SC42025 Filtering & Identification

Final Assignment

18 January 2019

TU Delft
Coordinated by Michel Verhaegen

Alexander Berndt 4698959 A.E.Berndt@student.tudelft.nl

Matthijs Bekendam 4725751 J.M.Bekendam@student.tudelft.nl

Contents

Introduction

Adaptive Optics

Adaptive Optics involves the clarification of a distorted wavefront traversing into a lense. This is typically seen in telescopes where the view from Earth of extraterrestrial objects such as stars and planets is distorted by inconsistencies in the atmosphere. By using sensors and a closed-loop control scheme, these distortions can be accounted for to yield a clearer view of the star or planet.

The atmospheric aberrations can be measured using a Shack-Hartmann (SH) sensor. The aberrations are caused by a distorted wavefront. The SH sensor specifically measures the slopes of the wavefront. Using filtering techniques, the wavefront can be estimated using these slope measurements and then a deformable mirror can be used to counteract the estimated wavefront to yield a clear view. Figure 1 shows a histogram of the location of the slopes of open-loop control (obtained from ϕ_{sim} dataset 1). The aim will be to ultimately reduce these distributions, which directly implies a clearer image.

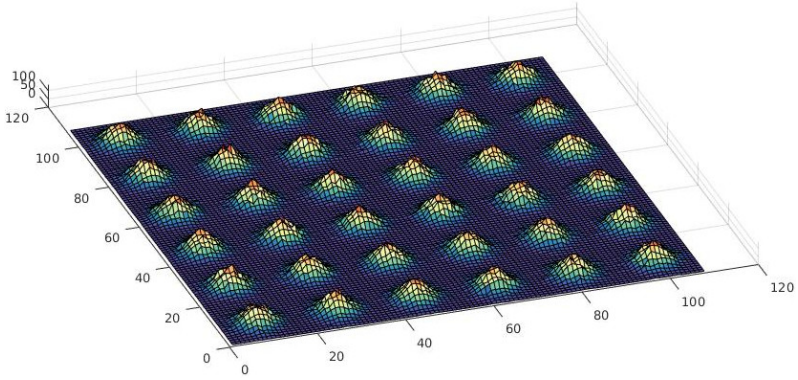

Figure 1: Slope Measurements Distribution Histogram of Open Loop Control

Using this control scheme, very powerful results can be obtained. Figure 2 shows the a planet viewed through a telescope with and without adaptive optics.

Figure 2: View of Neptune from ESO's Very Large Telescope [1]

In this assignment, we investigate different methods of estimating the incoming wavefront with which the control action is determined. These methods are then compared to each other and to the open-loop (no control) case.

1 Random Walk Model

In this section wave-front reconstruction is attempted by making use of a random walk model. The reconstruction will be based on sensor data $s(k)$ and, when available, prior statistical information.

Questions 1.1 - 1.5 focus on deriving new expressions (unbiased minimum variance estimates) and minimizing the control law. Questions 1.6 and 1.7 illustrate the quality of the model by the closed-loop variance and the Variance-Accounted-For (VAF) as described in Equation 10.22 in [2].

Question 1.1

Given measurements of the open-loop slopes $s_o(k)$ and assuming no prior information on vectorized wavefront $\phi(k)$, an estimate of $\phi(k)$ can be determined using linear least squares. The matrix G is not full column rank, meaning that the solutions are not unique. hence a general solution can be found by the use of the SVD (singular value decomposition). Let the SVD of the matrix G be given by

$$G = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}. \tag{1.1}$$

The minimization problem can be written as

$$\min_{\phi} ||G\phi(k) - s_0(k)||_2^2 = \min_{\phi} ||U_1 \Sigma V_1^T \phi(k) - s_0(k)||_2^2. \tag{1.2}$$

Introducing the partitioned vector

$$\begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \phi(k). \tag{1.3}$$

Using equation 1.3 the minimization problem can be written as

$$\min_{\phi} ||G\phi(k) - s_0(k)||_2^2 = \min_{\xi_1} ||U_1 \Sigma \xi_1 - s_0(k)||_2^2. \tag{1.4}$$

Since ξ_2 does not influence the value of $\min_\phi ||G\phi(k) - s_0(k)||_2^2$, it can be chosen arbitrary. In order to obtain the smallest 2-norm, the value ξ_2 is set to 0. The solution becomes

$$\hat{\phi}(k) = V_1 \Sigma^{-1} U_1^T s_0(k) \tag{1.5}$$

The solution cannot be unique due to the rank deficiency of G. To yield a unique solution, additional constraints on the vector ϕ are needed.

Question 1.2

Now that prior knowledge of the statistical properties of $\phi(k)$ is available, an unbiased minimum variance estimate, denoted by $\hat{\phi}(k|k)$, can be determined. The estimate will be determined using the open-loop measurements $s_0(k)$, prior wavefront information $E[\phi(k)] = 0$ and $E[\phi(k)\phi^T] = C_\Phi(0) > 0$ and the noise variance σ_e^2. The covariance matrix $C_\phi(0)$ is approximated as

$$C_\phi(0) \approx \frac{1}{N_t} \sum_{i=1}^{N_t} \phi(i)\phi(i)^T. \tag{1.6}$$

Let us define $P \doteq C_\phi(0)$ and following Theorem 4.3 of the book:

$$\tilde{\phi}(k) = \begin{bmatrix} M & N \end{bmatrix} \begin{bmatrix} s_o(k) \\ \bar{\phi}(k) \end{bmatrix} \tag{1.7}$$

Since we know a priori that $\bar{\phi} = 0$, we can write

$$\phi(k) - \tilde{\phi}(k) = (I - MG)\phi(k) - Me(k)$$

$$\begin{aligned} E[(\phi(k) - \tilde{\phi}(k))(\phi(k) - \tilde{\phi}(k))^T] &= (I - MG)E[\phi(k)\phi(k)^T](I - MG)^T + ME[e(k)\phi(k)^T](I - MG)^T + \\ &\quad (I - MG)E[\phi(k)e(k)^T]M^T + ME[e(k)e(k)^T]M^T \\ &= (I - MG)P(I - MG)^T + \sigma_e^2 MM^T \\ &= \begin{bmatrix} I & -M \end{bmatrix} \begin{bmatrix} P & PG^T \\ GP & GPG^T + \sigma_e^2 I \end{bmatrix} \begin{bmatrix} I \\ -M^T \end{bmatrix} \end{aligned}$$

As performed on pg. 114 Section 4.5.3 [2], using Lemma 2.3, we find

$$M = PG^T(GPG^T + \sigma_e^2 I)^{-1}$$

Since the covariance matrix P is positive definite, the matrix inversion lemma (Lemma 2.2 on pg. 19) could be applied for an alternative notation of M:

$$M = (P^1 + F^T W F)^{-1} F^T W$$

Substituting the former M into Equation 1.7 we obtain the following, optimal estimate of $\phi(k)$

$$\hat{\phi}(k) = PG^T(GPG^T + \sigma_e^2 I)^{-1} s_0(k) \tag{1.8}$$

recall that $P = C_\phi(0)$

Question 1.3

Considering the closed loop system, an unbiased minimum variance estimate of the residual wavefront, denoted by $\hat{\epsilon}(k|k)$, will be derived. The estimate will be determined using the measurements $s(k)$, prior wavefront information $E[\epsilon(k)] = 0$ and $E[\epsilon(k)\epsilon^T = C_\Phi(0) > 0$ and the noise variance σ_e^2. This derivation can be solved in a similar fashion as the previous question.

$$\tilde{\epsilon}(k) = \begin{bmatrix} M & N \end{bmatrix} \begin{bmatrix} s(k) \\ \bar{\epsilon}(k) \end{bmatrix} \tag{1.9}$$

Since we know a priori that $\bar{\epsilon} = 0$, we can write

$$\epsilon(k) - \tilde{\epsilon}(k) = (I - MG)\epsilon(k) - Me(k)$$

$$\begin{aligned} E[(\epsilon(k) - \tilde{\epsilon}(k))(\epsilon(k) - \tilde{\epsilon}(k))^T] &= (I - MG)E[\epsilon(k)\epsilon(k)^T](I - MG)^T + ME[e(k)\epsilon(k)^T](I - MG)^T + \\ &\quad (I - MG)E[\epsilon(k)e(k)^T]M^T + ME[e(k)e(k)^T]M^T \\ &= (I - MG)P(I - MG)^T + \sigma_e^2 MM^T \\ &= \begin{bmatrix} I & -M \end{bmatrix} \begin{bmatrix} P & PG^T \\ GP & GPG^T + \sigma_e^2 I \end{bmatrix} \begin{bmatrix} I \\ -M^T \end{bmatrix} \end{aligned}$$

As performed on pg. 114 Section 4.5.3 [2], using Lemma 2.3, we find

$$M = PG^T(GPG^T + \sigma_e^2 I)^{-1}$$

Substituting M into Equation 1.7 we obtain the following, optimal estimate of $\epsilon(k)$

$$\hat{\epsilon}(k) = PG^T(GPG^T + \sigma_e^2 I)^{-1} s_k(k) \tag{1.10}$$

Question 1.4

We would like an optimal prediction of $\epsilon(k)$ at $k+1$ at time step k, denoted $\epsilon_{k+1|k}$. We have the following relation

$$\epsilon(k+1) = \phi(k+1) - Hu(k)$$

For the optimal state estimate, we have

$$\hat{\epsilon}(k+1|k) = \hat{\phi}(k+1|k) - H\hat{u}(k)$$

where $\hat{u}(k)$ is determined as

$$\hat{u}(k) = \arg\min_{u(k)} ||\epsilon(k+1)||_2^2 = \arg\min_{u(k)} ||\hat{\phi}(k+1) - Hu(k)||_2^2$$

We therefore need an optimal value of $\phi(k+1)$ and the above equation can be solved to determine $\hat{\epsilon}(k+1)$. To solve $\hat{\phi}(k+1)$ we do

$$\begin{aligned} \phi(k+1) &= \phi(k) + \eta(k) \\ \epsilon(k) &= \phi(k) - Hu(k-1) \\ \phi(k) &= \epsilon(k) + Hu(k-1) \end{aligned}$$

The best estimate of $\phi(k+1)$ given the measurements at $\phi(k)$ is defined as

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

Using the weighted least-squares in Section 4.5.2, we obtain the simple ($y = Fx + eps$ with $F = I$)

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

Therefore, we can determine the optimal control input $u(k)$ as follows

$$\hat{u}(k) = \arg\min_{u(k)} ||\hat{\epsilon}(k|k) + Hu(k-1) - Hu(k)||_2^2$$

Finally, the optimal estimate of $\epsilon(k+1)$ at time k is defined as

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

Question 1.5

We have defined $\delta u(k) := u(k) - u(k-1)$, using the result of Question 1.4, we can write

$$\hat{\epsilon}(k+1|k) = \hat{\epsilon}(k|k) - H\delta u(k) \tag{1.11}$$

$$\min_{u(k)} ||\hat{\epsilon}(k+1|k)||_2^2 \tag{1.12}$$

We can rewrite this using $\delta u(k)$ instead of $u(k)$

$$\min_{\delta u(k)} ||\hat{\epsilon}(k|k) - H\delta u(k)||_2^2$$

This is the form of a weighted least squares problem in Section 4.5.2 where $W = C_\phi(0)$, $F = H$, $y = \hat{\epsilon}(k|k)$ which results in

$$\delta u(k) = (H^T C_\phi(0) H)^{-1} H^T C_\phi(0) \hat{\epsilon}(k|k)$$

However, since H is square and invertible, we obtain

$$\delta u(k) = (H^{-1}C_\phi(0)^{-1}(H^T)^{-1})H^T C_\phi(0)\hat{\epsilon}(k|k)$$
$$\delta u(k) = H^{-1}C_\phi(0)^{-1}C_\phi(0)\hat{\epsilon}(k|k)$$
$$\delta u(k) = H^{-1}\hat{\epsilon}(k|k)$$

Having defined $\delta u(k) = u(k) - u(k-1)$ we have

$$u(k) = H^{-1}\hat{\epsilon}(k|k) + u(k-1) \tag{1.13}$$

where

$$\hat{\epsilon}(k|k) = PG^T(GPG^T + \sigma_e^2 I)^{-1}s_k(k).$$

In short, the optimal increment $\delta u(k)$ is defined as

$$\delta u(k) = H^{-1}PG^T(GPG^T + \sigma_e^2 I)^{-1}s_k(k) \tag{1.14}$$

Question 1.6

In this subsection, the closed loop (as illustrated in Figure 1) is investigated by taking a look at the variance of the residual wavefront. The variance of the open-loop v.s the variance of the closed loop is plotted in Figure 3.

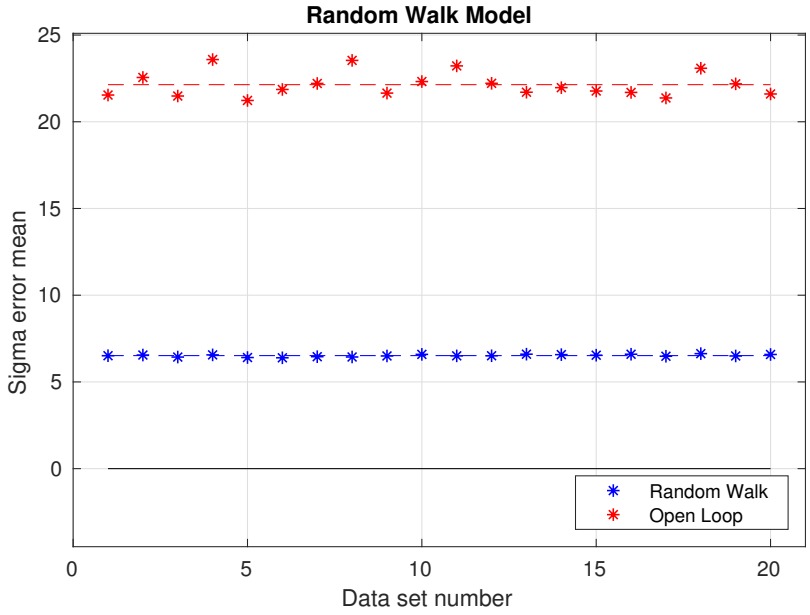

Figure 3: Comparison of Open Loop and Random Walk variance

Figure 3 clearly shows a significant improvement (the variance decreases) when the loop is closed.

Question 1.7

When constructing a new model, it is import to check its validity. To access the quality of a model, one could make use of the variance accounted for (VAF). The VAF has a value between 0% and 100%; the higher the VAF, the lower the prediction error and the better the model. The variance accounted for is determined using Equation 10.22 in the book and repeated here

$$\text{VAF}\Big(\phi(k+1|k), \hat{\phi}(k+1|k)\Big) = \max\left(0, \left(1 - \frac{\frac{1}{N}\sum_{k=1}^{N} ||\phi(k+1|k) - \hat{\phi}(k+1|k)||_2^2}{\frac{1}{N}\sum_{k=1}^{N} ||\phi(k+1|k)||_2^2}\right) \cdot 100\%\right)$$

Note that we first remove the mean. This is stated to ensure consistency. For the random walk model, the mean VAF for the 20 data-sets is 71.2%.

2 Vector Auto-Regressive Model of Order 1

The method used in section 1 ignores the spatio-temporal correlations. It is more reasonable to assume that the turbulence flows over the telescope aperture with a given pattern. The time it takes for the turbulence to cross the line of sight of the telescope is much longer than a sampling period. Therefore, in this section we attempt to reconstruct the wavefront using a Vector Auto-Regressive (VAR) model of order 1.

Questions 2.1 - 2.5 focus on deriving new expressions, computing the Kalman gain and determining the optimal control action for the improved model. Questions 2.6 and 2.7 illustrate the quality of the updated model by the closed-loop variance and the VAF.

Question 2.1

The VAR model of the wavefront evolution can be described as

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

Additionally, we define the following

$$C_\phi(0) = E[\phi(k)\phi(k)^T] \quad C_\phi(1) = E[\phi(k+1)\phi(k)^T] \tag{2.2}$$

From Equation 2.1, we can write $C_\phi(1)$ as

$$\begin{aligned} C_\phi(1) &= E[(A\phi(k) + w(k))\phi(k)^T] \\ &= E[A\phi(k)\phi(k)^T + w(k)\phi(k)^T] \\ &= A\underbrace{E[\phi(k)\phi(k)^T]}_{C_\phi(0)} + \underbrace{E[w(k)\phi(k)^T]}_{=0} \\ &= AC_\phi(0) \end{aligned} \tag{2.3}$$

Since the covariance matrices are assumed to be positive definite (and thus, nonsingular), we can write the final result

$$A = C_\phi(1)C_\phi(0)^{-1} \tag{2.4}$$

Question 2.2

The assumption is that the wavefront is a Wide-Sense Stationary (WSS) signal. From Definition 4.13 from [2], a WSS signal is defined as

- *Has constant mean* $E[\phi(k)] = \phi_m \ \ \forall\, k$
- *Auto-correlation function only depends on lag*
- *Has finite variance*

We want to define $C_w = E[w(k)w(k)^T]$, hence, let us start from

$$\begin{aligned} w(k)w(k)^T &= \big(\phi(k+1) - A\phi(k)\big)\big(\phi(k+1) - A\phi(k)\big)^T \\ &= \phi(k+1)\phi(k+1)^T - A\phi(k)\phi(k+1)^T - \phi(k+1)\phi(k)^T A^T + A\phi(k)\phi(k)^T A^T \end{aligned} \tag{2.5}$$

Since $\phi(k)$ is assumed to be WSS, $E[\phi(k+1)\phi(k+1)^T] = E[\phi(k)\phi(k)^T] = C_\phi(0)$. Additionally, from Question 1, we know that $E[\phi(k+1)\phi(k)^T] = C_\phi(1) = AC_\phi(0)$. Finally, since $E[w(k)] = 0$, $C_w = E[w(k)w(k)^T]$. We can write

$$
\begin{aligned}
C_w = E\big[w(k)w(k)^T\big] &= E\big[\phi(k+1)\phi(k+1)^T\big] - AE\big[\phi(k+1)\phi(k)^T\big]^T - \\
&\quad E\big[\phi(k+1)\phi(k)^T\big]A^T + AE\big[\phi(k)\phi(k)^T\big]A^T \\
&= C_\phi(0) - A\big(AC_\phi(0)\big)^T - AC_\phi(0)A^T + AC_\phi(0)A^T
\end{aligned} \tag{2.6}
$$

$$
C_w = C_\phi(0) - AC_\phi(0)A^T
$$

Note that we use the fact $\big(\phi(k+1)\phi(k)^T\big)^T = \phi(k)\phi(k+1)^T$ which implies that $C_\phi(0)$ and $C_\phi(1)$ are symmetric.

Question 2.3

We are given the following equations to form a state-space model

$$
\epsilon(k) = \phi(k) - \phi_{DM} = \phi(k) - Hu(k-1) \tag{2.7}
$$

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

$$
\phi(k+1) = A\phi(k) + w(k) \tag{2.9}
$$

We need the state to be $\epsilon(k)$ and the output to be $s(k)$. From Equation 2.7 we can write

$$
\begin{aligned}
\epsilon(k+1) &= \phi(k+1) - Hu(k) \\
&= A\phi(k) + w(k) - Hu(k)
\end{aligned} \tag{2.10}
$$

Rearranging Equation 2.7, we can define

$$
\phi(k) = \epsilon(k) + Hu(k-1) \tag{2.11}
$$

Substituting Equation 2.11 into Equation 2.10 we obtain

$$
\begin{aligned}
\epsilon(k+1) &= A\big(\epsilon(k) + Hu(k-1)\big) + w(k) - Hu(k) \\
&= A\epsilon(k) + AHu(k-1) + w(k) - Hu(k)
\end{aligned} \tag{2.12}
$$

From Equations 2.12 and 2.8 we can represent a state-space model as follows

$$
\begin{aligned}
\epsilon(k+1) &= A\epsilon(k) + \begin{bmatrix} -H & AH \end{bmatrix} \begin{bmatrix} u(k) \\ u(k-1) \end{bmatrix} + w(k) \\
s(k) &= G\epsilon(k) + e(k)
\end{aligned} \tag{2.13}
$$

Question 2.4

From Section 5.3 in [2], given an LTI system

$$
\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) + w(k) \\
y(k) &= Cx(k) + Du(k) + v(k)
\end{aligned} \tag{2.14}
$$

Matching the state space representation from Question 3 with Equation 2.14, we find that $A = A$, $B = \begin{bmatrix} -H & AH \end{bmatrix}$, $C = G$ and $D = 0$. The noise signals are also $w(k) = w(k)$ and $v(k) = e(k)$.

Observer form is shown on Section 5.2 of [2] and is as follows:

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + K\Big(y(k) - C\hat{x}(k) - Du(k)\Big) \tag{2.15}$$

Using our specific system, the observer form looks as follows

$$\hat{\epsilon}(k+1) = A\hat{\epsilon}(k) + \begin{bmatrix} -H & AH \end{bmatrix} \begin{bmatrix} u(k) \\ u(k-1) \end{bmatrix} + K\Big(s(k) - G\hat{\epsilon}(k)\Big) \tag{2.16}$$

Since we are considering an LTI system, the Kalman gain will converge to a constant value over time as described in Section 5.7 of [2]. The Kalman gain can thus be determined as

$$K = (S + APC^T)(CPC^T + R)^{-1} \tag{2.17}$$

However, since the signal $w(k)$ is uncorrelated with both the measurement noise $e(k)$ and the turbulent wavefront $\phi(k)$, we know that $S = 0$.

Question 2.5

We need to estimate $\hat{\epsilon}(k+1|k)$

$$E\left[\begin{bmatrix} w(k) \\ e(k) \end{bmatrix} \begin{bmatrix} w(j)^T & e(j)^T \end{bmatrix} \right] = \left[\begin{bmatrix} R & S^T \\ S & Q \end{bmatrix} \right] \Delta(k-j)$$

We are given that $E[w(k)e(k)^T] = 0$ and $E[w(k)\phi(k)^T] = 0$. This means that $S = 0$. Also, we know that $E[w(k)w(k)^T] = C_w = C_\phi(0) - AC_\phi(0)A^T$ and $E[e(k)e(k)^T] = I\sigma_e$. We can therefore state

$$E\left[\begin{bmatrix} w(k) \\ e(k) \end{bmatrix} \begin{bmatrix} w(j)^T & e(j)^T \end{bmatrix} \right] = \left[\begin{bmatrix} C_w & 0 \\ 0 & I\sigma_e \end{bmatrix} \right] \Delta(k-j)$$

The Kalman gain can then be computed by solving the Discrete Algebraic Ricatti Equation as follows

$$P = APA^T + I\sigma_e - (0 + APG^T)(GPG^T + C_w)^{-1}(0 + APG^T)^T$$

Having solved the DARE, we obtain the following Kalman gain

$$K = (APG^T)(GPG^T + C_w)^{-1}$$

We can therefore write

$$\hat{\epsilon}(k+1) = A\hat{\epsilon}(k) - Hu(k) + AHu(k-1) + (APG^T)(GPG^T + C_w)^{-1}\Big(s(k) - G\hat{\epsilon}(k)\Big) \tag{2.18}$$

which is our optimal one step ahead prediction $\hat{\epsilon}(k+1|k)$ of $\epsilon(k)$.

Question 2.6

The control law is defined as

$$\min_{u(k)} ||\hat{\epsilon}(k+1|k)||_2^2 \tag{2.19}$$

where $\hat{\epsilon}(k+1|k)$ is defined as a function of $u(k)$ (derived from the previous question).

$$\underbrace{A\hat{\epsilon}(k) + AHu(k-1) + (APG^T)(GPG^T + C_w)^{-1}\Big(s(k) - G\hat{\epsilon}(k)\Big)}_{y} = \underbrace{H}_{F}\underbrace{u(k)}_{x} + \underbrace{\hat{\epsilon}(k+1)}_{\epsilon} \tag{2.20}$$

We note that since we only want to optimize $\hat{\epsilon}(k+1)$ in terms of $u(k)$, with all the other variables defined at time-step k, we have a linear-least squares problem as defined on pg. 109.

$$\hat{u}(k) = \big(H^T H\big)^{-1} H^T \left(A\hat{\epsilon}(k) + AHu(k-1) + (APG^T)(GPG^T + C_w)^{-1}\Big(s(k) - G\hat{\epsilon}(k)\Big) \right) \tag{2.21}$$

Since H is invertible and square and, additionally, we can group terms for $\hat{\epsilon}(k|k)$ this simplifies to

$$\hat{u}(k) = H^{-1}\Big((A - KG)\hat{\epsilon}(k) + AHu(k-1) + Ks(k)\Big) \tag{2.22}$$

Note: need to use $\hat{\epsilon}(k|k-1)$ for the Kalman equation since the optimal predictor is based on the previous optimal predicted state.

Question 2.7

For this question, we need to estimate the values of $C_\phi(0)$ and $C_\phi(1)$ from the given data. This will be done using an approximation as described in the assignment Equation 8. The approximations are

$$C_\phi(0) \approx \frac{1}{N}\sum_{i=1}^{N} \phi(i)\phi(i)^T$$

$$C_\phi(1) \approx \frac{1}{N-1}\sum_{i=2}^{N} \phi(i)\phi(i-1)^T$$

These estimates of $C_\phi(0)$ and $C_\phi(1)$ we can be used with the known matrix G and σ_e to determine A, C_w and K using the function *computeKalmanAR*. Additionally, $C_\phi(0)$ and $C_\phi(1)$ were forced symmetric by taking $A_{symmetric} = \frac{1}{2}\big(A + A^T\big)$. This was done to overcome numerical inaccuracies.

Question 2.8

In this subsection, the closed loop (as illustrated in Figure 1) is investigated by taking a look at the variance of the residual wavefront. The variance of the open loop, random walk model and VAR are compared and plotted in Figure 4.

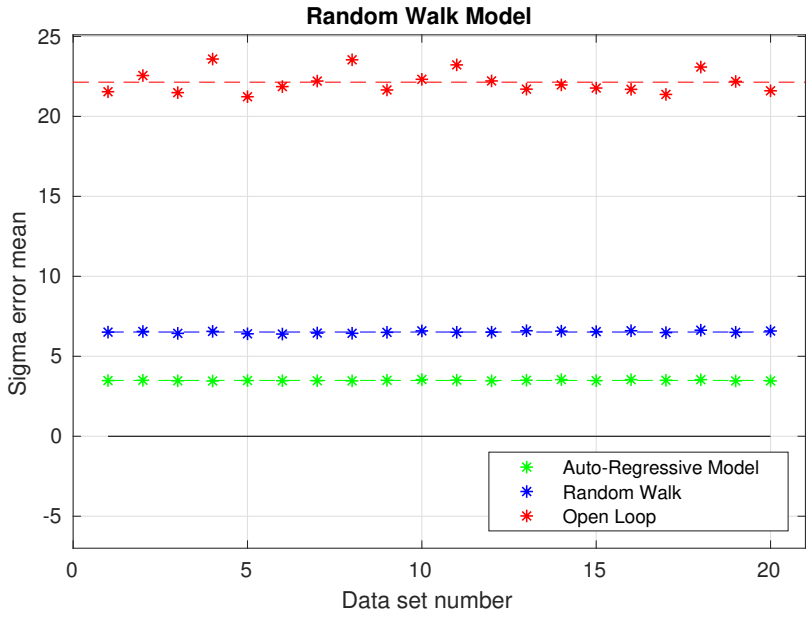

Figure 4: Comparison of Open Loop, Random Walk and Auto-Regressive Model variance

The variance accounted for for the wavefront estimate $\hat{\phi}(k+1|k)$ is tabulated as follows.

		VAF
1	Random Walk	71.2 %
2	VAR model	86.2 %

Table 1: Variance Accounted For comparison

3 Subspace Identification

Finally, we consider that the turbulence is modeled by a more general stochastic state-space model. We can formulate a total open-loop system of the form:

$$\begin{aligned}\phi(k+1) &= A_s\phi(k) + K_s v(k)\\ s(k) &= C_s\phi(k) + v(k)\end{aligned}$$

where $v(k)$ is the innovation sequence and $x(k)$ is a general state that does not necessarily equal $\epsilon(k)$.

In questions 3.1 and 3.2 it is explained how A_s,C_s and K_s can be estimated using subspace identification. Question 3.3 and 3.4 give an expression for a one step ahead predictor $\hat{\epsilon}(k+1|k)$ and the control law. Question 3.5 and 3.6 illustrate the quality of the updated model by the closed-loop variance and the VAF.

Question 3.1

Considering Chapter 9.6 in the book and following the format of equation 9.50 and 9.51, albeit without inputs, the data equation looks like

$$Y_{i,s,N} = \mathcal{O}_s X_{i,N} + S_s E_{i,s,N}$$

where we define

$$Y_{i,s,N} = \begin{bmatrix} s(i) & s(i+1) & s(i+2) & \dots & s(i+N-1)\\ s(i+1) & s(i+2) & s(i+3) & \dots & s(i+N)\\ s(i+2) & s(i+3) & s(i+4) & \dots & s(i+N+1)\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ s(i+s-1) & s(i+s) & s(i+s+1) & \dots & s(i+N+s-2)\end{bmatrix}$$

$$\mathcal{O}_s = \begin{bmatrix} C\\ CA\\ CA^2\\ \vdots\\ CA^{s-1}\end{bmatrix} \qquad X_{i,N} = \begin{bmatrix} x(i) & x(i+1) & \dots & x(N-1)\end{bmatrix}$$

$$S_s = \begin{bmatrix} I_l & 0 & 0 & \dots & 0\\ CK & I_l & 0 & \dots & 0\\ CAK & CK & I_l & \dots & 0\\ \vdots & & \ddots & \ddots & \\ CA^{s-2}K & CA^{s-3}K & \dots & CK & I_l\end{bmatrix} \qquad E_{i,s,N} = \begin{bmatrix} v(i) & v(i+1) & \dots & v(i+N-1)\\ v(i+1) & v(i+2) & \dots & v(i+N)\\ v(i+2) & v(i+3) & \dots & v(i+N+1)\\ \vdots & \vdots & \ddots & \vdots\\ v(i+s-1) & v(i+s) & \dots & v(i+N+s-2)\end{bmatrix}$$

Since we have no inputs to the system, the $U_{i,s,N}$ matrix is 0. This means that we do not need to determine a projection matrix $\Pi_{U_{0,s,N}}$ to cancel it out. Due to the noise $E_{i,s,N}$, however, we need to look for an instrumental variable Z_N to eliminate its affect on the relation between $Y_{i,s,N}$ and $\mathcal{O}_s X_{i,N}$. This is explained mathematically

Using the instrumental-variable approach, we look for a matrix Z_N that has the following properties (similar to Equations 9.53 and 9.54 in the book)

$$\lim_{N\to\infty} \frac{1}{N} E_{i,s,N} Z_N^T = 0$$
$$\operatorname{rank}\left(\lim_{N\to\infty} \frac{1}{N} X_{i,N} Z_N^T \right) = n$$

As suggested on pg. 323, let us try using past outputs for Z_N^T. Therefore, without any input, Z_N^T will be denoted as

$$Z_N^T = Y_{0,s,N}$$

This Z_N is based on the fact that the past outputs and the future noise inputs are uncorrelated (based on the ZMWN noise assumption and the system causality) as described in Section 9.6.1 of the book.

The N4SID subspace method will be applied to calculate the column space of $\mathcal{O}_s$ and a least-squares approach will used to identify A_s,C_s. The first step is to perform an RQ factorization

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

As proposed in Van Overschee en De Moor (1994), the following SVD

$$R_{21} R_{11}^{-1} Z_N = U_n \Sigma_n V_n^T$$

can be used to approximate the column space of $\mathcal{O}_s$ by the matrix U_n and to approximate the row space of the state sequence of a Kalman filter by

$$\hat{X}_{s,N} = \Sigma_n^{\frac{1}{2}} V_n^T$$

The system matrices A_s and C_s can now be estimated by solving the least-squares problem:

$$\min_{A_s,C_s} \left|\left| \begin{bmatrix} \hat{X}_{s+1,N} \\ Y_{s,1,N-1} \end{bmatrix} - \begin{bmatrix} A_s \\ C_s \end{bmatrix} \hat{X}_{s,N-1} \right|\right|_F^2 \tag{3.1}$$

Performing a transpose on Equation 3.1 will put the least-squares problem into implementable form and will not effect the outcome.

$$\min_{A_s^T,C_s^T} \left|\left| \begin{bmatrix} \hat{X}_{s+1,N}^T & Y_{s,1,N-1}^T \end{bmatrix} - \hat{X}_{s,N-1}^T \begin{bmatrix} A_s^T & C_s^T \end{bmatrix} \right|\right|_F^2$$

Question 3.2

The estimated state sequence $\hat{X}_{s,N}$ and the estimated system matrices $\hat{A}_T$,$\hat{C}_T$, can be related to the least-squares residuals as

$$\begin{bmatrix} \hat{W}_{s,1,N-1} \\ \hat{V}_{s,1,N-1} \end{bmatrix} = \begin{bmatrix} \hat{X}_{s+1,N} \\ Y_{s,1,N-1} \end{bmatrix} - \begin{bmatrix} \hat{A}_T \\ \hat{C}_T \end{bmatrix} \hat{X}_{s,N-1} \tag{3.2}$$

These residuals can be used to estimate the co variance matrix as follows:

$$\begin{bmatrix} \hat{Q} & \hat{S} \\ \hat{S}^T & \hat{R} \end{bmatrix} = \lim_{N\to\infty} \frac{1}{N} \begin{bmatrix} \hat{W}_{s,1,N} \\ \hat{V}_{s,1,N} \end{bmatrix} \begin{bmatrix} \hat{W}_{s,1,N}^T & \hat{V}_{s,1,N}^T. \end{bmatrix} \tag{3.3}$$

The solution P of the following Ricatti equation:

$$\hat{P} = \hat{A}_T \hat{P} \hat{A}_T^T + \hat{Q} - (\hat{S} + \hat{A}_T \hat{P} \hat{C_T^T})(\hat{C}_T \hat{P} \hat{C}_T^T + \hat{R})^{-1}(\hat{S} + \hat{A}_T \hat{P} \hat{C_T^T})^T \tag{3.4}$$

can be used to obtain the estimate of the Kalman gain:

$$\hat{K}_T = (\hat{S} + \hat{A}_T \hat{P} \hat{C_T^T})(\hat{C}_T \hat{P} \hat{C}_T^T + \hat{R})^{-1}. \tag{3.5}$$

Question 3.3

We have an identified model as determined in Question 3.1 and 3.2 denoted by

$$x(k+1) = A_s x(k) + K_s v(k)$$
$$s(k) = C_s x(k) + v(k)$$

An observer can be defined of the form

$$\hat{s}(k+1|k) = C_s \hat{x}(k+1|k)$$

where we obtain $\hat{x}(k+1|k)$ using the predictor model as defined in Equation 5.78 as

$$\hat{x}(k+1|k) = (A_s - K_s C_s)\hat{x}(k|k-1) + K_s s(k)$$

and K_s is the Kalman gain estimation up until a similarity transformation as determined in Question 3.2. Using this estimate of the predicted state $\hat{x}(k+1k)$ we can obtain an optimal estimate of the future measurement $\hat{x}(k+1|k)$, and from this, an optimal estimate of $\hat{\phi}(k+1|k)$ as follows

$$\hat{s}(k+1|k) = C_s \hat{x}(k+1|k)$$

$$\hat{\phi}(k+1|k) = V_1 \Sigma^{-1} U_1^T \hat{s}(k+1|k)$$

Using this optimal estimate of $\hat{\phi}(k+1|k)$, we can define the residual using Equation 6 and 7 of the assignment as

$$\min_{u(k)} ||\hat{\epsilon}(k+1|k)||_2^2$$

where

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

Question 3.4

Rewriting the equations of 3.3, we can define the optimal control input $u(k)$ at time step k as

$$\hat{u}(k) = \arg\min_{u(k)} ||\hat{\phi}(k+1|k) - Hu(k)||_2^2$$

We note that since H is invertible and square, $\hat{u}(k)$ is simply defined as

$$\hat{u}(k) = H^{-1}\hat{\phi}(k+1|k)$$

We can express $\hat{\phi}(k+1|k)$ using the equations defined throughout Question 3.3. This results in the following

$$\hat{u}(k) = H^{-1}\Big(V_1 \Sigma^{-1} U_1^T C_s \Big((A_s - K_s C_s)\hat{x}(k|k-1) + K_s s(k)\Big)\Big)$$

Note that $s(k)$ is the open-loop measurement of the slope since the

Question 3.5

In this subsection, a MATLAB routine was written implementing the N4SID subspace method discussed in 3.1 and calculating the corresponding Kalman gain discussed in 3.2. With a given N_{val} it was possible to validate the subspace routine.

Furthermore, the closed loop (as illustrated in Figure 1) is investigated by taking a look at the variance of the residual wavefront. The variance of the open loop, random walk model, VAR and N4SID are compared and plotted in Figure 5.

Question 3.6

In order to determine the model order we use to model the wavefront, the singular values of Σ as well as the VAF of different model orders was analyzed. This is as discussed in Chapters 9 and 10 in the textbook, where the complication of determining the model order using singular values for systems with process and measurement noise need to be identified is discussed.

We noticed that the VAF did not increase significantly with an increase in order after $n \approx 150$. Hence, the model order was chosen to be $n = 150$ and a $s = 10$ corresponding to $2sp^2 = 720 > n$.

The mean variance of $\epsilon(k+1)$ of 1.879.

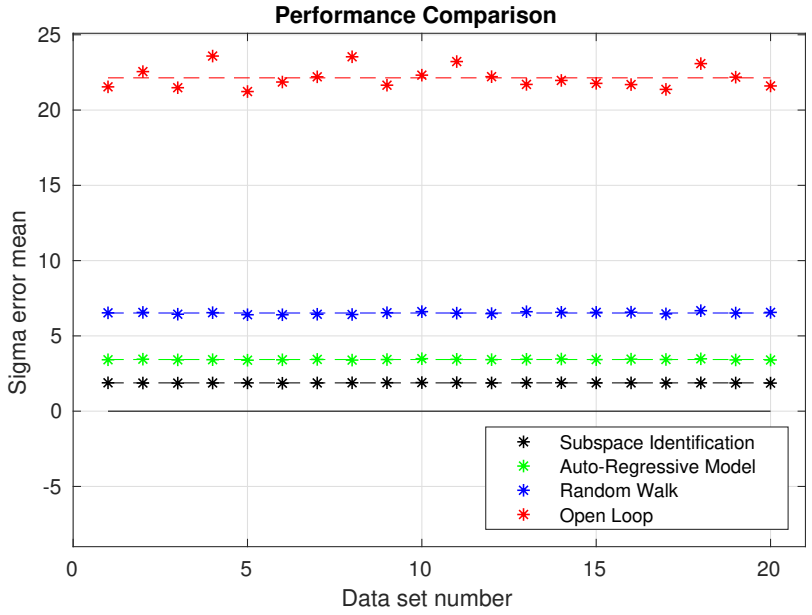

Figure 5: Comparison of OL, RW, AR and Subspace Identification Model variance

		VAF
1	Random Walk	71.2 %
2	VAR model	86.2 %
2	SubId model	92.4 %

Table 2: Variance Accounted For comparison

4 Critical Thinking

Finally, we will look back at the methods we have used and make some critical remarks on their performance, scalability and physical interpretation.

Question 4.1

We want to write this equation as a residual of the wavefront slopes $s(k)$. The residual is defined as

$$\begin{aligned} s(k) = & \underbrace{s_m(k)}_{\text{measured slope}} - \underbrace{s_{\text{DM}}(k)}_{\text{slopes induced by DM}} \\ & = s_m(k) - GHu(k-1) - Ge(k) \end{aligned}$$

The objective becomes to minimize the slope residual. This is written mathematically as

$$\min_{u(k)} ||s(k+1|k)||_2^2 = \min_{u(k)} ||\hat{s}_m(k+1|k) - GHu(k)||_2^2$$

To solve this, we need an optimal estimate of the predicted slopes at $k+1$, denoted $\hat{s}_m(k+1|k)$. Assuming a random walk model, we can determine this as

$$\hat{s}(k+1|k) = \hat{s}(k|k) + v(k)$$

The unbiased minimum variance estimate of $\hat{s}(k+1|k)$ given the random walk model is $s(k)$. Hence, we obtain the following optimization problem

$$\min_{u(k)} ||s(k) - GHu(k)||_2^2$$

We then determine $u(k)$ by solving this linear least squares problem (recalling that since G is not full column rank, GH is also not full column rank). Defining $F = GH$

$$\delta u(k) = U_1^T \Sigma^{-1} V_1 s(k)$$

Where U_1, Σ, V_1 are computed from the singular value decomposition of F as described on pg 33 in the book. Figure 6 shows the comparison between only estimating the slopes (Question 4.1) and estimating the wavefront using a random walk model (Question 1). We obtained a variance of approximately $var_{eps} = 7.45$.

Question 4.2

The results of the three methods of Questions 1 through 3, as well as the results of Question 4.1 are illustrated in Figure 7. We note that the methods, in order of performance, are ranked as follows:

1. SubId method
2. VAR model of order 1
3. Random Walk model of wavefront
4. Random Walk model of slopes
5. No control

This is to be expected since each method incrementally decreases the accuracy of the model of the wavefront. The random walk models assume no a-priori information about the wavefront (or slopes) meaning a less accurate wavefront estimation. Next, the Kalman filter assumes a VAR model of order 1, which incorporates the estimated a-priori information $C_\phi(0)$, $C_\phi(1)$ which are obtained from real wavefront data. Finally, the subspace identification method involves an identification method for the open-loop data, which yielded a even

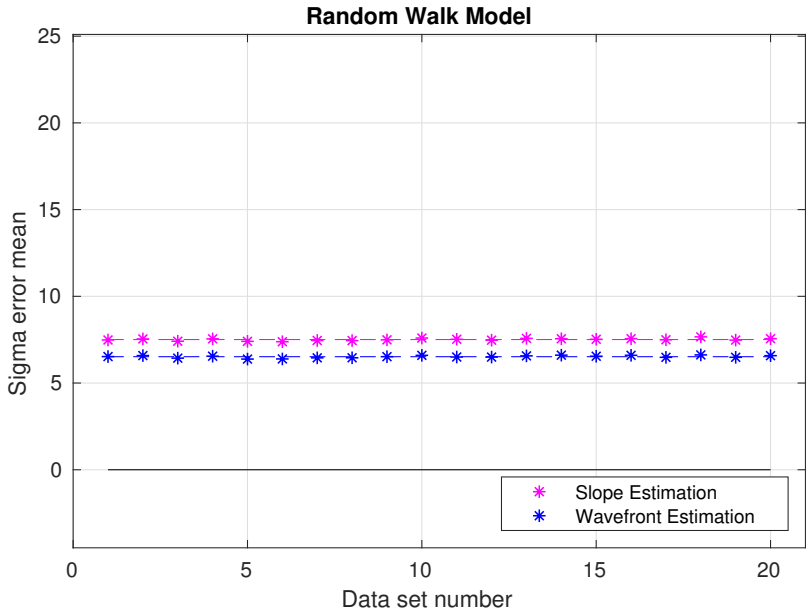

Figure 6: Comparison of Wavefront and Slopes Reconstruction using Random Walk

more accurate prediction of the wavefront.

Note that the more information we have about the wavefront, the more accurate the estimate of a the wavefront at the next time-step.

This *modelaccuracy* is not only represented by the variance in the predicted wavefront residual $\epsilon(k+1|k)$, but also in the variance-accounted-for of the wavefront model determined in each section by computing the $VAF(\phi(k+1), \hat{\phi}(k+1|k))$

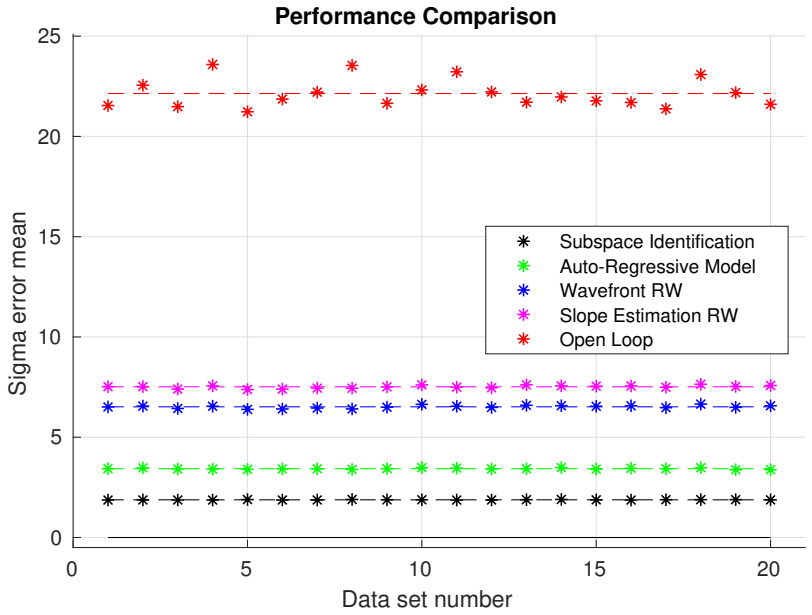

Figure 7: Comparison of Open Loop, RW, VAR, SubID and Slope Residual Control Schemes

Question 4.3

The unobservable modes of the Shack-Hartmann sensor. Since G relates the slopes $s(k)$ to the wavefront $\phi(k)$, the unobservable modes lie within the nullspace of the matrix G. We use the SVD of G to determine the vectors spanning this nullspace as follows

$$\text{svd}(G) = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$

where the null-space of G is represented by the range space of V_2.

$$\ker(G) = \text{range}(V_2)$$

We know that one of the unobservable modes corresponds to an offset in the wavefront measurement (since integrating the slopes gives us no information of the offset of the wavefront). This mode is represented by

$$\bar{v}_{\text{mode 1}} = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \end{bmatrix}^T$$

Assuming that these two modes act independently of one another, the mode perpendicular to $\bar{v}_{\text{mode 1}}$ will represent this second mode. We find the perpendicular vector $\bar{v}_{\text{mode 2}}$ to $\bar{v}_{\text{mode 1}}$ which spans the null-space which is denoted by

$$\bar{v}_{\text{mode 2}} = \begin{bmatrix} 1 & -1 & 1 & -1 & \dots & 1 \end{bmatrix}^T$$

Modes 1 and 2 are reshaped to 7×7 and shown in Figure 8. The second mode is known in the adaptive optics industry as a $waffle$-mode. One way of mitigating this mode is by momentarily opening the closed loop controlled system and resetting the deformable mirror (DM) inputs to 0 periodically.

As discussed in [1], one way of mitigating the affects of this waffle-mode in closed-loop could be by using a constrained Receding Horizon Controller, however the details thereof are deemed out of the scope of this question.

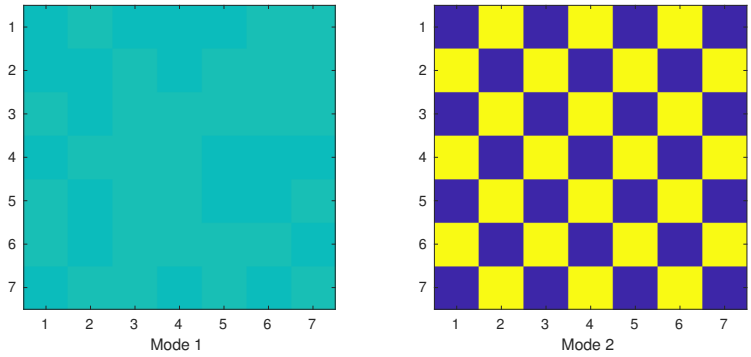

Figure 8: Visual Representation of Unobservable Modes - Mode 1: Offset, Mode 2: Waffle-mode

Question 4.4

The relation between Question 1.2, we note that

$$\hat{\phi}(k) = PG^T(GPG^T + \sigma_e^2 I)^{-1} s_0(k) \tag{4.1}$$

This was determined using a weighted least squares application as described in [2]. To minimize the variance, the following equation was formulated

$$(\phi - \tilde{\phi})(\phi - \tilde{\phi})^T = \begin{bmatrix} I & -M \end{bmatrix} \begin{bmatrix} P & PG^T \\ GP & GPG^T + \sigma_e^2 I \end{bmatrix} \begin{bmatrix} I \\ -M^T \end{bmatrix}$$

where the objective was to determine M such that $E\big[(\phi-\tilde{\phi})(\phi-\tilde{\phi})^T\big]$. We can rewrite this as

$$(\phi-\tilde{\phi})(\phi-\tilde{\phi})^T = \begin{bmatrix} I & -M \end{bmatrix} \underbrace{\begin{bmatrix} P & PG^T \\ GP & GPG^T \end{bmatrix} \begin{bmatrix} I \\ -M^T \end{bmatrix}}_{A(M)} + \sigma_e^2 MM^T$$

The minimization problem can be written as

$$\min_M \Big(A(M) + \sigma_e^2 MM^T \Big)$$

Recalling that $\phi = Ms(k)$, we can rewrite this in terms of $\phi(k)$ as shown in Equation 14 of the assignment. We can isolate M by taking an outer-product inverse of $s(k)$ as follows

$$M = \phi(k)s(k)^T \Big(s(k)s(k)^T \Big)^{-1}$$

Defining $\phi\tilde{(k)} = \phi(k)s(k)^T \Big(s(k)s(k)^T \Big)^{-1}$, the result is a minimization problem of the form

$$\min_{\tilde{\phi}(k)} \Big(A\big(\tilde{\phi}(k)\big) + \sigma_e^2 \tilde{\phi}(k)\tilde{\phi}(k)^T \Big)$$

Since $s(k)$ is a constant at time k and recalling the definition of $\tilde{\phi}(k)$, we can define $A\big(\tilde{\phi}(k)\big) = \tilde{A}\big(\phi(k)\big)$ and $\tilde{B}\big(\phi(k)\big) = \tilde{\phi}(k)\tilde{\phi}(k)^T$ which results in a minimization as described in Equation 14.

Influence of σ_e

From the formulation above, we note that σ_e can be a form of cost function weight (analogous to measurement noise covariance matrix R in the Kalman filter or the input weighting matrix of the LQ regulator [2]).

	σ_e	**Result**
1	Large	The estimate of $\phi(k)$ is more weighted on the process model, influenced by $C_\phi(0)$
2	Small	The estimate of $\phi(k)$ is more weighted on the measuremt, influenced by $s(k)$

Table 3: Influence of σ_e

Question 4.5

To analyze the scalability of the control schemes, we consider the in-the-loop calculations required to determine the control action. The sizes are shown below (for the Fried geometry).

		Dimensions
1	G	$2p^2 \times (p+1)^2$
2	H	$(p+1)^2 \times (p+1)^2$
3	$\phi(k)$	$(p+1)^2 \times 1$
4	s(k)	$2p^2 \times 1$

All the methods use a

- wavefront/slope prediction step

$$x(k+1) = Ax(k)$$

- control action determination step

$$u(k) = u(k-1) + Ms(k)$$

[2] Recognizing how the Kalman Filter and Linear Quadratic regulator are duals of each other

We have a control action determined by

$$u(k) = Ms(k) + u(k-1)$$

where M and N are general matrixes representing the optimal output for timestep k. The matrix M and N does not depend on the measurements and can be determined and its calculation is not considered here. However, the multiplication between M and $s(k)$ requires $f(p^2)$ multiplications and $g(p^2)$ summations depending on the matrix sparsity. However, assuming a worst case scenario, we note that these calculations are quadratic.

Similarly, the $u(k-1)$ and $Ms(k)$ need to be added which requires operations of the order p^2. Hence, the online calculations are quadratic in nature ($\mathcal{O}(n^2)$). This means that this system is not linearly scalable.

MATLAB Code

4.1 Question 1

4.1.1 AOloopRW

```
function [var_eps] = AOloopRW(G,H,C_phi_0,sigmae,phik)
    % Question 2 online AO simulation for closedloop measurements
    % IN
    % phik : incoming turbulence wavefront
    % sigmae: measurement noise parameter
    % H    : influence matrix mapping the wavefront on the mirror
    % G    : measurement matrix
    % OUT
    % sigma : mean variance of the residual wavefront

    n = size(H,1);        % n = 49 dimension lifted wavefront
    ns = size(G,1);       % ns = 72 dimension lifted sensor slopes
    T = length(phik);     % T = 5000 number of temporal phase points

    epsk = zeros(n,T);    % residual wavefront
    eps_piston_removed = zeros(n,T); % residual wavefront with mean removed (to avoid variance bias
        in results)
    sk = zeros(ns,T);     % ns = 72 slopes measurements
    u = zeros(n,T);       % n = 49 inputs
    sigma = zeros(T,1);

    vaf_top = zeros(n,T);
    vaf_bot = zeros(n,T);

    yk = zeros(n,T);
    ykhat = zeros(n,T);

    P = C_phi_0;
    M = inv(H)*(P*G')*inv(G*P*G' + eye(ns,ns)*sigmae.^2);

    % to determine phi(k+1|k)
    M_eps = (P*G')*inv(G*P*G' + eye(ns,ns)*sigmae.^2);

    for k = 1:T-1
        % NOTE: at time k+1 currently

        % apply previous step control action
        epsk(:,k+1) = phik(:,k+1)-H*u(:,k);

        % simulate the sensor readings at time k+1
        sk(:,k+1) = G*epsk(:,k+1) + sigmae*randn(ns,1);

        %% NOW WE NEED TO DETERMINE THE OPTIMAL CONTROL INPUT

        % update next control input:
        delta_u = M*sk(:,k+1);
        u(:,k+1) = delta_u + u(:,k);

        % determine the estimate of phi(k+1|k) - question 1.4
        phi_k_plus_1_estimate = M_eps*sk(:,k+1) + H*u(:,k);
        phi_k_plus_1 = phik(:,k+1);

        ykhat(:,k) = phi_k_plus_1_estimate;
```

```
        yk(:,k) = phi_k_plus_1;

        % for the VAF calculation
        vaf_top(:,k) = norm(phi_k_plus_1 - phi_k_plus_1_estimate,2);
        vaf_bot(:,k) = norm(phi_k_plus_1,2);

        % performance logging
        eps_piston_removed(:,k+1) = epsk(:,k+1)-mean(epsk(:,k+1));
        sigma(k+1) = var(eps_piston_removed(:,k+1));
    end

%     VAF = 1-sum(vaf_top)/sum(vaf_bot);
%     fprintf('VAF = %g \n',VAF);

    y = yk;
    yhat = ykhat;

    var_eps = mean(sigma);
end
```

4.2 Question 2

4.2.1 computeKalmanAR

```
function [A, Cw, K] = computeKalmanAR(C_phi_0, C_phi_1, G, sigma_e)
    %% Compute Kalman Auto-Regressive
    % Inputs:
    %   - C_phi_0: variance of phi(k)
    %   - C_phi_1: E[phi(k+1)phi(k)^T]
    %   - G: measurement matrix
    %   - sigma_e: noise variance
    % Outputs:
    %   -

    %% Start

    % Q2.1 compute A
    A = C_phi_1/C_phi_0;

    % Q2.2 estimate C_w
    Cw = C_phi_0 - A*C_phi_0*A';

    % force symmetry
    Cw = 0.5*(Cw+Cw');

    R = Cw;
    Q = eye(size(G,1))*(sigma_e.^2);

    % solve DARE and compute Kalman gain K
    [~,~,K] = dare(A',G',R,Q);
    K = K';
end
```

4.2.2 AOloopAR

```
function [var_eps] = AOloopAR(G,H,C_phi_0,sigmae,A,Cw,K,phik)
    % Question 2 online AO simulation for closedloop measurements
    % IN
    % phik : incoming turbulence wavefront
    % sigmae: measurement noise parameter
    % H    : influence matrix mapping the wavefront on the mirror
    % G    : measurement matrix
    % OUT
    % sigma : mean variance of the residual wavefront

    n = size(H,1);        % n = 49 dimension lifted wavefront
    ns = size(G,1);       % ns = 72 dimension lifted sensor slopes
    T = length(phik);     % T = 5000 number of temporal phase points

    epsk = zeros(n,T);    % residual wavefront
    eps_piston_removed = zeros(n,T); % residual wavefront with mean removed (to avoid variance bias
        in results)
    sk = zeros(ns,T);     % ns = 72 slopes measurements
    u = zeros(n,T);       % n = 49 inputs
    sigma = zeros(T,1);

    % eps_k_km1 = zeros(n,T); % prediction residual

    yk = zeros(n,T);
    ykhat = zeros(n,T);

    vaf_top = zeros(n,T);
    vaf_bot = zeros(n,T);
    P = C_phi_0;
    M = inv(H)*(P*G')*inv(G*P*G' + eye(ns,ns)*sigmae.^2);

    % to determine phi(k+1|k)
    M_eps = (P*G')*inv(G*P*G' + eye(ns,ns)*sigmae.^2);

    for k = 1:T-1
        % NOTE: at time k+1 currently

        %% PREPARE DATA FOR STEP k+1

        % apply previous step control action
        epsk(:,k+1) = phik(:,k+1)-H*u(:,k);

        % simulate the sensor readings at time k+1
        sk(:,k+1) = G*epsk(:,k+1) + sigmae*randn(ns,1);

        %% NOW WE NEED TO DETERMINE THE OPTIMAL CONTROL INPUT

        % get best estimate of epsilon(k|k)
        epsk_est = M_eps*sk(:,k+1);

        % update next control action:
        u(:,k+1) = inv(H)*((A-K*G)*epsk_est + A*H*u(:,k) + K*sk(:,k+1) );

        % determine the estimate of phi(k+1|k) - question 1.4
        ykhat(:,k) = epsk_est + H*u(:,k);
        yk(:,k) = phik(:,k+1);

        %% performance logging
        eps_piston_removed(:,k+1) = epsk(:,k+1)-mean(epsk(:,k+1));
```

```
        sigma(k+1) = var(eps_piston_removed(:,k+1));
    end

    y = yk;
    yhat = ykhat;

    eps = eps_piston_removed;

    var_eps = mean(sigma);
end
```

4.3 Question 3

4.3.1 SubId

```
function [As,Cs,Ks] = SubId(s_id,N_id,N_val,s,n)
    %% SubId Function
    % IN
    % s_id:    slope measurements
    % N_id:    number of points for identification
    % N_val:   number of points for validation
    % s:       upper bound
    % n:       system order
    % OUT
    % As:      A matrix of identified system
    % Cs:      C matrix of identified system
    % Ks:      Kalman gain of identified system

    %% STATE SPACE MATRICES

    disp('N4SID Linear Least Squares');

    l = size(s_id,1); % = 72
    h = l*s;

    disp('-> building hankel matrices');
    Y_0_s_N = zeros(l*s,N_id);
    Y_s_s_N = zeros(l*s,N_id);
    Y_s_1_Nm1 = zeros(l,N_id);
    for i = 1:(N_id)
        j = (i-1) + s;
        Y_0_s_N(:,i) = reshape(s_id(:,i:(i+(s-1))),[s*l,1]);
        Y_s_s_N(:,i) = reshape(s_id(:,j:(j+(s-1))),[s*l,1]);
        Y_s_1_Nm1(:,i) = reshape(s_id(:,j),[l,1]);
    end
    disp('  done');

    disp('-> RQ factorization');
    [R,~] = getRQ_factorization([Y_0_s_N; Y_s_s_N]);
    R11 = R(1:h,1:h);
    R21 = R(h+1:2*h,1:h);
    disp('  done');

    % SVD
    disp('-> determine SVD');
    [~,Sigma,V] = svd((R21/R11)*Y_0_s_N);
    Sigma_n = Sigma(1:l*n,1:l*n);
```

```
    V_n = V(:,1:l*n);
    disp('  done');

    % Xhat
    disp('-> determine Xhat');
    Xhat = sqrtm(Sigma_n)*V_n';
    disp('  done');

    Xhat_sp1_N = Xhat(:,2:N_id); % Xhat s+1,N
    Xhat_s_Nm1 = Xhat(:,1:N_id-1); % Xhat s,N-1

    % Solve LLS
    disp('-> solve LLS');
    Y_s_1_Nm1 = Y_s_1_Nm1(:,1:end-1);

    y = [Xhat_sp1_N' Y_s_1_Nm1'];
    F = Xhat_s_Nm1';

    xopt = pinv(F)*y;

    disp('  done');
    As = xopt(1:n*l,1:n*l)';
    Cs = xopt(1:n*l,n*l+1:(n+1)*l)';

    %% KALMAN GAIN ESTIMATION

    disp('N4SID Kalman Gain Determination');

    What_s_1_Nm1 = Xhat_sp1_N - As*Xhat_s_Nm1;
    Vhat_s_1_Nm1 = Y_s_1_Nm1 - Cs*Xhat_s_Nm1;

    CovMat = (1/N_id)*[What_s_1_Nm1; Vhat_s_1_Nm1]*[What_s_1_Nm1' Vhat_s_1_Nm1'];

    Qhat = CovMat(1:n*l,1:n*l);
    Shat = CovMat(1:n*l,n*l+1:(n+1)*l);
    Rhat = CovMat(n*l+1:(n+1)*l,n*l+1:(n+1)*l);

    % dare(A,B,Q,R,S,E)
    disp('-> solving DARE');
    [~,~,Ks] = dare(As',Cs',Qhat,Rhat,Shat);
    Ks = Ks';
    disp('  done!');

    %% VALIDATION

    disp('Validation');

    x = zeros((n*l),N_val);
    shat = zeros(l,N_val);

    for k = 1:N_val
       x(:,k+1) = (As-Ks*Cs)*x(:,k) + Ks*s_id(:,k);
       shat(:,k) = Cs*x(:,k);
    end
    disp('  done');

%    figure(1);
%    clf;
%    plot(shat(1,:));
```

```
%     hold on
%     plot(s_id(1,:)+0.01);

%     VAF = 1-var(norm(shat-s_id,2))/var(norm(shat,2));
%     disp('VAF of validation');
%     disp(VAF);

end
```

4.3.2 AOloopSID

```
function [var_eps] = AOloopSID(G,H,As,Cs,Ks,sigmae,phik)
    % Question 3 online AO simulation for closedloop measurements
    % IN
    % phik : incoming turbulence wavefront
    % sigmae: measurement noise parameter
    % H     : influence matrix mapping the wavefront on the mirror
    % G     : measurement matrix
    % OUT
    % sigma : mean variance of the residual wavefront

    n = size(H,1);        % n = 49 dimension lifted wavefront
    ns = size(G,1);       % ns = 72 dimension lifted sensor slopes
    T = length(phik);     % T = 5000 number of temporal phase points

    epsk = zeros(n,T);    % residual wavefront
    eps_piston_removed = zeros(n,T); % residual wavefront with mean removed (to avoid variance bias
        in results)
    sk = zeros(ns,T);     % ns = 72 slopes measurements
    u = zeros(n,T);       % n = 49 inputs
    xhat = zeros(size(As,1),T+1);
    sigma = zeros(1,T);

    eps_k_kp1_est = zeros(n,T);
    phi_k_kp1_est = zeros(n,T);
    phi_k_kp1 = zeros(n,T);

    % determine SVD based optimal estimate
    [U,S,V] = svd(G);
    n = 47;

    % decompose as defined on pg 34
    Sigma = S(1:n,1:n);
    U1 = U(:,1:n);
    U2 = U(:,n+1:end);

    V_transpose = V';

    V1 = V_transpose(1:n,:)';
    V2 = V_transpose(n+1:end,:)';

    M2 = H\(V1/Sigma)*U1'*Cs;

    for k = 2:T
        %% PREPARE DATA FOR STEP k

        % apply previous step control action determined at k-1
```

```
        epsk(:,k) = phik(:,k) - H*u(:,k-1);

        % simulate the sensor readings at time k
        sk(:,k) = G*phik(:,k) + sigmae*randn(ns,1);

        %% DETERMINE THE OPTIMAL CONTROL INPUT

        % predict x(k+1|k) using measurement s(k)
        xhat(:,k+1) = (As-Ks*Cs)*xhat(:,k) + Ks*( sk(:,k) ); %+ G*H*u(:,k-1)

        % determine optimal u(k) given x(k+1|k) (from s(k))
        uhat = M2*xhat(:,k+1);

        % update next control action:
        u(:,k) = uhat;

        %% VAF TEST
        eps_k_kp1_est(:,k) = V1/Sigma*U1'*Cs*xhat(:,k+1) - H*u(:,k);
        phi_k_kp1_est(:,k) = V1/Sigma*U1'*Cs*xhat(:,k) ;
        phi_k_kp1(:,k) = phik(:,k);

        %% PREFORMANCE ACCURACY
        eps_piston_removed(:,k) = epsk(:,k)-mean(epsk(:,k));
        sigma(k) = var(eps_piston_removed(:,k));
    end
    var_eps = sigma;
end
```

4.3.3 getRQ$_{factorization}$

```
function [R,Q] = getRQ_factorization(A)
%%    Function that determines the RQ factorization of A
%     A: n x m matrix where m > n
%     R:
%

%     [n,~] = size(A);
%     I = eye(n);
%     I_reverse = fliplr(I);
%     [Q_, R_] = qr((I_reverse*A)');
%     R = I_reverse*R_'*I_reverse;
%     Q = I_reverse*Q_';

    [Qq, Rq] = qr(A');
    R = Rq';
    Q = Qq';
end
```

4.4 Main Runfile and Question 4

4.4.1 Main Run File

```
%% SC42015 Filtering and Identification
%
% final assignment - Adaptive Optics
```

```
%
% Alexander Berndt
% Matthijs Bekendam

%% OPEN-LOOP CASE (NO CONTROL)

%% QUESTION 1 - RANDOM WALK MODEL

clc
clear

load('turbulenceData.mat');
load('systemMatrices.mat');

Nr = 20;
var_eps_RW = zeros(1,Nr);
vaf_RW = zeros(1,Nr);

% loop through each dataset of 5000 datapoints each
for i = 1:Nr

   phisim = phiSim{i};

   % estimate C_phi_0 from this data
   N = size(phisim,2);
   C_phi_0 = (1/N)*(phisim*phisim');

   % get the var_eps value for dataset i
   [var_eps_RW(i),y,yhat] = AOloopRW(G,H,C_phi_0,sigmae,phisim);

   vaf_c = calculateVAF(y,yhat);
   fprintf('%g VAF %g \n',i,vaf_c);
   vaf_RW(i) = vaf_c;
end

% determine mean var_eps
var_mean_RW = mean(var_eps_RW);
vaf_mean_RW = mean(vaf_RW);

%% QUESTION 2 - VECTOR-AUTOREGRESSIVE MODEL

clc
clear

load('turbulenceData.mat');
load('systemMatrices.mat');

Nr = 20;
var_eps_KF = inf(1,Nr);
vaf_KF = inf(1,Nr);

% loop through each dataset of 5000 datapoints each
for i = 1:Nr

    %% GET DATA FROM IDENTIFICATION DATA
    phik = phiSim{i};

    %% ESTIMATE C_phi_0 and C_phi_1 using equation 8 in assignment
    N = size(phik,2); % number of datapoints
```

```
63
64      phi_k = phik - mean(phik')';
65      C_phi_0 = (1/N)*(phi_k*phi_k');
66      % C_phi_0 = cov(phi_k');
67
68      phi_k = phik(:,1:end-1) - mean(phik(:,1:end-1)')';
69      phi_k_plus_1 = phik(:,2:end) - mean(phik(:,2:end)')';
70      C_phi_1 = (1/(N-1))*(phi_k_plus_1*phi_k');
71
72      %% DETERMINE KALMAN GAIN
73      [A, Cw, K] = computeKalmanAR(C_phi_0, C_phi_1, G, sigmae);
74
75      %% RUN SIMULATION
76      phik = phiSim{i};
77
78      [var_eps_KF(i),y,yhat,eps] = AOloopAR(G,H,C_phi_0,sigmae,A,Cw,K,phik);
79
80      vaf_c = calculateVAF(y,yhat);
81      fprintf('%g VAF %g \n',i,vaf_c);
82      vaf_KF(i) = vaf_c;
83  end
84
85  %% QUESTION 3 - SUBSPACE IDENTIFICATION
86
87  clc
88  clear
89
90  load('turbulenceData.mat');
91  load('systemMatrices.mat');
92
93  Nr = 1;
94  var_eps_SubId = inf(1,Nr);
95  vaf_SubId = inf(1,Nr);
96
97  for i = 1:Nr
98
99      phi_id = phiIdent{i};
100     s_id = G*phi_id; %+ randn(size(G,1),5000)*sigmae;
101
102     N_id = 4981;
103     N_val = 4000;
104
105     n = 2; %144
106     s = 10;
107
108     [As,Cs,Ks] = SubId(s_id, N_id, N_val, s, n);
109     phi_sim = phiIdent{i};
110     [var_eps,eps_k_kp1_est,phi_k_kp1_est,phik] = AOloopSID(G,H,As,Cs,Ks,sigmae,phi_sim);
111
112     vaf_SubId(i) = calculateVAF(phik,phi_k_kp1_est);
113     var_eps_SubId(i) = mean(var_eps);
114 end
115
116 %% QUESTION 4 - CRITICAL THINKING
117
118 %% QUESTION 4.1
119
120 clc
121 clear
```

```
122
123 load('turbulenceData.mat');
124 load('systemMatrices.mat');
125
126 Nr = 20;
127 var_eps_RW_slopes = zeros(1,Nr);
128
129 for i = 1:Nr
130    % get data
131    phisim = phiSim{i};
132
133    % estimate C_phi_0 from this data
134    N = size(phisim,2);
135
136    % get the var_eps value for dataset i
137    [var_eps_RW_slopes(i),y,u] = AOloopRWslopes(G,H,sigmae,phisim);
138 end
139
140 % determine mean var_eps
141 var_mean_RW_slopes = mean(var_eps_RW_slopes);
142
143 %% QUESTION 4.2
144
145 %% QUESTION 4.3 - Unobservable modes
146
147 clc
148 clear
149 load('systemMatrices.mat');
150
151 rk = rank(G) % = 47
152 A = null(G);
153
154 % the columns in A span the null space of G
155 % in other words, they span the space containing the unobservable modes of
156 % G
157
158 % vec1 and vec2 are an orthonormal basis for the nullspace
159 vec1 = A(:,1);
160 vec2 = A(:,2);
161
162 % basis is orthonormal because
163 % - norm(vec1,2) = 1.000 and norm(vec2,2) = 1.000
164 % - vec1'*vec2 = 0.00 implying orthogonality
165 y = ones(49,1);
166 a = pinv(A)*y;
167
168 % unobservable mode corresponding to [1 1 1 .. 1]'
169 mode1 = vec1*a(1) + vec2*a(2);
170 mode2 = vec1; %vec1+vec2-mode1;
171
172 figure(1);
173 title('Unobservable Modes of G');
174 subplot(1,2,1);
175 imagesc(reshape(mode1,[7 7]));
176 xlabel('Mode 1');
177
178 subplot(1,2,2);
179 imagesc(reshape(mode2,[7 7]));
180 xlabel('Mode 2');
```

4.4.2 AOloopRWslopes

```
function [var_eps, epsk, u] = AOloopRWslopes(G,H,sigmae,phik)
    % Question 4.1 online AO simulation for closedloop measurements
    % IN
    % phik : incoming turbulence wavefront
    % sigmae: measurement noise parameter
    % H    : influence matrix mapping the wavefront on the mirror
    % G    : measurement matrix
    % OUT
    % sigma : mean variance of the residual wavefront

    n = size(H,1);        % n = 49 dimension lifted wavefront
    ns = size(G,1);       % ns = 72 dimension lifted sensor slopes
    T = length(phik);     % T = 5000 number of temporal phase points

    epsk = zeros(n,T);    % residual wavefront
    eps_piston_removed = zeros(n,T); % residual wavefront with mean removed (to avoid variance bias
        in results)
    sk = zeros(ns,T);     % ns = 72 slopes measurements
    u = zeros(n,T);       % n = 49 inputs
    sigma = zeros(1,T-2);

    yk = zeros(n,T);
    ykhat = zeros(n,T);

    % Obtain optimal u(k) given s(k) as u(k) = Ms(k)
    F = G*H;
    [U,S,V] = svd(F);

    n = 47;

    % decompose as defined on pg 34
    Sigma = S(1:n,1:n);
    U1 = U(:,1:n);
    U2 = U(:,n+1:end);

    V_transpose = V';

    V1 = V_transpose(1:n,:)';
    V2 = V_transpose(n+1:end,:)';

    Ftest = U1*Sigma*V1';

    norm(Ftest-F);

    M = V1/Sigma*U1';

    % initialize
    epsk(:,1) = phik(:,1);

    for k = 2:T-1
        % apply previous step control action
        epsk(:,k) = phik(:,k)-H*u(:,k-1);

        % simulate the sensor readings at time k+1
        sk(:,k) = G*epsk(:,k) + sigmae*randn(ns,1);

```

```
        %% NOW WE NEED TO DETERMINE THE OPTIMAL CONTROL INPUT

        % update next control input:
        u(:,k) = M*sk(:,k) + u(:,k-1);

        % from studocu
%         delta_u = Mstudocu*s(:,k);
%         u(:,k) = delta_u + u(:,k-1);

        % performance logging
        eps_piston_removed(:,k) = epsk(:,k)-mean(epsk(:,k));
        sigma(k-1) = var(eps_piston_removed(:,k));
    end

    var_eps = mean(sigma);
end
```

4.4.3 calculateVAF

```
function [vaf] = calculateVAF(y,yhat)

    % remove spatial mean
    y = y-mean(y);
    yhat = yhat-mean(yhat);

    N = Ny;
    top = 0;
    bottom = 0;

    for i = 1:N
        top = top + norm( y(:,i) - yhat(:,i) )^2;
        bottom = bottom + norm(y(:,i))^2;
    end
    vaf = 1-top/bottom;
end
```

References

[1] Konnik, M., and Doná, J. D. Waffle mode mitigation in adaptive optics systems: A constrained receding horizon control approach. In *2013 American Control Conference* (June 2013), pp. 3390–3396.

[2] Verhaegen, M., and Verdult, V. *Filtering and System Identification: A Least Squares Approach*, 1st ed. Cambridge University Press, New York, NY, USA, 2007.