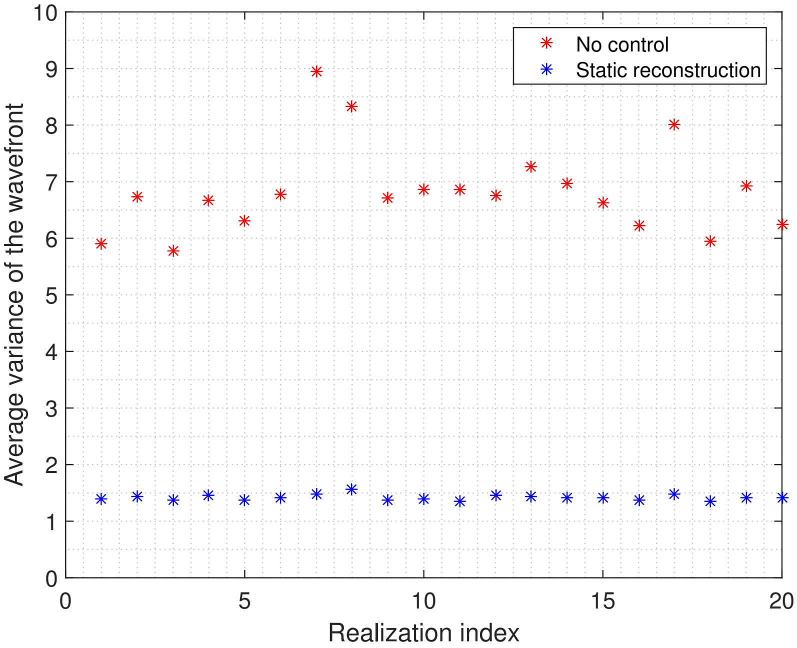

Figure 1.2: Comparison of wavefront variances for 20 turbulence realizations. Horizontal axis represents index of the realization, vertical axis represents average variance of the wavefront. Red plot represents variance of the wavefront without any control applied, blue represents variance, when control based on static wavefront reconstruction is applied.

The result of this control method can be seen in Figure 1.2. As observed, the variance of the wavefront was gradually reduced, which means the image has a better quality. However, the result can be improved by taking into account dynamics of the turbulence. This is done in further chapters.

The code for this method is provided in Appendix A.

2 | Vector Auto-Regressive model of order 1

2.1 Introduction

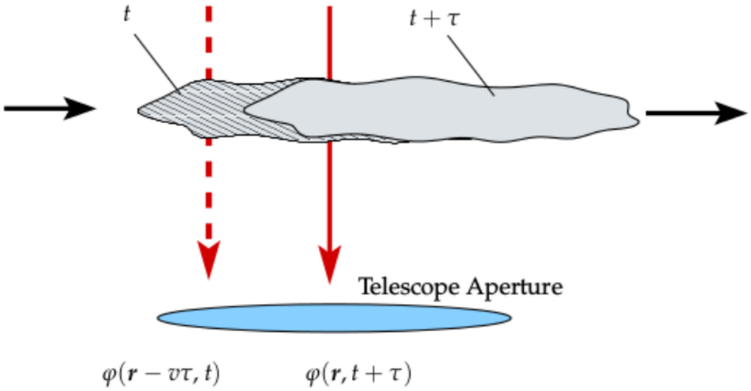
t
$t+\tau$
Telescope Aperture
$\varphi(\boldsymbol{r} - v\tau, t)$
$\varphi(\boldsymbol{r}, t+\tau)$

Figure 2.1: Frozen flow model of atmospheric turbulence

In Chapter 1 random-walk propagation of atmospheric turbulence was considered, and based on its prediction the loop was closed. This method, however, ignores all kind of spatio-temporal correlation. In this chapter the frozen flow model (Figure 2.1) of turbulence is assumed in order to make a more accurate prediction and to close the loop. The following Vector Auto-Regressive (VAR) model is studied:

$$\phi(k+1) = A\phi(k) + w(k) \tag{2.1}$$

where $A \in \mathbb{R}^{(p+1)^2 \times (p+1)^2}$, and $w(k) \sim \mathcal{N}(0, C_w)$. Moreover, the signal $w(k)$ is assumed to be uncorrelated with the measurement noise $e(k)$ and the turbulent wavefront $\phi(k)$:

$$E[w(k)e(k)^T] = 0, \qquad E[w(k)\phi(k)^T] = 0 \tag{2.2}$$

2.2 Estimation of model parameters

Covariance matrices of turbulent wavefront are denoted by

$$C_\phi(0) = E[\phi(k)\phi(k)^T], \tag{2.3}$$

$$C_\phi(1) = E[\phi(k+1)\phi(k)^T], \tag{2.4}$$

and are known. First, system matrix A is derived from covariance matrices by explicitly writing $E[\phi(k+1)\phi(k)^T]$ using system model (2.1):

$$\begin{aligned} C_\phi(1) &= E[\phi(k+1)\phi(k)^T] \\ &= E[(A\phi(k) + w(k))\phi(k)^T] \\ &= AE[\phi(k)\phi(k)^T] + E[w(k)\phi(k)^T] \\ &= AC_\phi(0). \end{aligned}$$

From this relation matrix A can be expressed as

$$A = C_\phi(1)C_\phi(0)^{-1}. \tag{2.5}$$

Further, in order to construct a Kalman filter, the unknown covariance matrix C_w of the signal $w(k)$ must be expressed in terms of known matrices $C_\phi(0)$, $C_\phi(1)$ and A. Turbulent wavefront is assumed to be a Wide-Sense-Stationary (WSS) signal, i.e. $E[\phi(k+1)\phi(k+1)^T] = E[\phi(k)\phi(k)^T] = C_\phi(0)$. Using system model (2.1), Equation (2.2) and the above assumption, $E[\phi(k+1)\phi(k+1)^T]$ can be expressed as follows:

$$\begin{aligned} C_\phi(0) &= E[\phi(k+1)\phi(k+1)^T] \\ &= E[(A\phi(k) + w(k))(\phi(k)^T A^T + w(k)^T)] \\ &= AE[\phi(k)\phi(k)^T]A^T + E[w(k)w(k)^T] \\ &= AC_\phi(0)A^T + C_w. \end{aligned}$$

From this matrix C_w can be expressed:

$$C_w = C_\phi(0) - AC_\phi(0)A^T \tag{2.6}$$

2.3 State space model

The goal of adaptive optics systems is to minimize the covariance of residual wavefront $\epsilon(k) = \phi(k) - \phi_{DM}(k)$. Moreover, in closed loop wavefront sensor measures residual wavefront. Hence, in order to derive an optimal control algorithm, it is the residual wavefront $\epsilon(k)$ that needs to be estimated. A state space model where ϵ is the state variable is constructed for this purpose. Using the system model (2.1) and relation between $\phi(k)$ and $\phi_{DM}(k)$ the following relation is obtained:

$$\begin{aligned} \epsilon(k+1) + \phi_{DM}(k+1) &= A(\epsilon(k) + \phi_{DM}(k)) + w(k) \\ \epsilon(k+1) &= A\epsilon(k) + A\phi_{DM}(k) - \phi_{DM}(k+1) + w(k) \\ &= A\epsilon(k) + AHu(k-1) - Hu(k) + w(k) \\ &= A\epsilon(k) + \underbrace{\begin{bmatrix} -H & AH \end{bmatrix}}_{B} \begin{bmatrix} u(k) \\ u(k-1) \end{bmatrix} + w(k) \end{aligned} \tag{2.7}$$

Since in the closed loop operation the measured quantity is residual wavefront $\epsilon(k)$, the output equation reads as

$$s(k) = G\epsilon(k) + e(k). \tag{2.8}$$

2.4 Kalman filter

In this section an expression is given for Kalman filter based on the state-space model from Equation (2.7). The observer equation is defined as

$$\begin{aligned} \hat{\epsilon}(k+1) &= A\hat{\epsilon}(k) + Bu(k) + K[s(k) - \hat{s}(k)] \\ \hat{s(k)} &= G\hat{\epsilon(k)}, \end{aligned} \tag{2.9}$$

where K is the Kalman gain. Using the equations (2.7) and (2.8), and substituting them into Equation (2.9), will result in

$$\hat{\epsilon}(k+1) = (A - KG)\hat{\epsilon}(k) - Hu(k) + AHu(k-1) + Ks(k). \tag{2.10}$$

To compute the Kalman gain, the covariance matrices must first be defined. Using Equation (2.2) and known relations $E[w(k)w(k)^T] = C_w$ and $E[e(k)e(k)^T] = \sigma_e^2 I$, the covariance matrices are defined as follows:

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} = \begin{bmatrix} E[w(k)w(k)^T] & E[w(k)e(k)^T] \\ E[e(k)w(k)^T] & E[e(k)e(k)^T] \end{bmatrix} = \begin{bmatrix} C_w & 0 \\ 0 & \sigma^2 I \end{bmatrix} > 0 \tag{2.11}$$

The estimator covariance matrix $P = E[(x(k) - \hat{x}(k-1))((x(k) - \hat{x}(k-1))^T]$ can be obtained by solving the Riccati equation

$$P = APA^T + Q - (S + APG^T)(GPG^T + R)^{-1}(S + APG^T)^T \tag{2.12}$$

Kalman gain is then obtained from the following equation:

$$K = (S + APG^T)(GPG^T + R)^{-1} \tag{2.13}$$

2.5 Computing optimal input

In this section a method is given to minimize the residual wavefront online. For this purpose linear least squares approach will be used. First, observer equation (2.10) is reformulated in such way that all the measured quantities at k-th time instance are collected together:

$$\hat{\epsilon}(k+1|k) = \underbrace{[(A - KG)\hat{\epsilon}(k) + AHu(k-1) + Ks(k)]}_{y(k)} - \underbrace{H}_{F}\underbrace{u(k)}_{x}$$

The goal of this adaptive optics system is to minimize the residual wavefront covariance, i.e.

$$\min_{u(k)} \left\| [(A - KG)\hat{\epsilon}(k) + AHu(k-1) + Ks(k)] - Hu(k) \right\|_2^2 \tag{2.14}$$

The least squares solution of this problem is

$$\hat{u}(k) = H^{-1}[(A - KG)\hat{\epsilon}(k) + AH\hat{u}(k-1) + Ks(k)] \tag{2.15}$$

The MATLAB code implementing this method is given in files *computeKalmanAR.m* and *AOloopAR.m*, provided in Appendix B.

Performance of this method has been tested on 20 turbulence realizations. The result is provided in Figure 2.2.

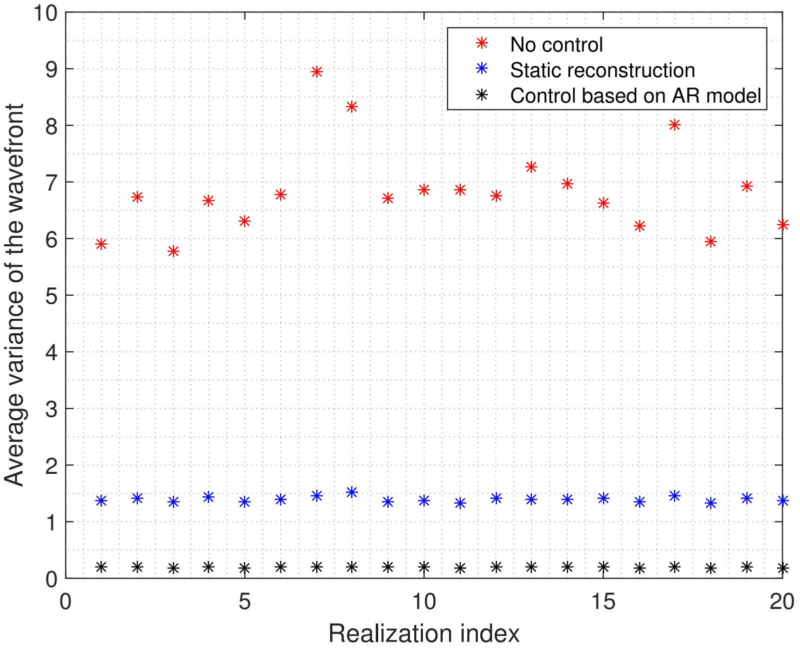

Figure 2.2: Performance test of control based on Kalman filter for 20 turbulence realizations

As it can be observed in Figure 2.2, the performance has gradually improved. Using a Kalman filter based on a VAR model, the average residual wavefront covariance for 20 turbulence realizations has now become 0.1941. In comparison to the method described in Chapter 1, which yielded a covariance of 1.3968, a decrease of 86% is obtained.

3 | Subspace identification

3.1 Introduction

In this chapter subspace identification technique will be used to derive state space model of atmospheric turbulence from open loop wavefront sensor data. The following model is assumed:

$$\begin{aligned} \phi(k+1) &= A\phi(k) + Kv(k) \\ s(k) &= C\phi(k) + v(k), \end{aligned}$$

where $\phi(k)$ is the state vector representing wavefront, $s(k)$ is open loop wavefront measurement, v(k) is zero-mean white noise process with identity covariance matrix (innovation signal), and K is Kalman gain matrix.

3.2 Theoretical description of PO-MOESP

A combination of $PO-MOESP$ and $N4SID$ methods will be used in this chapter. The available data is split into two sets: identification set and validation set. From now until validation chapter only identification data set will be taken into account. First, Hankel matrix of outputs $S_{0,2s,N}$ is constructed:

$$S_{0,2s,N} \begin{bmatrix} s(0) & s(1) & \cdots & s(N-1) \\ s(1) & s(2) & \cdots & s(N) \\ \vdots & \vdots & \ddots & \ddots \\ s(2s-1) & s(2s) & \cdots & s(N+2s-2) \end{bmatrix}.$$

Here, s - starting index of Hankel matrix must not be confused with system output, s(k). Moreover, since the identified system has 47 outputs, each entry in the above defined Hankel matrix is a column vector with 47 entries. Because of this, the constructed matrix is block-Hankel.

The above Hankel matrix is then split into two Hankel matrices: $S_{0,s,N}$ containing only first s rows of $S_{0,2s,N}$, and $S_{s,s,N}$, containing rows from $(s+1)$ to $2s$. The data equation reads as

$$S_{s,s,N} = \mathcal{O}_s \Phi_{s,N} + T_v V_{s,s,N}, \tag{3.1}$$

where $\mathcal{O}_s$ is the extended observability matrix, $V_{s,s,N}$ is the innovation signal Hankel matrix starting from index s.

$$\mathcal{O}_s = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{s-1} \end{bmatrix} \qquad T_v = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ CK & I & 0 & \cdots & 0 \\ CAK & CK & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \\ CA^{s-2}K & CA^{s-3}K & \cdots & CK & I \end{bmatrix}$$

In order to perform system identification, innovation signal Hankel matrix must be removed. This is achieved by multiplying both sides of Equation (3.1) by transposed "past outputs" Hankel matrix, $S_{0,s,N}$, since innovation signal $v(k)$ is assumed to be uncorrelated with the output.

$$S_{s,s,N} S_{0,s,N}^T = \mathcal{O}_s \Phi_{s,N} S_{0,s,N}^T \tag{3.2}$$

The C matrix up to unknown similarity transformation can be extracted from SVD of $S_{s,s,N} S_{0,s,N}^T$, because $range\{S_{s,s,N} S_{0,s,N}\} = range\{O_s\}$. If

$$S_{s,s,N} S_{0,s,N}^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix},$$

then C_T matrix is equal to the first l rows and first n columns of the U matrix, where l is the number of outputs and n is the number of desired system order. The matrix A_T up to unknown similarity transformation can be computed from the following overdetermined equation:

$$U(1:(s-1)l,:)A_T = U(l+1:sl,:)$$

3.3 N4SID approach

In theory it is possible to simply multiply $S_{s,s,N}$ and $S_{0,s,N}^T$ and take SVD decomposition of the product. However, in this case very large dimensions of matrices lead to numerical errors. For instance, matrix A computed directly from SVD of $S_{s,s,N}S_{0,s,N}^T$ is unstable. This issue can be solved by using RQ-factorization. However, estimation of Kalman gain from PO-MOESP method requires knowledge of covariance matrices of process and measurement noise, and also their cross-covariance. Because this information is not available, system matrices, noise covariance matrices and Kalman gain will be computed using N4SID approach. First, RQ decomposition of stacked output Hankel matrices is taken:

$$\begin{bmatrix} S_{0,s,N} \\ S_{s,s,N} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix},$$

Then, SVD decomposition of $R_{21}R_{11}^{-1}S_{0,s,N} = U_n\Sigma_n V_n^T$ is taken, where n is the desired order of identified system. Next, state sequence $\hat{\Phi}_{s,N}$ is estimated as $\hat{\Phi}_{s,N} = \Sigma_n^{1/2}V_n^T$. The estimated state sequence is then used to estimate system matrices from least squares problem:

$$\min_{A_T,C_T} \left\| \begin{bmatrix} \hat{\Phi}_{s+1,N} \\ S_{s,1,N-1} \end{bmatrix} - \begin{bmatrix} A_T \\ C_T \end{bmatrix} \hat{\Phi}_{s,N-1} \right\|_F^2 .$$

The solution to this problem is

$$\begin{bmatrix} \hat{A}_T \\ \hat{C}_T \end{bmatrix} = \begin{bmatrix} \hat{\Phi}_{s+1,N} \\ S_{s,1,N-1} \end{bmatrix} \hat{\Phi}_{s,N-1}^T (\hat{\Phi}_{s,N-1}\hat{\Phi}_{s,N-1}^T)^{-1}.$$

In order to calculate estimated Kalman gain, it is first necessary to estimate auto-covariance matrices of process and measurement noise, Q and R, and their cross-covariance matrix, S. First, residuals of least squares problem are calculated:

$$\begin{bmatrix} \hat{W}_{s,1,N-1} \\ \hat{V}_{s,1,N-1} \end{bmatrix} = \begin{bmatrix} \hat{\Phi}_{s+1,N} \\ S_{s,1,N-1} \end{bmatrix} - \begin{bmatrix} \hat{A}_T \\ \hat{C}_T \end{bmatrix} \hat{\Phi}_{s,N-1}.$$

These residuals are now used to estimate the covariance matrices as follows:

$$\begin{bmatrix} \hat{Q} & \hat{S} \\ \hat{S}^T & \hat{R} \end{bmatrix} = \frac{1}{N} \begin{bmatrix} \hat{W}_{s,1,N-1} \\ \hat{V}_{s,1,N-1} \end{bmatrix} \begin{bmatrix} \hat{W}_{s,1,N-1}^T & \hat{V}_{s,1,N-1}^T \end{bmatrix}.$$

The estimated covariance matrices are now used to calculate solution of the following Riccati equation:

$$\hat{P} = \hat{A}_T\hat{P}\hat{A}_T^T + \hat{Q} - (\hat{S} + \hat{A}_T\hat{P}\hat{C}_T^T)(\hat{C}_T\hat{P}\hat{C}_T^T + \hat{R})^{-1}(\hat{S} + \hat{A}_T\hat{P}\hat{C}_T^T)^T. \tag{3.3}$$

Finally, Kalman gain is calculated from the solution of Equation (3.3):

$$\hat{K} = (\hat{S} + \hat{A}_T\hat{P}\hat{C}_T^T)(\hat{R} + \hat{C}_T\hat{P}\hat{C}_T^T)^{-1}.$$

3.4 Comments on using the method and results

The MATLAB code *pomoesp.m*, implementing the above described method, is provided in Appendix C.
The supplied data must be detrended, i.e. mean value removed, before being used for identification. In general, it also needs to be decimated if sampling frequency is too high, i.e. low-pass filter is applied and sampling frequency is reduced k times by taking every $k-th$ sample in data sequence. However, in the described problem nothing is known about sampling frequency.

The use of subspace identification method is not trivial at all. Even though it provides accurate observer, there are still degrees of freedom left: choice of output system order and number of block rows of Hankel matrices. Ideally, system order can be accurately enough determined from the number of nonzero singular values of SVD decomposition of corresponding Hankel matrices. However, this is not the case in the given example, as can be seen in Figure 3.1.

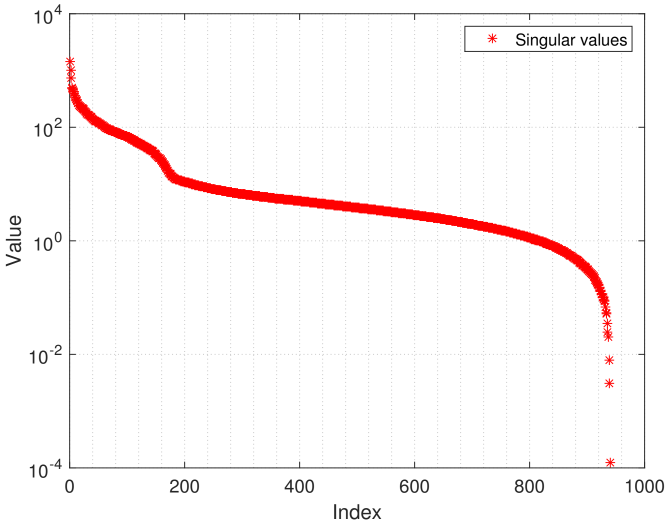

Figure 3.1: Singular values of $S_{s,s,N}S_{0,s,N}^T$

A measure of quality of the identified model is resulting residual wavefront covariance, when identified system matrices are used in H2-optimal control. A test has been made in order to determine the optimal model order. Using turbulence realization number 5, covariance of residual wavefront has been computed for various system orders from 10 to 800. During the test, number of block Hankel rows was fixed to value $s = 20$. The result of this test is represented in Figure 3.2.

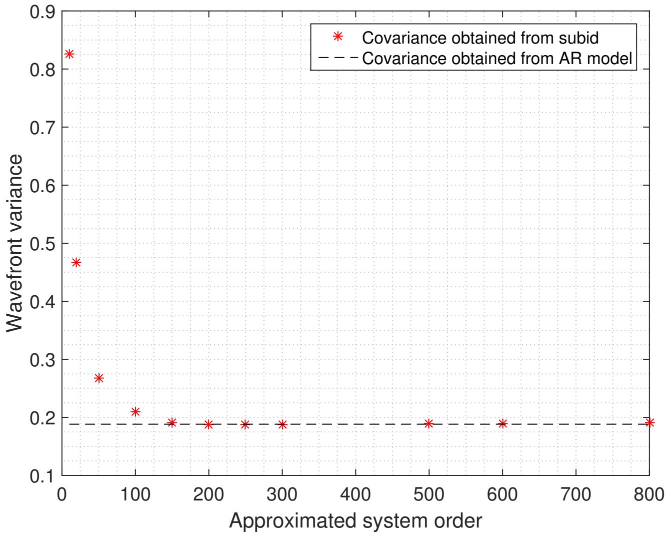

Figure 3.2: Test of optimal system order for s = 20. Horizontal axis represents chosen system order. Vertical axis represents residual wavefront covariance. Red plot shows covariance obtained from H2-optimal control. Black line represents covariance obtained using AR-model and is given for reference.

Another indication of model quality is variance-accounted-for. The closer it it to 100, the better is the prediction of the constructed observer. A test has been made for a number of observer orders varying from 10 to 700. The result is represented in Figure 3.3

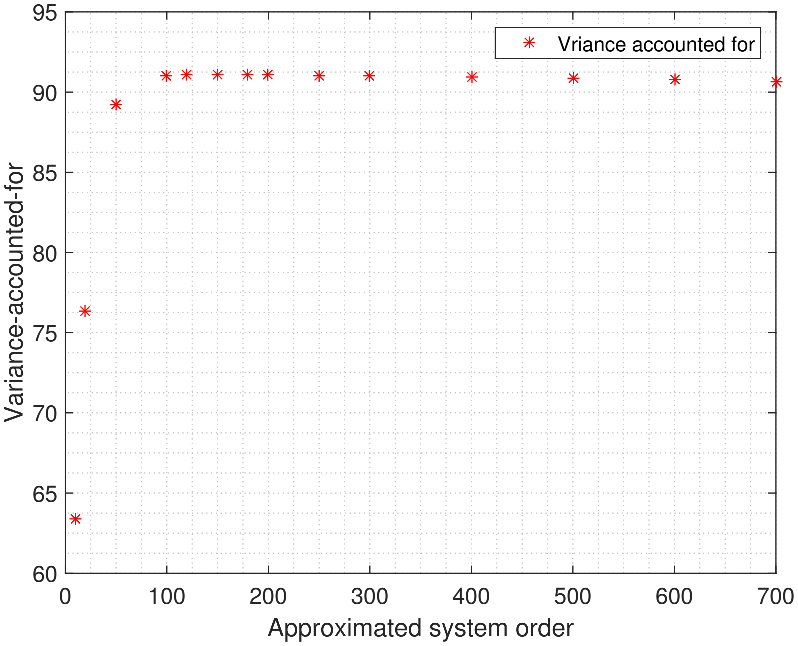

Figure 3.3: Test of optimal system order for s = 20. Horizontal axis represents chosen system order. Vertical axis represents variance-accounted-for.

As it can be seen from Figure 3.2, optimal system order is $n = 200$. It will be chosen for the simulation.

The second degree of freedom, number of block Hankel rows s, has also been studied. In the test, system order has been fixed to $n = 200$. Number of block Hankel rows varied between 10 and 100 and covariance of the residual wavefront has been calculated using obtained system matrices. The result of this test is represented in Figure 3.4.

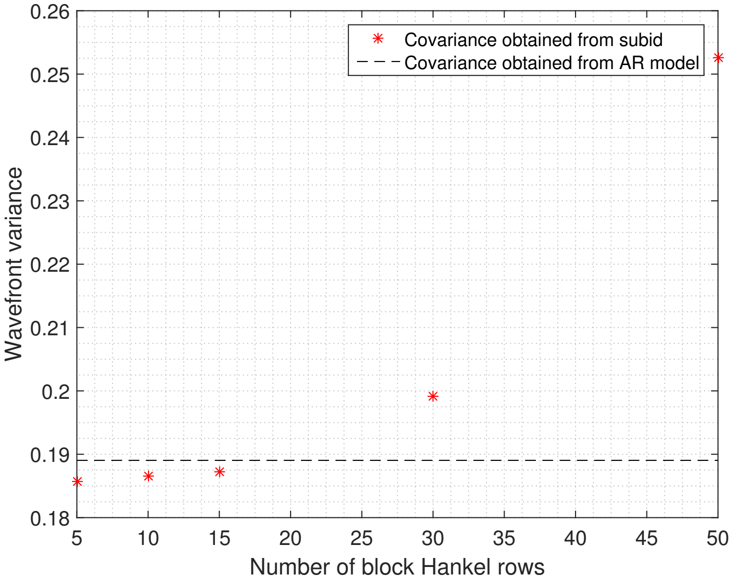

Figure 3.4: Test of optimal nuber of block Hankel rows for n = 200. Horizontal axis represents chosen value of rows. Vertical axis represents residual wavefront covariance. Red plot shows covariance obtained from H2-optimal control. Black line represents covariance obtained using AR-model and is given for reference.

From the two tests above, optimal parameters for subspace identification were chosen to be $n = 200, s = 5$. System matrices obtained with these settings were used in H2-optimal control. Resulting residual wavefront covariance is represented in Figure 3.5.

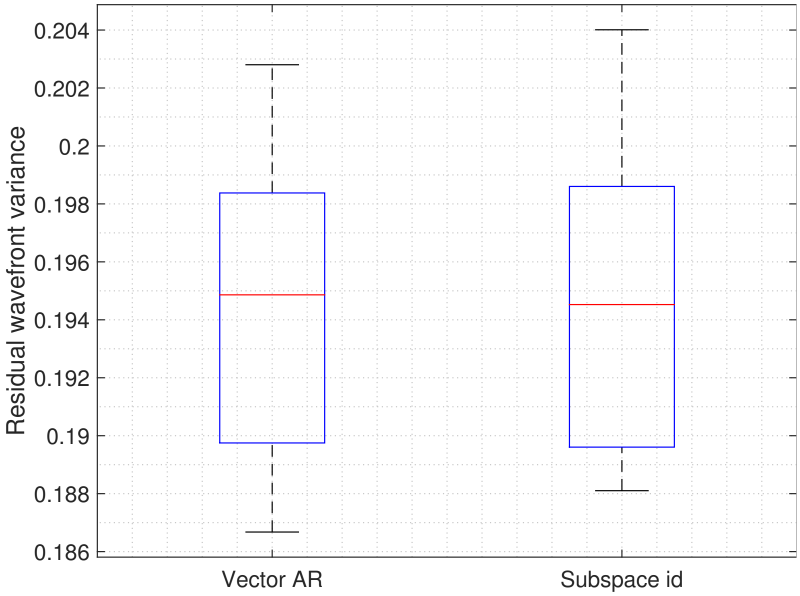

Figure 3.5: Boxplot comparison of performance of AR model and subspace identification technique

As it can be seen, the variance of the residual wavefront has decreased comparing to the methods implemented before. Results are analysed more thoroughly in Chapter 4.

4 | Results analysis

In this section the performance of the systems described in this report will be evaluated. In Chapter 1 a random walk model of wavefront was constructed and based on this model control has been derived. The optimal value of parameter λ was derived to be $\lambda = 40$. In Chapter 2 vector Auto-Regressive model of turbulence was constructed, and based on this model optimal control was derived. In Chapter 3 subspace identification of turbulence model was performed and observer of order $n = 200$ was derived. The derived model was then used in H2-optimal control algorithm to minimize residual wavefront covariance. Tests have been performed on 20 turbulence realizations. The result is depicted in box-plot in Figure 4.1 and in Table 4.1.

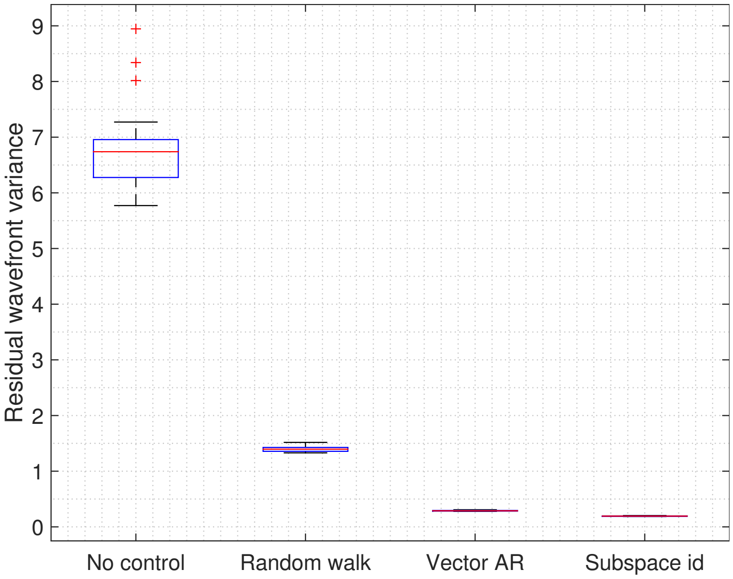

Figure 4.1: Comparison of modeling and control methods for 20 turbulence realizations each

Table 4.1: Mean of variance of residual wavefront for 20 turbulence realizations

Nocontrol	Random walk	V-AR	Subid
6.8434	1.3974	0.1941	0.1923

It can be observed, that both the vector AR method and the subspace identification method have similar effect on minimizing the variance. Subspace identification technique in combination with H2-optimal control on average yields variance of residual wavefront that is approximately the same as the VAR model, however the method has better noise immunity.

An interesting effect can be observed from graphical representation of residual wavefront. Using turbulence realization 5 and H2-optimal control based on matrices obtained from subspace identification, order $n = 200$, a residual wavefront for 1000 time samples was calculated. The result is depicted in Figure 4.2.

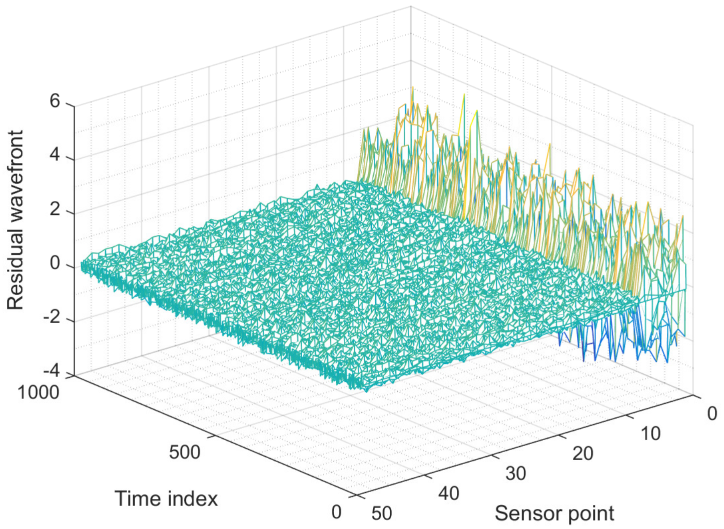

Figure 4.2: Visualisation of residual wavefront time evolution.

It can be observed that the first 7 points at which the residual is measured always have a higher covariance, then at the rest of the sensor. This fact can be easily explained. Once the wavefront is measured at 49 points of the Shack-Hartmann sensor, the constructed observer is able to predict, how this measured wavefront will influence other points of the sensor later in time. However, when the wind blows (i.e. turbulence layer moves) from one direction, predictor is not able to predict what new data it will sense. The 7 samples with high covariance represent the first row of wavefront sensor array, from which the wind blows.

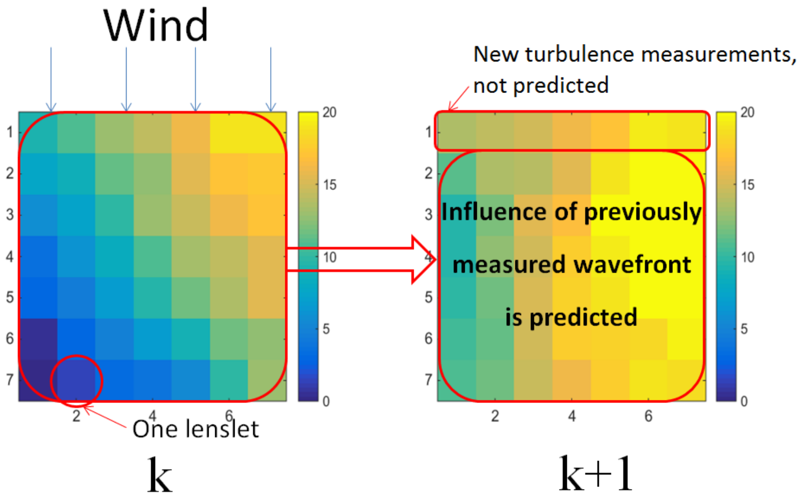

Figure 4.3: Graphical explanation: derived predictor is able to predict how measured wavefront will evolve in time over the sensor aperture, but it does not predict what new data it will sense.

5 | Conclusion

The goal of this project was to analyze the performance of three different methods for wavefront reconstruction from slopes measurements performed by Shack-Hartmann sensor. The first method, analyzed in Chapter 1, is based on random walk model and ignores all kind of spatio-temporal dynamics. As expected, method one resulted in the highest minimum variance (average variance of residual is 1.3977). The second method, analyzed in Chapter 2, is based on the frozen-flow model of turbulence. The Kalman filter is constructed for this model and an optimal controller is derived from Kalman filter output. This method has a greater performance (average variance of residual 0.1941), however it requires prior knowledge about measurement noise variance and turbulent wavefront covariance. The last analyzed method, subspace identification, has the best performance (average variance of residual 0.1923). It is also the most robust method of all methods studied, since it only relates on data, deriving all required information from it.

References

[1] V. Verdult M. Verhaegen. “Filtering and System Identification: a least-squares approach”. In: *Cambridge University Press* (2007).

Appendices

A | AoloopMVM

```matlab
function [var_eps] = AOloopMVM(G, H, C_phi_0, sigma_e, phi, lambda);
% Function for closing the loop based on random walk wavefront model
% INPUT:
%   G - matrix relation between phi and slopes measurement s
%   H - matrix relation between input u(k-1) and deformable mirror induced
%         wavefront phi_dm(k)
%   C_phi, C_phi1 - covariace matrices of wavefront phi
%   sigma_e - measurement noise variance
%   phi - incoming turbulent wavefront used for simulation only
%   lambda - weighting factor of regularization problem
% OUTPUT:
%   var_eps - residual wavefron variance
%
% SEE THE REPORT FOR THEORETICAL DESCRIPTION



    W_e = 1/sigma_e^2;

    C_phi = C_phi_0;
    W_phi = inv(sqrtm(C_phi)*sqrtm(C_phi)');

    u_opt = zeros(size(phi,1),size(phi,2));
    eps = zeros(size(phi,1),size(phi,2));
    e = zeros (size(G,2), size(phi,2));

    e = random('norm', zeros(size(G,1),1), sigma_e * ones(size(G,1),1));
    eps(:,1) = phi(:,1);
    s = G*eps(:,1) + e;
    u_opt(:,1) = inv(H'*W_phi*H)*H'*W_phi*inv(G'*W_e*G + lambda*W_phi)*G'*W_e*s;

    for i = 2:size(phi,2)
        eps(:,i) = phi(:,i) - H*u_opt(:,i-1);
        e = random('norm', zeros(size(G,1),1), sigma_e * ones(size(G,1),1));
        s = G*eps(:,i) + e;
        u_opt(:,i) = inv(H'*W_phi*H)*H'*W_phi*inv(G'*W_e*G + lambda*W_phi)*G'*W_e*s + u_opt(:,i-1);
    end

    eps = detrend(eps, 'constant');

    var_eps = mean(var(eps));
end
```

B | VAR model and Kalman gain

```
function [A,Cw,K]=computeKalmanAR(C_phi, C_phi1, G, sigma_e)
% Calculation of Kalman gain , A matrix and process noise covariance Cw
% INPUT:
%   C_phi - covariance matrix of phi(k)*phi(k)^T
%   C_phi1 - covariance matrix of phi(k)*phi(k+1)^T
%   sigma_e - variance of measurement noise
% OUTPUT:
%   A - system matrix in model phi(k+1) = Aphi*(k) + w(k)
%   Cw - process noise covariance
%   K - Kalman gain
%


A=C_phi1/C_phi;
C=G;
Cw=round(C_phi-A*C_phi*A', 8);
S=0;
R=sigma_e^2 * eye(size(G,1));


% Solve the Ricatti equation
 [P,L,K]=dare(A',C',Cw, R);
K = K';
end
```

```
function [var_eps] = AOloopAR(G, H, C_phi, C_phi1, sigma_e, phi)
% Closed loop simulation of AO system based on Kalman gain
% INPUT:
%   G - matrix relation between phi and slopes measurement s
%   H - matrix relation between input u(k-1) and deformable mirror induced
%       wavefront phi_dm(k)
%   C_phi, C_phi1 - covariace matrices of wavefront phi
%   sigma_e - measurement noise variance
%   phi - incoming turbulent wavefront used for simulation only
% OUTPUT:
%   var_eps - residual wavefron variance

%% Define necessary matrices

   W_e = 1/sigma_e^2;
   W = inv(sqrtm(C_phi)*sqrtm(C_phi)');
%     A = C_phi1\C_phi;

   %% Closing the loop

   % Allocating memory
   u_opt = zeros(size(phi,1),size(phi,2));
   eps = zeros(size(phi,1),size(phi,2));
   eps_hat = zeros(size(phi,1),size(phi,2)+1);
   e = zeros (size(G,2), size(phi,2));

   % Define matrix required for least squares solution
   M = inv(H);
```

```
    % Compute Kalman gain
    [A, Cw, K] = computeKalmanAR(C_phi, C_phi1, G, sigma_e);

    % Simulate the initial conditions of closed loop
    e = random('norm', zeros(size(G,1),1), sigma_e * ones(size(G,1),1));
    eps(:,1) = phi(:,1);
    s = G*eps(:,1) + e;
    u_opt(:,1) = M*((A-K*G)*eps_hat(:,1) + K*s);
    eps_hat(:,2) = (A-K*G)*eps_hat(:,1) - H*u_opt(:,1) + K*s;

    % Close the loop and simulate for the remaining time steps
    for i = 2:size(phi,2)
        eps(:,i) = phi(:,i) - H*u_opt(:,i-1);
        e = random('norm', zeros(size(G,1),1), sigma_e * ones(size(G,1),1));
        s = G*eps(:,i) + e;
        u_opt(:,i) = M*((A-K*G)*eps_hat(:,i) + A*H*u_opt(:,i-1) + K*s);
        eps_hat(:,i+1) = (A-K*G)*eps_hat(:,i) - H*u_opt(:,i) + A*H*u_opt(:,i-1) + K*s;
    end

    % Remove the mean of residual
    eps = detrend(eps, 'constant');

    % Compute the variance
    var_eps = mean(var(eps));
end
```

C | pomoesp.m

```
function [Aest, Cest, K, vaf] = pomoesp(sk, Nid, Nval, s, n)
% POMOESP
% Not really po-moesp, but a combination with N4SID.
% WARNING: WRITTEN SPECIALLY FOR PRACTICAL, DO NOT USE FOR GENERAL SYSTEMS
% Estimates A, C and Kalman gain matrices of an autonomous system from data.
%
% INPUT:
%    sk - data
%    Nid - amount of data used for identification (rule of thumb: 2/3 of
%    total data)
%    Nval - data used for validation
%    s - number of block rows of Hankel matrix
%    n - desired output system order
% OUTPUT:
%    Aest - estimated n-by-n A matrix
%    Cest - estimated (number-of-outputs)-by-n matrix
%   K - Kalman gain matrix, n-by-(number-of-outputs)
%    vaf - variance-accounted-for - the more, the better
%
% For theoretical description, see book "Filtering and System
% Identification: A Least Squares Approach (M. Verhaegen, V. Verdult),
% pages 332-334.


%% 1. Detrend data

skdecd=sk;
for k=1:length(sk(:,1))
    skdecd(k,:)=detrend(skdecd(k,:));
end

sk = skdecd;



%% 2. Hankel matrix and instrumental variable
length_out = size(sk,2);

% Sanity check - don't use more data then you have
if (Nid + Nval > length_out)
    error('Number of data points for identification PLUS number of data points for validation must be less than supplied data length');
end

% Number of outputs of the identified system - denoted by l in the book
nout = size(sk,1);

% Construct a big Hankel matrix of outputs. Hankel matrix will have 2*s
% block-rows, and Nid - 2*s +1 columns
y_hankel_outputs = zeros(nout*2*s, (Nid-2*s+1));
    for i = 1:Nid-2*s+1
       y_hankel_outputs(:,i) = reshape(sk(:,i:i+2*s-1), 2*s*nout,1);
    end
```

```
 % Get dimensions of output Hankel matrix - just in case
[y1, y2] = size(y_hankel_outputs);

% Instrumental variable - used to eliminate inovation signal from data
% equation
instr_var = y_hankel_outputs(1:s*nout,:);


%% 3. RQ-decomposition

% Perform RQ decomposition of the big Hankel matrix. You don't really use
% Q, so don't save it for the sake of saving memory.
R=triu(qr(y_hankel_outputs'))';

% Extract the matrices you need for further calculations
R11 = R(1:nout*s, 1:nout*s);
R21 = R(nout*s+1:end, 1:nout*s);


%% 4. SVD decomposition

% Perform SVD decomposition - will be used to calculate state sequence and
% system matrices. You can also approximate system order by looking at how
% many nonzero singular values there are in S matrix. U is not used in this
% case, so don't save it.

[~, S, V] = svd(R21*(R11\eye(length(R11)))*instr_var);


%% 5. Estimation of system matrices

% Estimated state sequence
Xest = sqrtm(S(1:n, 1:n))*V(:,1:n)';

% Extract its dimensions
[x1, x2] = size(Xest);

% Construct two chopped matrices of states - used further for least squares
Xest1 = Xest(:, 2:x2);
Xest0 = Xest(:,1:x2-1);

% Cut a part of output Hankel matrix
Ycut = y_hankel_outputs(s*nout+1:(s+1)*nout,1:y2-1);

% Estimated state matrices from least squares problem. Used solution for
% Frobenius norm - see page 302 of the book, top of the page.
sysest = [Xest1;Ycut]*pinv(Xest0);

% Extract system matrices
Aest = sysest(1:n, 1:n);
Cest = sysest(n+1:n+nout,1:n);


%% 6. Estimation of covariances and Kalman gain

```

```
resid_est = [Xest1;Ycut] - sysest*Xest0;
covest = 1/(Nid-1) * (resid_est*resid_est');
% Covariance of process noise
Qest = covest(1:n, 1:n);
% Covariance of measurement noise
Rest = covest(n+1:n+nout, n+1: n+nout);
% Cross-covariance
Sest = covest(1:n, n+1: n+nout);

% Kalman gain calculation - see documentation of dare() and compare it to
% equation on page 334 of the book to see why exaxtly A' and C' are sent to
% the function
[~, ~, K] = dare(Aest', Cest', Qest, Rest, Sest);

K = K';


%% 7. Simulation and verification

phi_sim = zeros(n,Nval);
s_sim = zeros(nout, Nval);
for i=1:Nval
    phi_sim(:,i+1) = (Aest - K*Cest)*phi_sim(:,i) + K*sk(:,Nid + i);
    s_sim(:,i) = Cest*phi_sim(:,i);
end

% Compute variance-accounted-for
vaf = computevaf(s_sim, sk(:,Nid+1:Nid+Nval));


end
```