CAMBRIDGE UNIVERSITY

ENGINEERING DEPARTMENT

SPEECH MODELLING: MODELS, PARAMETER ESTIMATION AND ITS APPLICATION TO SPEECH ENHANCEMENT

GA Smith [1], AJ Robinson [1], M Niranjan [2]

CUED/F-INFENG/TR.345

August 19, 1999

[1] Cambridge University Engineering Department
Trumpington Street
Cambridge. CB2 1PZ
England

E-mail: {gas1003,ajr}@eng.cam.ac.uk
http://www-svr.eng.cam.ac.uk/~{gas1003,ajr}

[2] Computer Science Department
Sheffield University
Sheffield. S1 4DP
England

E-mail: M.Niranjan@dcs.shef.ac.uk
http://www.dcs.shef.ac.uk/~niranjan/

Abstract

In this report system identification techniques are applied to speech. The purpose is to compare different models and different parameter estimation techniques on both a theoretical and an empirical basis. Results are then used in the practical problem of speech enhancement in additive white Gaussian noise.

Two model families are identified: polynomial and state space models. These are compared within a common state space framework, which makes explicit the assumptions and constraints of different models regarding process noise, observation noise, input-output delays and initial conditions. State space models are then cast in block matrix form because this representation is used in subspace state space system identification (4SID) methods. Next, three common parameter estimation techniques are compared: prediction error minimisation (PEM), instrumental variables (IV) and 4SID. These models and parameter estimation techniques are compared through experiments on real clean and noisy speech data, and evaluated in terms of their prediction errors, spectrograms and perceptual quality of the one-step-ahead predicted waveform. Finally, the best models are used to initialise Kalman filters which are used to filter noisy speech (where noise is white, additive and Gaussian) in the speech enhancement problem.

In general, the results are that modelling accuracy is improved by using the glottal waveform, a more general noise model and non-zero initial conditions. This is evident by reduced model prediction errors and better noise model spectrograms. Voiced speech can be more accurately modelled than non-voiced speech. Regarding parameter estimation techniques, PEM gives smallest prediction errors, then 4SID then IV. 4SID methods have advantages, for example the one-step-ahead predicted waveform does not seem to suffer from musical noise like PEM methods. Other advantages include numerical stability and the use of a frequency-weighted balanced state space basis, which allows model order to be reduced in a simpler and better manner. In addition, PEM and 4SID weight modelling errors differently in the frequency domain. PEM, 4SID and IV methods can be used to initialise a Kalman filter, which can be applied to speech enhancement.

Contents

1 Introduction

The aim of this report is to model the speech process. Although speech is generally non-stationary, its properties change slowly with time. Speech can therefore be divided into short frames during which the speech is essentially stationary. This report models each frame as a discrete time linear time-invariant (LTI) dynamical system. A system identification approach is adopted throughout.

1.1 System Identification

System identification is the process of inferring models of dynamical systems from observed data. A model is a mathematical description which relates variables. A dynamical system is where output depends on both the present and past history of inputs. System identification can be placed within a probabilistic or heuristic framework. System identification methods are well discussed by Ljung [21], Söderström and Stoica [30], and Heiji [13]. Juang [15] presents a more interdisciplinary approach. System identification consists of 3 stages which are often iterated until the selected model is sufficiently valid. These stages are described below and the system identification procedure is illustrated in figure 1.

1. **Specification**. This includes the design of the experiment, the selection of a model set and the selection of a criterion of fit for parameter estimation.
2. **Identification**. This includes the determination of the model which best describes the data given the criterion of fit, and the estimation of its parameters.
3. **Validation**. The validity of the model is tested.

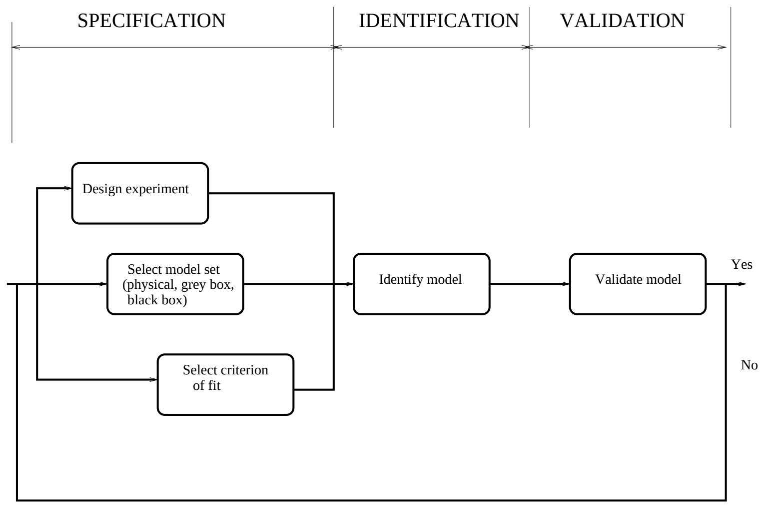

Figure 1: Procedure for sstem identification

This report has two principal aims. Firstly, different model sets are investigated. Secondly, different parameter estimation techniques are investigated. Model set selection is critical in system identification. The choice of a poor model can lead to poor model accuracy regardless of the quality of the parameter estimation algorithm. A priori information can aid model set selection; generally the more information the better.

There are three general categories of models.

- **Physical models**. These are constructed using the physical equations which underly the phenomenon, and require much a priori knowledge. They are generally constructed from first principles and are not data-derived.
- **Grey box models** are data-derived input-output mappings, where the equations have no particular physical meaning, but the parameters do.
- **Black box models** are data-derived input-output mappings, where parameters and equations have no particular physical meaning but simply aim to reconstruct the mapping as faithfully as possible.

The selection of a model set involves important questions such as whether the process is linear or non-linear, stationary or non-stationary, deterministic, stochastic or combined deterministic-stochastic. Deterministic models assume that the noise is not significant and that variables are observable and satisfy noise-free relationships. Stochastic models assume that noise is significant. Differences between deterministic and stochastic models can be introduced in two ways: either in the variables (the errors-in-variables approach), or in the equations which link the variables (the errors-in-equations approach).

For deterministic systems, the aim may be to model the system exactly or to obtain an approximation of the exact model. For stochastic systems, the aim is to obtain an approximate model. Approximation is necessary because the phenomenon may be complex and only partly known, or all explanatory variables may not be known. Model approximation may require the definition of measures of model complexity and distances between models. Differences between the identified and *true* models are due to modelling errors and disturbances.

This report concerns the modelling of the speech process. It is helpful to understand the physics of speech production and speech perception [3, 8, 24, 25]. Speech is a non-linear process which slowly varies with time. Both deterministic glottal waveform input and random noise drive the speech process. Deterministic input dominates during voiced speech, whereas random noise dominates during unvoiced speech. Therefore speech is a complicated natural process, and it is likely that at best only an approximate model can be derived. The approximation usually made is that speech is short-term stationary and linear. In this report experiments are conducted by dividing speech into short frames, and then modelling each frame as a linear time-invariant (LTI) dynamical system. A variety of stochastic, deterministic and combined deterministic-stochastic models are investigated which are principally black-box in nature. However the estimated parameters can often be understood in terms of the physics of the speech production process. Non-linear and non-stationary models are not included in the scope of this report.

1.2 The General Discrete Time LTI Model

A parametric linear time-invariant model can be represented by

$$\begin{aligned} y(t) &= G(q,\theta)u(t) + H(q,\theta)e(t) \qquad (1) \\ & \quad f_e(x,\theta) \quad \text{is the PDF of } e(t) \\ & \quad \{e(t)\} \quad \text{is white noise} \end{aligned}$$

$\{u(t), y(t)\}$ are system input-output pairs, $e(t)$ is termed the innovation or 1-step ahead prediction error, $f_e(x,\theta)$ is the probability density function of the innovations process, and θ is a vector of model parameters to be estimated. $qu(t) = u(t+1)$

and $q^{-1}u(t) = u(t-1)$ are the time-domain forward and backward shift operators respectively[1]. $G(q,\theta)$ and $H(q,\theta)$ are termed the transfer function and noise model respectively.

$$G(q,\theta) = \Sigma_{k=1}^{\infty} g(k,\theta)q^{-k} \tag{2}$$
$$H(q,\theta) = 1 + \Sigma_{k=1}^{\infty} h(k,\theta)q^{-k} \tag{3}$$

Equation 1 is a *probabilistic model* which requires the definition of H, G and $f_e(.)$. H and G are often assumed to be of finite length, and f_e is often assumed Gaussian and therefore described fully in terms of its first and second moments. With some manipulation, equation 1 can be written as

$$\hat{y}(t|\theta) = H^{-1}(q,\theta)G(q,\theta)u(t) + [1 - H^{-1}(q,\theta)]y(t) \tag{4}$$

where $\hat{y}(t|\theta)$ is the one-step ahead predictor. Equation 4 is termed the *predictor model* which does not require a probabilistic description of the innovations process. In fact it can be derived from a non-probabilistic approach. This predictor model can often be written in pseudolinear regression form

$$\begin{aligned} \hat{y}(t|\theta) &= \theta^T \phi(t,\theta) \\ &= \phi^T(t,\theta)\theta \end{aligned} \tag{5}$$

where θ is a vector of parameters, such as the coefficients of the transfer function and noise model, and $\phi(t,\theta)$ is a pseudo-regression vector containing ordered sequences of relevant past data, partly reconstructed using the current model. The exact form of the vectors depends on the type of transfer function and noise model used. If $\phi(t,\theta)$ does not depend on θ, then the relationship becomes linear regression, and $\phi(t)$ is termed the regression vector. Ljung [21] explains all these further.

1.3 Report Overview

The report investigates different discrete time dynamical models for the speech process and their assumptions regarding process noise, observation noise, deterministic input and non-zero initial conditions. Different parameter estimation techniques are then compared. Experiments are conducted on voiced and non-voiced speech, in both clean and noisy conditions, using various models and parameter estimation techniques.

The report is divided as follows. Sections 2 and 3 introduce the theory. The LTI dynamical system is introduced and then described in two representations: the polynomial and the state space representation. These are contrasted and compared. The state space representation is then extended to block matrix form. After this three parameter estimation algorithms are discussed: PEM (prediction error minimisation) , IV (instrumental variables) and 4SID (subspace state space system identification). Section 4 describes the tests used to validate a model and determine its modelling accuracy. Sections 5 and 6 describe and discuss the experiments on real speech, present results in tabular and graphical forms, and then apply these to the speech enhancement problem. Finally guidelines for future research and conclusions are made.

[1] If equations are written in the z-domain, then the forward shift operator would commonly be defined using the symbol z instead of q, giving $zu(t) = u(t+1)$. However in this report interest in focussed on time-domain rather than z-domain relationships. This difference is emphasised by using the symbol q instead, and the time-domain forward shift operator is defined as $qu(t) = u(t+1)$. Strictly speaking, the term transfer function should be reserved for z-domain transforms only eg $H(z)$. However for convenience the same terminology is used for the time-domain term $H(q)$ also. The corresponding frequency response is $H(e^{j\Omega})$ in both cases.

2 Model Set Selection

In this section, two general model families are presented and contrasted. These are termed the polynomial and state space families. Further explanation is given by Ljung [21].

2.1 The Polynomial Model Family

The most general polynomial description of the discrete time LTI model is

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \tag{6}$$

For most practical purposes, this structure is too general and one or several of the polynomials are often fixed to zero or unity. Polynomials are described in long-hand and short-hand notation as

$$\begin{array}{rclclcl}
A(q) & = & 1+ & \sum_{i=1}^{n_a} a_i q^{-i} & = & [1, a_1, \ldots, a_{n_a}] \\
B(q) & = & b_0+ & \sum_{i=1}^{n_b} b_i q^{-i} & = & [b_0, b_1, \ldots, b_{n_b}] \\
C(q) & = & 1+ & \sum_{i=1}^{n_c} c_i q^{-i} & = & [1, c_1, \ldots, c_{n_c}] \\
D(q) & = & 1+ & \sum_{i=1}^{n_d} d_i q^{-i} & = & [1, d_1, \ldots, d_{n_d}] \\
F(q) & = & 1+ & \sum_{i=1}^{n_f} f_i q^{-i} & = & [1, f_1, \ldots, f_{n_f}]
\end{array}$$

A non-zero b_0 implies that there is a zero-sample delay between input and output. In many cases of polynomial modelling, there is at least a unit-sample delay which means $b_0 = 0$.

2.2 The State Space Model Family

A second method to represent a discrete time LTI dynamical systems is to use the state space representation [16, 21, 28]. State space systems are defined in terms of a pair of equations.

$$x(t+1) = \mathbf{A}x(t) + \mathbf{B}u(t) + w(t) \tag{7}$$
$$y(t) = \mathbf{C}x(t) + \mathbf{D}u(t) + v(t) \tag{8}$$

$x(t) \in \mathbb{R}^{p\times 1}$, $\mathbf{A} \in \mathbb{R}^{p\times p}$, $\mathbf{B} \in \mathbb{R}^{p\times 1}$, $\mathbf{C} \in \mathbb{R}^{1\times p}$, $\mathbf{D} \in \mathbb{R}$ and p is the order of the system. $v(t) \in \mathbb{R}^{1\times 1}$ and $w(t) \in \mathbb{R}^{p\times 1}$ are noises. They are assumed zero-mean and temporally white. They are independent of each other, the states and the outputs.

Equation 7 is termed the state transition equation. The states $x(t)$ evolve according to a first-order Markov process with a deterministic input and corrupted by process noise. Equation 8 is the observation or measurement equation. Observations $y(t)$ are linear combinations of the states and deterministic input and corrupted by observation or measurement noise. Although the states $x(t)$ can be considered as the underlying causes of the process, these are hidden from the observer. Only the outputs $y(t)$ are observed. If the noises are assumed spatially Gaussian distributed such that $w(t) \sim N(0, \mathbf{Q})$ and $v(t) \sim N(0, \mathbf{R})$, then the system is termed a Gaussian dynamical system. Covariances are defined as $\mathbf{R} \in \mathbb{R}^{1\times 1}$ and $\mathbf{Q} \in \mathbb{R}^{p\times p}$.

The noise processes are essential elements of the model. Without observation noise, the states would no longer be hidden but be related linearly to the input and output. Without process noise and in the absence of deterministic input, the state $x(t)$ would either decay to zero, or increase to infinity in an exponential manner in the direction of the leading eigenvector of $\mathbf{A}$.

This state space representation can be more readily analysed if it is represented in the *forward innovations* form. Matrices are now defined in terms of a vector of parameters θ.

$$x(t+1) = \mathbf{A}(\theta)x(t) + \mathbf{B}(\theta)u(t) + \mathbf{K}(\theta)e(t) \tag{9}$$
$$y(t) = \mathbf{C}(\theta)x(t) + \mathbf{D}(\theta)u(t) + e(t) \tag{10}$$

$e(t) \in \mathbb{R}^{1\times 1}$ is termed the innovation and $\mathbf{K}(\theta) \in \mathbb{R}^{p\times 1}$ is termed the Kalman gain. So called because the Kalman filter operates on this state space representation to optimally predict $x(t)$ and $y(t)$. Using the q-shift operator, these equations can be written as

$$\begin{aligned} y(t) = {} & \left\{\mathbf{C}(\theta)[q\mathbf{I} - \mathbf{A}(\theta)]^{-1}\mathbf{B}(\theta) + \mathbf{D}(\theta)\right\}u(t) \\ & + \left\{\mathbf{C}(\theta)[q\mathbf{I} - \mathbf{A}(\theta)]^{-1}\mathbf{K}(\theta) + \mathbf{I}\right\}e(t) \end{aligned} \tag{11}$$

Comparing this with equation 1, the transfer function and noise model are written as

$$G(q,\theta) = \mathbf{C}(\theta)[q\mathbf{I} - \mathbf{A}(\theta)]^{-1}\mathbf{B}(\theta) + \mathbf{D}(\theta) \tag{12}$$
$$H(q,\theta) = \mathbf{C}(\theta)[q\mathbf{I} - \mathbf{A}(\theta)]^{-1}\mathbf{K}(\theta) + \mathbf{I} \tag{13}$$

2.3 Polynomial Models in State Space Form

Generally it is difficult to directly compare polynomial models with state space models because of their different representations. Moreover, state space models separate noise into process and observation noises, whereas polynomial models do not separate noise explicitly in such a manner. In this section the relationship between these two representations is investigated by converting state space models into some well-known polynomial models. Special notice is taken of the manner in which polynomial models consider observation and process noises and the constraints which they apply to state space system matrices. The polynomial models considered in this section are listed in table 1. Polynomial models are identified by acronyms. ARMAX means autoregressive moving average with exogenous inputs, ARX means autoregressive with exogenous inputs, OE means output-error, ARMA means autoregressive moving average, and AR means autoregressive.

Model	Equations		
ARMAX	$A(q)y(t) =$	$B(q)u(t)$	$+C(q)e(t)$
ARX	$A(q)y(t) =$	$B(q)u(t)$	$+e(t)$
OE	$F(q)y(t) =$	$B(q)u(t)$	$+F(q)e(t)$
ARMA	$A(q)y(t) =$		$+C(q)e(t)$
AR	$A(q)y(t) =$		$+e(t)$

Table 1: Common polynomial models

The conversion from state space to polynomial model is achieved by using the companion parametrisation (or observer canonical parametrisation) of the state space model as described in [16, 21]. Consider the state space system in forward innovations form with matrices defined in companion parametrisation as below. The θ dependence is dropped for convenience.

$$
\begin{aligned}
x(t+1) &= \mathbf{A}(\theta)x(t) + \mathbf{B}(\theta)u(t) + \mathbf{K}(\theta)e(t) \\
y(t) &= \mathbf{C}(\theta)x(t) + \mathbf{D}(\theta)u(t) + e(t)
\end{aligned}
$$

$$
\mathbf{A}(\theta) = \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \ddots & 0 \\ -a_{p-1} & 0 & & & 1 \\ -a_p & 0 & & & 0 \end{bmatrix} \quad \mathbf{K}(\theta) = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_{p-1} \\ k_p \end{bmatrix}
$$

$$
x(t) = \begin{bmatrix} x_1(t+1) & \\ x_2(t+1) & \\ \vdots & c \\ x_p(t+1) & \end{bmatrix} \qquad \mathbf{B}(\theta) = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{bmatrix}
$$

$$
\begin{aligned}
\mathbf{C}(\theta) &= \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \\
\mathbf{D}(\theta) &= [b_0] \\
\theta &= [a_1, a_2, \dots a_p, b_0, b_1, b_2, \dots, b_p, k_1, k_2, \dots, k_p]
\end{aligned}
$$

This state space system in companion parametrisation can be converted to an ARMAX model in polynomial representation, such that the ARMAX polynomials are written in terms of a_i,b_i and k_i. Specific details are given in Appendix 1. The ARMAX model is defined as

$$
A(q)y(t) = B(q)u(t) + C(q)e(t)
$$

The polynomials are defined in short-hand notation as

$$
\begin{aligned}
A(q) &= [1,\ a_1,\ a_2,\ \dots,\ a_p] \\
B(q) &= [0,\ b_1,\ b_2,\ \dots,\ b_p] + [b_0,\ a_1b_0,\ a_2b_0, \dots, a_pb_0] \\
C(q) &= [1,\ a_1,\ a_2,\ \dots,\ a_p] + [0,\ k_1,\ k_2,\ \dots,\ k_p]
\end{aligned}
$$

Short-hand notation is described in section 2.1. The following observations regarding the polynomials are made

The A(q) polynomial
This polynomial is responsible for autoregression on the output time series and is dependent on the a_i parameters.

The B(q) polynomial
This polynomial consists of two sets of parameters: b_i and a_i. The b_i parameters, for $i > 1$, define a moving average process acting on the exogenous input. The a_i parameters define a second moving average process acting on the input, but only exists if there is a feedforward term between input and output. In other words, only if there is a zero-sample time delay between input and output which means a non-zero **D**.

The C(q) polynomial
This is the most interesting polynomial. It is made up of two sets of parameters: a_i and k_i. The a_i parameters are due to observation noise. The k_i parameters are due to process noise. Therefore a_i shapes the observation noise contribution to the noise model, and k_i shapes the process noise contribution to the noise model.

There therefore exists a one-to-one mapping between an ARMAX model in state space observer canonical form and polynomial form. It is illuminating to consider what happens when process noise, observation noise and the **D** matrix are separately set to zero. In such a way it is possible to derive ARMA, ARX, and even AR models (with further constraints). These are shown together with another model called the direction-of-arrival (DOA) model in table 2 for an order 3 state space. Results can be readily extended to higher orders using similar mathematics as in Appendix 1. Two models are considered in a little more detail. These are the AR model and the DOA model.

2.4 The AR model

In this section, the AR model is derived from the 2nd ARMA model listed in the table. The ARMA model is given by

$$\begin{aligned} x(t+1) &= \mathbf{A}(\theta)x(t) + \mathbf{K}(\theta)e(t) && (14) \\ y(t) &= \mathbf{C}(\theta)x(t) && (15) \end{aligned}$$

If $\mathbf{K}(\theta) = [0, 1, 0, 0]^T$, then the polynomials reduce to

$$\begin{aligned} A(q) &= [1, a_1, a_2, a_3] \\ B(q) &= [0, 0, 0, 0] \\ C(q) &= [0, 1, 0, 0] \end{aligned}$$

The polynomial model is therefore $A(q)y(t) = q^{-1}e(t)$. Because $e(t)$ is a random process, time enumeration has little meaning, and so, neglecting end effects, this process is identical to $A(q)y(t) = e(t)$. Therefore the AR model assumes that observation noise is zero, and that process noise only enters via the second state component.

2.5 The DOA model

The DOA (direction-of-arrival) parametric form appears in two research fields: the time series problem of estimating decaying exponentials in noisy data, and the spatial problem of estimating the direction-of-arrival of sinusoids impinging on an antenna array. Refer to [26] and [34] for more details. In this section, the state space model listed under DOA in table 2 is developed further to give an equation which is standard for DOA problems.

Consider a stochastic system with no process noise and no deterministic input. There is therefore no energy driving the system. Instead the state vector decays exponentially to zero from an initial value or else explodes in an exponential manner in the direction of the leading eigenvector of **A**, depending on whether the leading eigenvalue is less than or greater than unity respectively. The state space equations are

$$\begin{aligned} x(t+1) &= \mathbf{A}(\theta)x(t) && (16) \\ y(t) &= \mathbf{C}(\theta)x(t) + e(t) && (17) \end{aligned}$$

ARMAX model	
$x(t+1) = \mathbf{A}x(t) + \mathbf{B}u(t) + \mathbf{K}e(t)$	$A(q) = [1, a_1, a_2, a_3]$
$y(t) = \mathbf{C}x(t) + \mathbf{D}u(t) + e(t)$	$B(q) = [0, b_1, b_2, b_3] + b_0[1, a_1, a_2, a_3]$
	$C(q) = [1, a_1, a_2, a_3] + [0, k_1, k_2, k_3]$
$x(t+1) = \mathbf{A}x(t) + \mathbf{B}u(t) + \mathbf{K}e(t)$	$A(q) = [1, a_1, a_2, a_3]$
$y(t) = \mathbf{C}x(t) + e(t)$	$B(q) = [0, b_1, b_2, b_3] + [0, 0, 0, 0]$
	$C(q) = [1, a_1, a_2, a_3] + [0, k_1, k_2, k_3]$
$x(t+1) = \mathbf{A}x(t) + \mathbf{B}u(t) + \mathbf{K}e(t)$	$A(q) = [1, a_1, a_2, a_3]$
$y(t) = \mathbf{C}x(t) + \mathbf{D}u(t)$	$B(q) = [0, b_1, b_2, b_3] + b_0[1, a_1, a_2, a_3]$
	$C(q) = [0, 0, 0, 0] + [0, k_1, k_2, k_3]$
OE model	
$x(t+1) = \mathbf{A}x(t) + \mathbf{B}u(t)$	$A(q) = [1, a_1, a_2, a_3]$
$y(t) = \mathbf{C}x(t) + \mathbf{D}u(t) + e(t)$	$B(q) = [0, b_1, b_2, b_3] + b_0[1, a_1, a_2, a_3]$
	$C(q) = [1, a_1, a_2, a_3] + [0, 0, 0, 0]$
$x(t+1) = \mathbf{A}x(t) + \mathbf{B}u(t)$	$A(q) = [1, a_1, a_2, a_3]$
$y(t) = \mathbf{C}x(t) + e(t)$	$B(q) = [0, b_1, b_2, b_3] + [0, 0, 0, 0]$
	$C(q) = [1, a_1, a_2, a_3] + [0, 0, 0, 0]$
ARX model	
$x(t+1) = \mathbf{A}x(t) + \mathbf{B}u(t) + \mathbf{K}^\sharp e(t)$	$A(q) = [1, a_1, a_2, a_3]$
$y(t) = \mathbf{C}x(t) + \mathbf{D}u(t)$	$B(q) = [0, b_1, b_2, b_3] + b_0[1, a_1, a_2, a_3]$
	$C(q) = [0, 0, 0, 0] + [0, 1, 0, 0]$
$x(t+1) = \mathbf{A}x(t) + \mathbf{B}u(t) + \mathbf{K}^\sharp e(t)$	$A(q) = [1, a_1, a_2, a_3]$
$y(t) = \mathbf{C}x(t)$	$B(q) = [0, b_1, b_2, b_3] + [0, 0, 0, 0]$
	$C(q) = [0, 0, 0, 0] + [0, 1, 0, 0]$
ARMA model	
$x(t+1) = \mathbf{A}x(t) + \mathbf{K}e(t)$	$A(q) = [1, a_1, a_2, a_3]$
$y(t) = \mathbf{C}x(t) + e(t)$	$B(q) = [0, 0, 0, 0] + [0, 0, 0, 0]$
	$C(q) = [1, a_1, a_2, a_3] + [0, k_1, k_2, k_3]$
$x(t+1) = \mathbf{A}x(t) + \mathbf{K}e(t)$	$A(q) = [1, a_1, a_2, a_3]$
$y(t) = \mathbf{C}x(t)$	$B(q) = [0, 0, 0, 0] + [0, 0, 0, 0]$
	$C(q) = [0, 0, 0, 0] + [0, k_1, k_2, k_3]$
AR model	
$x(t+1) = \mathbf{A}x(t) + \mathbf{K}^\sharp e(t)$	$A(q) = [1, a_1, a_2, a_3]$
$y(t) = \mathbf{C}x(t)$	$B(q) = [0, 0, 0, 0] + [0, 0, 0, 0]$
	$C(q) = [0, 0, 0, 0] + [0, 1, 0, 0]$
DOA model $^\natural$	
$x(t+1) = \mathbf{A}x(t)$	$A(q) = [1, a_1, a_2, a_3]$
$y(t) = \mathbf{C}x(t) + e(t)$	$B(q) = [0, 0, 0, 0] + [0, 0, 0, 0]$
	$C(q) = [1, a_1, a_2, a_3] + [0, 0, 0, 0]$
$^\sharp$ $\mathbf{K} = [0, 1, 0]$	
$^\natural$ Requires non-zero initial conditions on the state vector.	

Table 2: Polynomial models in companion state space form. For convenience, the dependency on θ is dropped and the state space is order 3.

Consider a linear transform of the state space $\tilde{x}(t) = \mathbf{R}x(t)$, where $\mathbf{R}$ is non-singular. The state equations can be rewritten as

$$\tilde{x}(t+1) = \mathbf{R}\mathbf{A}(\theta)\mathbf{R}^{-1}\tilde{x}(t) \tag{18}$$
$$y(t) = \mathbf{C}(\theta)\mathbf{R}^{-1}\tilde{x}(t) + e(t) \tag{19}$$

If $\mathbf{A}(\theta)$ has p linearly independent eigenvectors, then $\mathbf{R}$ can be chosen such that it diagonalises the state transition matrix $\mathbf{R}\mathbf{A}(\theta)\mathbf{R}^{-1} = diag(\lambda_1, \ldots \lambda_p)$ where λ_i are the eigenvalues of $\mathbf{A}(\theta)$. Refer to Kailath [16] for more details. However if $\mathbf{A}(\theta)$ does not have p linearly independent eigenvectors, then it is not possible to diagonalise the $\mathbf{A}$ matrix. Once in diagonal form, the output becomes a sum of decaying exponentials corrupted by additive output noise, where initial amplitudes are determined by the initial conditions on the state vector $x_{0,i}$, and the entries of the $\mathbf{C}(\theta)$ matrix c_i

$$y(t) \quad = \quad \sum_{i=1}^{p} c_i \; x_{0,i} \; \lambda_i^t + e(t) \tag{20}$$

This parametric form is the standard DOA equation. From this argument, it becomes clear that the DOA model is a state space system where process noise is zero, observation noise need not be zero, and the state vector has non-zero initial conditions. Further details on the DOA model and how DOA algorithms adopt subspace state space parameter estimation methods are discussed more fully in Appendix 2. The DOA model can be realised in state space or polynomial form.

2.6 State Space Models in Block Matrix Form

State space models can be extended one step further to block matrix form. The reason for this is because this block matrix formulation is used in the 4SID algorithms which have gained recent popularity, particularly in the control literature. 4SID is an acronym for subspace state space system identification and is explained in section 3.3. Consider the general state space equations written in forward innovations form as in equations 9 and 10 except the dependency on θ is dropped. Refer to Van Overschee and De Moor [39] and Chui [7]. In this section single-input-single-output (SISO) systems only are considered.

$$x(t+1) \quad = \quad \mathbf{A}x(t) + \mathbf{B}u(t) + \mathbf{K}e(t) \tag{21}$$
$$y(t) \quad = \quad \mathbf{C}x(t) + \mathbf{D}u(t) + e(t) \tag{22}$$

Consider input $u(t)$ in Hankel matrix form. The number of rows in the Hankel matrix equals $2i$ where i is termed the block size. The number of columns is j. Therefore each Hankel matrix contains samples from time 0 to $(2i + j - 2)$.

$$\mathbf{U}_{0|2i-1} \stackrel{\text{def}}{=} \begin{bmatrix} u(0) & u(1) & \ldots & u(j-1) \\ u(1) & u(2) & \ldots & u(j) \\ \vdots & \vdots & & \vdots \\ u(2i-1) & u(2i) & \ldots & u(2i+j-2) \end{bmatrix}$$

This $\mathbf{U}_{0|2i-1}$ can be segmented into two blocks each with i number of rows.

$$
\mathbf{U}_{0|2i-1} \stackrel{def}{=} \left[\begin{array}{ccccc}
u(0) & u(1) & u(2) & \dots & u(j-1) \\
u(1) & u(2) & u(3) & \dots & u(j) \\
\dots & \dots & \dots & \dots & \dots \\
u(i-1) & u(i) & u(i+1) & \dots & u(i+j-2) \\
\hline
u(i) & u(i+1) & u(i+2) & \dots & u(i+j-1) \\
u(i+1) & u(i+2) & u(i+3) & \dots & u(i+j) \\
\dots & \dots & \dots & \dots & \dots \\
u(2i-1) & u(2i) & u(2i+1) & \dots & u(2i+j-2)
\end{array}\right]
$$

$$
\stackrel{def}{=} \left[\begin{array}{c} \mathbf{U}_{0|i-1} \\ \hline \mathbf{U}_{i|2i-1} \end{array}\right] \stackrel{def}{=} \left[\begin{array}{c} \mathbf{U}_p \\ \hline \mathbf{U}_f \end{array}\right]
$$

The output and innovations are similarly formed into Hankel matrices to give $\mathbf{Y}_{0|2i-1}$ and $\mathbf{E}_{0|2i-1}$ which are similarly separated into upper and lower blocks. For convenience of notation, the upper block is referred to as the *past* data, and the lower block as the *future* data, denoted with subscripts p and f respectively. Thus the following definitions are made:

$$
\mathbf{U}_{0|2i-1} \stackrel{def}{=} \left[\begin{array}{c} \mathbf{U}_p \\ \hline \mathbf{U}_f \end{array}\right], \quad \mathbf{Y}_{0|2i-1} \stackrel{def}{=} \left[\begin{array}{c} \mathbf{Y}_p \\ \hline \mathbf{Y}_f \end{array}\right], \quad \mathbf{E}_{0|2i-1} \stackrel{def}{=} \left[\begin{array}{c} \mathbf{E}_p \\ \hline \mathbf{E}_f \end{array}\right]
$$

where

$$
\begin{array}{ll}
\mathbf{U}_p \stackrel{def}{=} \mathbf{U}_{0|i-1}, & \mathbf{U}_f \stackrel{def}{=} \mathbf{U}_{i|2i-1} \\
\mathbf{Y}_p \stackrel{def}{=} \mathbf{Y}_{0|i-1}, & \mathbf{Y}_f \stackrel{def}{=} \mathbf{Y}_{i|2i-1} \\
\mathbf{E}_p \stackrel{def}{=} \mathbf{E}_{0|i-1}, & \mathbf{E}_f \stackrel{def}{=} \mathbf{E}_{i|2i-1}
\end{array}
$$

Having conveniently represented input, output and innovations data, it is now necessary to represent the state vectors in a block matrix form. Consider a sequence of state vectors from time $t = 0$ to time $t = (j-1)$ stacked side-by-side in a single matrix $\mathbf{X}_{0,j-1}$. Consider a matrix $\mathbf{X}_{i,i+j-1}$ similarly defined.

$$
\begin{aligned}
\mathbf{X}_{0,j-1} &\stackrel{\text{def}}{=} \left[\begin{array}{ccccc} x(0) & x(1) & x(2) & \dots & x(j-1) \end{array}\right] \\
\mathbf{X}_{i,i+j-1} &\stackrel{\text{def}}{=} \left[\begin{array}{ccccc} x(i) & x(i+1) & x(i+2) & \dots & x(i+j-1) \end{array}\right]
\end{aligned}
$$

By some trivial calculation using the state space equations in innovations form, it can be shown that the following equations hold:

$$
\mathbf{X}_{i,i+j-1} = \mathbf{A}^i \mathbf{X}_{0,j-1} + \mathbf{\Delta}_i^d \mathbf{U}_{0|i-1} + \mathbf{\Delta}_i^w \mathbf{E}_{0|i-1} \tag{23}
$$

$$
\mathbf{Y}_{0|i-1} = \mathbf{\Gamma}_i \mathbf{X}_{0,j-1} + \mathbf{H}_i^d \mathbf{U}_{0|i-1} + \mathbf{H}_i^w \mathbf{E}_{0|i-1} \tag{24}
$$

$\mathbf{\Delta}_i^d$ and $\mathbf{\Delta}_i^w$ are reversed extended controllability matrices, $\mathbf{\Gamma}_i$ is the extended observability matrix and $\mathbf{H}_i^d$ and $\mathbf{H}_i^w$ are Toeplitz matrices. These are all defined below where $\mathbf{I}_1$ is a 1×1 identity matrix.

$$
\begin{aligned}
\mathbf{\Delta}_i^d &\stackrel{\text{def}}{=} \left[\begin{array}{ccccc} \mathbf{A}^{i-1}\mathbf{B} & \mathbf{A}^{i-2}\mathbf{B} & \dots & \mathbf{AB} & \mathbf{B} \end{array}\right] \\
\mathbf{\Delta}_i^w &\stackrel{\text{def}}{=} \left[\begin{array}{ccccc} \mathbf{A}^{i-1}\mathbf{K} & \mathbf{A}^{i-2}\mathbf{K} & \dots & \mathbf{AK} & \mathbf{K} \end{array}\right]
\end{aligned}
$$

$$
\mathbf{\Gamma}_i \stackrel{\text{def}}{=} \left[\begin{array}{c} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{i-1} \end{array}\right] \quad
\mathbf{H}_i^w \stackrel{\text{def}}{=} \left[\begin{array}{cccc} \mathbf{I}_1 & & & 0 \\ \mathbf{CK} & \mathbf{I}_1 & & \\ \vdots & \ddots & \ddots & \\ \mathbf{CA}^{i-2}\mathbf{K} & \dots & \mathbf{CK} & \mathbf{I}_1 \end{array}\right] \quad
\mathbf{H}_i^d \stackrel{\text{def}}{=} \left[\begin{array}{cccc} \mathbf{D} & & & 0 \\ \mathbf{CB} & \mathbf{D} & & \\ \vdots & \ddots & \ddots & \\ \mathbf{CA}^{i-2}\mathbf{B} & \dots & \mathbf{CB} & \mathbf{D} \end{array}\right]
$$

This is the full state space block representation of the system. The notation for equations 23 and 24 can be simplified to express relationships between the past and the future. Note the similarity with the state space equations 9 and 10.

$$
\begin{aligned}
\mathbf{X}_f &= \mathbf{A}^i\mathbf{X}_p + \mathbf{\Delta}_i^d\mathbf{U}_p + \mathbf{\Delta_i}^w\mathbf{E}_p & (25)\\
\mathbf{Y}_p &= \mathbf{\Gamma}_i\mathbf{X}_p + \mathbf{H}_i^d\mathbf{U}_p + \mathbf{H}_i^w\mathbf{E}_p & (26)\\
\mathbf{Y}_f &= \mathbf{\Gamma}_i\mathbf{X}_f + \mathbf{H}_i^d\mathbf{U}_f + \mathbf{H}_i^w\mathbf{E}_f & (27)
\end{aligned}
$$

2.7 Initial Conditions

There are two methods to make an optimal estimate of the dynamics of a process at time $t = T$, or a prediction of the output at time $t > T$. The first method involves knowing all input data from time $t = -\infty$ to $t = T$. The second method involves knowing all input data from say $t = 0$ to $t = T$ plus the initial conditions on the state vector at $t = 0$. The state vector has the property of storing information about the past history of inputs for $t < 0$. Therefore, given a *finite* amount of input data, it is possible to estimate without bias the dynamics of the process and to predict outputs only if initial conditions on the state vector are known. Otherwise bias is introduced into the estimate. This shows the importance of modelling the initial conditions of the state accurately. Both polynomial and state space models allow initial conditions to be included in the representation. A difficult problem may be to estimate accurately this initial state without a priori knowledge, which becomes an identification or parameter estimation problem.

However as data lengths get longer, the effects of initial conditions become less and less for an asymptotically stable system, at a rate determined by the time constants of the system. These in turn depend on the eigenvalues of $\mathbf{A}(\theta)$. So for long data lengths, estimates of the dynamics are fairly accurate even if the initial state vector is unknown and assumed zero instead.

2.8 A Comparison Between State Space and Polynomial Models

A significant difference between the polynomial and state space models concerns model order reduction. Consider the state space system in general form, where dependency on θ is again dropped for convenience of notation.

$$
\begin{aligned}
x(t+1) &= \mathbf{A}x(t) + \mathbf{B}u(t) + w(t) & (28)\\
y(t) &= \mathbf{C}x(t) + \mathbf{D}u(t) + v(t) & (29)
\end{aligned}
$$

The state space is not unique but shows rotational invariance. Let $\tilde{x} = \mathbf{R}x$, where $\mathbf{R}$ is a non-singular matrix called a similarity transformation. $\mathbf{R}$ is often rotational, but may also be time-dependent or may change the number of states. The state space equations can be rewritten as

$$
\begin{aligned}
\tilde{x}(t+1) &= \mathbf{RAR}^{-1}\tilde{x}(t) + \mathbf{RB}u(t) + \mathbf{R}w(t) & (30)\\
y(t) &= \mathbf{CR}^{-1}\tilde{x}(t) + \mathbf{D}u(t) + v(t) & (31)
\end{aligned}
$$

The system matrices $\mathbf{A}$ and $\mathbf{RAR}^{-1}$ are said to be similar. Therefore it is meaningless to talk of the states of a system but instead the states of a particular *realization* of the system. Now consider partitioning the state space such that $x(t) = [x_1(t)^T x_2^T(t)]^T$.

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{A}_{12} \\ \mathbf{A}_{21}\mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} u(t) + w(t) \quad (32)$$

$$y(t) = [\mathbf{C}_1\mathbf{C}_2] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \mathbf{D}u(t) + v(t) \quad (33)$$

Elimination of the $x_2(t)$ subspace is achieved by simple truncation of the state space matrices. This gives a reduced order state space system.

$$x_1(t+1) = \mathbf{A}_{11}x_1(t) + \mathbf{B}_1u(t) + w(t) \quad (34)$$

$$y(t) = \mathbf{C}_1x_1(t) + \mathbf{D}u(t) + v(t) \quad (35)$$

Subspace elimination is therefore used in the low order approximation of a higher order system. The choice of state space realization or basis determines the subspace rejected and the frequency-domain distribution of errors between the high order and low order model.

One method of reducing the order of a polynomial model is simply to truncate its polynomials. In effect, this is the same as first casting the polynomial model into companion parametrisation state space form and then eliminating a subspace using the mathematics as described above. However the subspace eliminated may not coincide with the *weakest* subsystem of the model, and therefore may lead to the elimination of important information pertaining to the dynamics of the process. A more meaningful method for model order reduction is first to find a *balanced* state space basis [23]. Once the state space is cast in such a balanced realization, then simple truncation of state space matrices *does* result in the elimination of the weakest subsystem of the model. The rejected subsystem may either be noise, or may be signal modes which play an insignificant part in the dynamics of the process.

Therefore the state space model has the advantage over the polynomial model in that there is greater flexibility in the choice of state space basis. This basis can be chosen in an optimal manner to facilitate model order reduction. 4SID methods represent the state space in a *frequency-weighted* balanced form. This frequency-weighting weights the frequency-domain distribution of modelling errors between the high and low order models. Refer to Enns [9], Van Overschee and De Moor [36, 39] and Kim et alia [18] for further details, including discussion on error bounds.

Apart from the frequency-weighted balanced basis, another important state space basis is that of the diagonal form of $\mathbf{A}$. Diagonalisation of the $\mathbf{A}$ matrix means that the output $y(t)$ is a sum of independent *modes* which evolve separately. These physical modes can be interpreted as belonging to certain speech formants or the speech harmonics (for voiced speech) depending on the dimensionality of the state space. The equations then describe how much noise is added to each mode. Moreover time constraints may be added to control the evolution in time of each mode. However it is only possible to diagonalise the $\mathbf{A}$ matrix if it has linearly independent eigenvectors.

2.9 Summary

A discrete-time LTI dynamical system is described in terms of a transfer function and a noise model.

$$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t) \tag{36}$$

This model can be represented in two ways: in polynomial or state space representation. To compare these two representations on a like-by-like basis, many polynomial models can be cast into state space form: for example, ARMAX, ARX, OE, ARMA and AR. The advantage of such a comparison is that it becomes clear that polynomial models often make assumptions about the structure of the system matrices and the nature of observation and process noise. Furthermore, order reduction of polynomial models by simple truncation of polynomials to remove insignificant signal modes or noise modes does not necessarily result in the elimination of the weakest subsystem. A state space model on the other hand can be chosen which applies no constraints to the system matrices and noise, and casts the state space into a balanced basis, which lends itself readily to order reduction by elimination of the weakest subsystem. Both polynomial and state space models allow initial conditions for the state vector to be included.

3 Parameter Estimation Techniques

Having investigated different models for the speech process and different ways of representing these, it is necessary to consider algorithms to estimate the parameters of these models. There are three popular methods for parameter estimation. These are prediction error minimisation (PEM), instrumental variables (IV) and subspace state space system identification (4SID). Maximum likelihood (ML) and least squares (LS) are subsets of PEM. Direction-of-arrival (DOA) methods are generally a subset of 4SID. The purpose of this section is to outline PEM, IV and 4SID parameter estimation algorithms briefly. A detailed investigation of the relationship between these three algorithms and their subsets regarding the distribution of modelling errors is left for a future report. Below is a description of the fundamentals of the three techniques.

3.1 Prediction Error Minimisation (PEM)

PEM methods are discussed extensively by Ljung [21]. They attempt to determine the parameter θ which minimises some function of a possibly prefiltered sequence of say N samples of the innovations e_F. These innovations are equal to the one-step-ahead prediction errors which explains the meaning of the PEM acronym.

$$\theta \;= \arg_\theta \;\min\; \textstyle\sum_{t=1}^{N} f\big(e_F(t,\theta)\big) \tag{37}$$

Often a squared norm is selected as the function, so that PEM minimises the sum-squared innovations sequence.

$$\begin{aligned} \theta &= \arg_\theta \;\min\; \sum_{t=1}^{N} e_F^2(t,\theta) && (38)\\ &= \arg_\theta \;\min\; \sum_{t=1}^{N} \Big(H(q,\theta)^{-1}\big[y_F(t) - G(q,\theta)u_F(t)\big]\Big)^2 && (39) \end{aligned}$$

where $y_F(t)$ and $u_F(t)$ are prefiltered output and input respectively. For AR and ARX models this optimisation reduces to a single-step least-squares algorithm. However for more complicated models such as ARMA and ARMAX, the algorithm is iterative and may become prone to problems common to iterative algorithms such as convergence to local rather than global optima, sensitivity to initial conditions, slow convergence, lack of convergence etc.

Maximum likelihood (ML) methods are probabilistic methods. In some cases these may be considered as PEM methods. For example, consider the following probability model for the filtered innovations: innovations are assumed Gaussian distributed, zero mean and temporally independent. Innovations are scalar and the innovations covariance matrix reduces to a scalar variance $\sigma_{e_F}^2$ which must be known a priori.

$$\begin{aligned} p(\{e_F(1),\ldots,e_F(N)\}|\theta,\sigma_{e_F}) &= \Pi_{t=1}^{N} p(e_F(t)|\theta,\sigma_{e_F}) && (40)\\ &= \Pi_{t=1}^{N}(2\pi\sigma_{e_F}^2)^{-1/2}\exp\left\{-\frac{e_F^2(t,\theta)}{2\sigma_{e_F}^2}\right\} && (41) \end{aligned}$$

ML determines the following estimate of θ.

$$
\begin{aligned}
\theta &= \arg_\theta \max \log p(\{e_F(1),\ldots,e_F(N)\}|\theta,\sigma_{e_F}) \\
&= \arg_\theta \max \sum_{t=1}^{N} -\frac{e_F^2(t,\theta)}{2\sigma_{e_F}^2} \\
&= \arg_\theta \min \sum_{t=1}^{N} \frac{e_F^2(t,\theta)}{2\sigma_{e_F}^2} \\
&= \arg_\theta \min \sum_{t=1}^{N} e_F^2(t,\theta) \\
&= \arg_\theta \min \sum_{t=1}^{N} \Big(H(q,\theta)^{-1}\left[y_F(t) - G(q,\theta)u_F(t)\right] \Big)^2 \qquad (42)
\end{aligned}
$$

For scalar innovations sequences of known variance, ML estimation is therefore identical to PEM with a squared innovation norm. In some cases, maximum likelihood methods are therefore a subset of PEM. The relationship between PEM and ML is discussed by Åstrom [1] and Ljung [21].

3.2 Instrumental Variables (IV)

The motivation behind instrumental variable methods is that the prediction errors for a given model should be independent of and uncorrelated with past data. Otherwise the predictor $\hat{y}(t|\theta)$ misses some important dynamics and information concerning the output which feeds through to the prediction errors. The aim of IV methods is to estimate a model such that the prediction errors are maximally uncorrelated with some vector of the past data ζ. This is termed the correlation vector, instrument or instrumental variable. Usually ζ is selected to be finite-dimensional and may simply be a vector sequence of past output and input samples. Consider a model where the predictor $\hat{y}(t|\theta)$ is given as in the predictor model in equation 5

$$
\hat{y}(t|\theta) = \phi^T(t,\theta)\theta
$$

where $\theta \in \mathbb{R}^{d\times 1}$. If the vector $\phi(t,\theta)$ is independent of the model parameters θ then the equation is termed *linear* regression. Otherwise it is known as *pseudo-linear* regression. The IV method can be summarised as follows.

- Determine a filtered sequence of prediction errors, where $L(q)$ is a linear prefilter.

$$
e_F(t,\theta) = L(q)\left[y(t) - \phi^T(t)\theta\right] \qquad (43)
$$

- Determine a suitable set of instruments which operates on the *past* input sequence. These instruments $\zeta(t,\theta)$ should be maximally uncorrelated with observation noise, but maximally correlated with the regression variable ϕ. $K_u(q,\theta)$ is a d-dimensional column vector of linear filters.

$$
\zeta(t,\theta) = K_u(q,\theta)u(t) \qquad (44)
$$

- Determine the parameter θ which maximally uncorrelates the instruments with the filtered prediction errors. A shaping function α can be introduced.

$$\theta^{IV} = \arg_\theta \min \left| \frac{1}{N} \sum_{t=1}^{N} \zeta(t,\theta)\alpha(\epsilon_F(t,\theta)) \right|^2 \qquad (45)$$

  An appropriate norm is selected. If the dimensions of the instruments ζ equals d, then θ^{IV} is determined as that value of θ where the right-hand side of equation 45 equals zero and this is termed the *IV method*. If the dimension of $\zeta > d$, it is necessary to minimise the norm, and the technique is termed the *extended IV method*.

An advantage of the IV method is its simplicity. Ljung in chapter 15 of [21] advises that IV methods can be used for a quick estimate of the transfer function, which can later be refined if necessary by a PEM method. This is in fact the general algorithm which several MatLab iterative PEM functions adopt when no user-defined initialisation estimates are given.

In this report two IV methods are investigated: one for an AR model and the other for an ARX model. The AR method employed during experiments is an algorithm which computes an approximately optimal choice of IV procedure to estimate the AR part of a scalar time series $A(q)y(t) = v(t)$. The noise sequence $v(t)$ is assumed to be a moving average process. Refer to [33] for further details and the *ivar* MatLab function in the manual [22]. The second IV method estimates the parameters of an ARX model using an approximately optimal 4-stage instrumental variable procedure as detailed in chapter 15 of [21] and the *iv4* MatLab function [22].

3.3 Subspace State Space System Identification (4SID)

PEM and IV methods operate by summing over the innovations sequence, and therefore act on a vector sequence. 4SID methods on the other hand work with block matrices. Van Overschee and De Moor present a comprehensive overview of subspace system identification methods in [36, 37, 38, 39].

Consider the most general state space model: the combined deterministic-stochastic state space model. Equations 25, 26 and 27 give this model in forward innovations block matrix form. 4SID methods operate on these forward innovations block matrix equations by identifying subspaces which are orthogonal to nuisance signals (such as noise and deterministic input), but correlated with signals of interest. 4SID methods then project input and output data down onto these subspaces. Hence the reasoning behind the name: subspace state space system identification.

4SID algorithms are generally composed of two stages

- low-rank approximation and estimation of the extended observability matrix directly from the input-output data, by means of an oblique or orthogonal projection.

- estimation of the system matrices from the column space of the observability matrix using a least-squares (LS) criterion.

Consider the future output block equation from equation 27

$$\mathbf{Y}_f = \mathbf{\Gamma}_i \mathbf{X}_f + \mathbf{H}_i^d \mathbf{U}_f + \mathbf{H}_i^w \mathbf{E}_f \qquad (46)$$

Denote matrix $\mathbf{W}_p = (\mathbf{U}_p^T \mathbf{Y}_p^T)^T$. To remove noise and deterministic input effects, matrices undergo an oblique projection[2] onto the row space of $\mathbf{W}_p$ along the row space of $\mathbf{U}_f$ (refer to Appendix 3 for details of the oblique projection).

$$\mathbf{Y}_f/_{\mathbf{U_f}}\mathfrak{W}_\mathrm{p} = \mathbf{\Gamma}_i\mathbf{X}_f/_{\mathbf{U_f}}\mathfrak{W}_\mathrm{p} + \mathbf{H}_i^d\mathbf{U}_f/_{\mathbf{U_f}}\mathfrak{W}_\mathrm{p} + \mathbf{H}_i^w\mathbf{E}_f/_{\mathbf{U_f}}\mathfrak{W}_\mathrm{p} \tag{47}$$

By definition $\mathbf{U}_f/_{\mathbf{U_f}}\mathfrak{W}_\mathrm{p} = 0$ and provided the Hankel matrices are sufficiently large, then $\mathbf{E}_f/_{\mathbf{U_f}}\mathfrak{W}_\mathrm{p} \approx 0$. Equation 47 can therefore be rewritten as

$$\begin{aligned} \mathbf{Y}_f/_{\mathbf{U_f}}\mathfrak{W}_\mathrm{p} &= \mathbf{\Gamma}_i\mathbf{X}_f/_{\mathbf{U_f}}\mathfrak{W}_\mathrm{p} \\ &= \mathbf{\Gamma}_i\mathbf{X}_f \end{aligned} \tag{48}$$

Greater flexibility is obtained by weighting the oblique projection with weighting matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ and analysing the weighted oblique projection: $\mathbf{W}_1(\mathbf{Y}_f/_{\mathbf{U_f}}\mathfrak{W}_\mathrm{p})\mathbf{W}_2$. $\mathbf{W}_1$ and $\mathbf{W}_2$ relate to weighting the optimisation criterion in the frequency domain. This has the effect of weighting the frequency distribution of prediction errors.

Therefore the column space of the oblique projection $\mathbf{Y}_f/_{\mathbf{U_f}}\mathfrak{W}_\mathrm{p}$ matches the column space of the extended observability matrix. From the column space of this observability matrix the system matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ can be determined by a least-squares criterion. The process and observation noise covariances $(\mathbf{Q}, \mathbf{R})$ respectively can also be calculated. Refer to [39] for specific algorithm details. Unlike some PEM methods, 4SID methods are non-iterative.

In the absence of a deterministic input signal, the equations become stochastic only. Thus the oblique becomes an orthogonal projection onto $\mathbf{Y}_p$, and is similar to instrumental variable (IV) methods as explained in [41].

There are a variety of 4SID algorithms. These include N4SID (numerical algorithms for subspace state space system identification), MOESP (multivariable output-error state space) and CVA (canonical variate analysis) [39, 41]. N4SID, MOESP and CVA differ with respect to the weighting matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ and the implementation details of the algorithms, but are similar in theory.

The DOA (direction-of-arrival) problem which concerns determining the direction of arrival of signals impinging on an antenna array can also be cast in a subspace format. Refer to Appendix 2 for further details.

3.4 Summary

There are three popular methods for parameter estimation: PEM, IV and 4SID. In some cases ML can be considered a subset of PEM. DOA algorithms are a subset of 4SID. PEM methods are most popular with polynomial models and attempt to minimise a sequence of prediction errors. However algorithms may be iterative and complicated. IV methods are simpler than PEM methods and rely on minimising the correlation between the prediction errors and past data. 4SID methods work on state space models in block matrix form. They operate by identifying low-order subspaces which are orthogonal to undesirable signals such as deterministic input and noise, project the state space system onto these subspaces, and then estimate state space system matrices. 4SID algorithms are non-iterative.

[2]This oblique projection coincides with a minimum squared error between true data Y_f and its linear prediction from $\mathbf{W}_p$ and $\mathbf{U}_f$ [39]

4 Validation of Models

Section 1.1 describes system identification as consisting of three stages: specification, identification and validation. Section 2 addresses the issue of model set selection. Section 3 addresses the issue of identification of model parameters and the associated criterion of fit. The purpose of this section is to describe some simple methods for model validation. In other words, how do we know that a given model accurately models the output, and how can we say one model or parameter estimation technique is better or more appropriate than another? In this section, various methods of validation are presented including waveform prediction, parametric spectrograms and the whiteness and zero-mean properties of the innovations (otherwise known as one-step-ahead prediction errors).

4.1 Speech Reconstruction

Once a model is selected and its parameters estimated, it can be used to reconstruct the speech using the one-step ahead predicted waveform. This can then be compared with the original waveform using the sum-squared difference criterion. This report investigates one-step ahead prediction only. In other words, the output at time t is predicted using all input and output information up to and including $t-1$. The MatLab function *predict* [22] is used throughout. *predict* implements one of two methods.

- **Method 1.** For polynomial models, the predictor is calculated using equation 4, where G and H are defined in terms of polynomials.

$$\hat{y}(t|t-1) = H^{-1}(q,\theta)G(q,\theta)u(t) + [1 - H^{-1}(q,\theta)]y(t) \tag{49}$$

- **Method 2.** For state space models, one-step ahead prediction is calculated using a Kalman filter. This filter predicts the output using the following equations

$$\begin{aligned} y(t|t-1) &= \mathbf{C}x(t|t-1) + Du(t) &(50)\\ x(t|t-1) &= (\mathbf{A} - \mathbf{KC})x(t-1) + \mathbf{K}y(t-1) + (\mathbf{B} - \mathbf{KD})u(t-1) &(51) \end{aligned}$$

  $x(1|0)$ is the initial condition which need not be zero.

4.2 Spectral Estimation

Spectrograms of a waveform show the evolution in time of a spectrum. Typically, the waveform is divided into fixed-length and possibly overlapping frames. For each frame, the spectrum is calculated. These frame-based spectra are then compounded into a single chart such that the horizontal axis displays time, the vertical axis frequency, and the colour or grey-scale the energy at that particular frequency and time. In this report, *power* spectra only are considered. Two types of spectrogram are considered where λ is the estimated variance of the innovations.

- **Noise Model Spectrogram.** Spectra are defined using $\Phi_v(\omega) = \lambda|H(e^{j\omega T},\theta)|^2$.
- **Output Spectrogram.** Spectra are defined using $\Phi_y(\omega)$.

These are related via the following equations, where time-domain functions are converted to power spectra.

$$\begin{aligned} y(t) &= G(q,\theta)u(t) + H(q,\theta)e(t) \\ \Phi_y(\omega) &= |G(e^{j\omega T},\theta)|^2\Phi_u(\omega) + \lambda|H(e^{j\omega T},\theta)|^2 \end{aligned} \tag{52}$$

The noise model and output spectrograms differ because the former neglects input energy. This means that the two spectrograms may have very different energy ranges and hence may require different diagram scalings. Inspection of noise model and output spectrograms by eye provides useful information especially about how formants are modelled.

The difference between spectrograms from two different models can also be evaluated using spectral difference measures. Chapter 4 of [25] discusses many spectral difference measures related to speech processing. These spectral difference measures can be modified to take into account those perceptually important characteristics of speech. In this section, measures are not discussed in detail. Instead use is confined to the mean squared log spectral difference $d(S_1, S_2)^2$. This defines the difference between two spectra $S_1(e^{j\omega T})$ and $S_2(e^{j\omega T})$ as

$$d(S_1,S_2)^2 = \int_{-\pi}^{\pi} \left| \log S_1(e^{j\omega T}) - \log S_2(e^{j\omega T}) \right|^2 d\omega \tag{53}$$

The integration is approximated to a summation for discrete spectra. This difference measure between two spectra can be extended to the difference between two spectrograms by calculating this measure for each frame and then summing over all frames.

4.3 Innovations Whiteness Test

The quality of a model can be determined from its innovations sequence. An accurate model should produce a temporally white innovations sequence. Whiteness is tested using a statistical hypothesis test and conducted for each frame. Refer to [6] and [14] for details on statistical hypothesis tests, and Candy [5] for the particular whiteness test implemented during these experiments. (Candy uses this test in a Kalman Filter tuning application). The whiteness test adopts the null hypothesis H_0 that the prediction errors are white. The alternative hypothesis H_1 is that the prediction errors are not white. The test statistic used is the normalised biased autocovariance estimate and termed ρ. Under the null hypothesis, asymptotically for a large number of samples (such as 30 and above) the autocovariance estimate becomes normally distributed $\sim N(0, \frac{1}{N})$. The 95 % confidence limits are given by

$$I = \left[-\frac{1.96}{\sqrt{N}}, +\frac{1.96}{\sqrt{N}} \right]; \tag{54}$$

where N is the number of innovations. Under the null hypothesis, 95% of $\rho(i)$ for $i \neq 0$ estimates should therefore lie within this confidence interval. During these experiments the percentage of samples outside this confidence interval is calculated for each frame. If it exceeds 5 % then the null hypothesis is rejected at the 5 % significance. Note that the converse is not necessarily true. If the percentage is less than 5 %, then this does not necessarily imply that the null hypothesis is true; it just means that there is insufficient evidence to reject it.

4.4 Innovations Zero-Mean Test

For an accurate model the prediction error sequence should be zero-mean. During these experiments an hypothesis test is conducted on each frame to determine whether zero is a possible mean for the prediction error sequence. The test adopts the null hypothesis H_0 that the mean is zero. The alternative hypothesis H_1 is that the mean is not zero. The test statistic employed is

$$T = \frac{\bar{x} - \mu}{s} \tag{55}$$

where $\bar{x}$ is the sample mean, μ is the true mean and s is the sample standard deviation. Under the assumptions that the prediction errors are Gaussian distributed (it is necessary to make some assumption about the distribution of errors for this hypothesis test to be formulated), this test statistic is distributed according to a T-distribution. The p-value is of interest. The p-value is the probability of observing the given sample result under the assumption that the null hypothesis is true. And if the p-value is less than a significance level 5%, then the null hypothesis can be rejected at the 5% significance. However the converse is not necessarily true. If the p-value is greater than 5%, this does not necessarily imply that the null hypothesis is true; it just means that there is insufficient evidence to reject the null hypothesis. Therefore the p-value can be used to assess the zero-mean nature of the innovations sequence. Refer to [6] and [14] for more details on zero-mean hypothesis tests.

5 Experiments

Experiments are conducted on the phrase *"a small set of letters"*, spoken by an adult male. The speech waveform and laryngograph are obtained from the Eurom 0 database [11] and sampled at 16 kHz. The word *"small"* is particularly interesting because of the proximity of two of the frequency formants of the vowel in *"all"* which are between 3kHz and 4kHz and are only about 300 Hz apart (chapter 2 of [8]).

Due to non-stationarity, each speech waveform is divided into fixed-length, overlapping and windowed (hamming) analysis frames 15 ms (240 samples) in duration, shifted 7.5 ms (120 samples) each frame. Speech is assumed stationary within each frame. Overlap gives smoother parameter transitions and better quality reconstruction. Reconstruction employs the overlap-add method.

Due to the time delay for air to flow from the glottis to the microphone, it is necessary to align the laryngograph with the speech signal prior to analysis. From several tests, this delay was estimated to be 12 samples by considering the time difference between the peaks in the derivative of the laryngograph and peaks in the speech signal residue after inverse filtering the speech with an AR model. This was applied to the entire sentence uniformly. However there may be slight differences in delay across the sentence depending on whether the speaker moved his head significantly during speaking.

Using the laryngograph, the degree of voicing for each frame is determined and the frames divided into two classes: voiced and non-voiced frames, where non-voiced includes both unvoiced speech and silence. In all there are 87 voiced frames, and 83 non-voiced frames. Voiced frames are those frames which contain a laryngograph where derivatives are above a given threshold[3]. The voiced frames are analysed separately from the non-voiced frames.

Because of the difficulties in modelling the glottal waveform, input during voiced frames is approximated as a series of Rosenberg pulses [27], with closure synchronised to the dominant impulses in the derivative of the laryngographic data. Each pulse is a piecewise trigonometric function. Opening and closing phases of a glottal pulse are T_o and T_c respectively where T is the glottal period. During these experiments, $T_o = 0.4T$, $T_c = 0.16T$ and $\alpha = 1$ are a priori estimates which remain fixed during the analysis. Rosenberg pulses are described by

$$
\begin{aligned}
x(t) &= \frac{\alpha}{2}\left[1-\cos\left(\frac{\pi t}{T_o}\right)\right] && 0 \le t \le T_o \\
&= \alpha\cos\left[\frac{\pi(t-T_o)}{2T_c}\right] && T_o \le t \le T_o + T_c \\
&= 0 && T_o + T_c < t < T_p
\end{aligned}
$$

Input during non-voiced frames is set to zero. This glottal waveform is also framed and windowed as for the speech.

Noisy speech is prepared by adding white additive Gaussian noise to the clean waveform to give 20dB SNR averaged over the whole waveform. Throughout the experiments speech is *not* preemphasised prior to analysis.

5.1 Experimental Details

The previous section describes how the speech waveform *a small set of letters* is divided into fixed-length, overlapping frames. In this report, four sets of experiments are conducted.

Experiment Set 1 The purpose of these experiments is to compare the accuracy of different speech models using a common parameter estimation technique. Voiced

[3]The threshold for laryngograph derivatives is defined as 1000. These derivatives are typically -1000 to +5000 for the voiced *all* sound in *small* for example.

speech frames only are modelled because these voiced frames have deterministic input. Models considered are ARMAX, ARX, OE, ARMA and AR. PEM is used as the common parameter estimation technique. Experiments are conducted on both clean voiced speech and noisy voiced speech.

Experiment Set 2 The purpose of these experiments is to compare the accuracy of different parameter estimation techniques for common speech models. Again voiced speech frames only are modelled. Parameter estimation techniques contrasted are PEM, IV and 4SID. These are compared on a variety of models. Experiments are conducted on both clean voiced speech and noisy voiced speech.

Experiment Set 3 These experiments analyse non-voiced speech frames only. The purpose is to ascertain whether trends displayed during experiment sets 1 and 2 conducted on *voiced* speech also apply to *non-voiced* speech. Only ARMA and AR models are considered due to lack of deterministic input for non-voiced sounds. Only PEM and IV parameter estimation techniques are employed. 4SID techniques are not considered because the MatLab function for 4SID seems to require deterministic input. Experiments are conducted on both clean non-voiced speech and noisy non-voiced speech.

Experiment Set 4 The purpose of these experiments is to apply the speech modelling results from previous experiments to the practical problem of speech enhancement. Noisy speech as used in previous experiments are enhanced using a Kalman filter.

Information concerning the models analysed are given in table 3 together with the corresponding MatLab functions (where z denotes a two column vector with the first column as output and the second as input). $n = 12$ during the experiments, meaning models are either order 12 or 13.

For the ARMAX model, there are three model types: ARMAX1, ARMAX2 and ARMAX3. These models vary the order of the $A(q)$ polynomial, and the time delay between input and output. The purpose is to determine whether there is advantage in including a zero-sample delay or unit-sample delay in the polynomial model. (This zero-sample delay term corresponds to a non-zero $\mathbf{D}$ matrix in the state space model.) This is also done for all other models.

5.2 Algorithm Details for MatLab Functions

Default options for each MatLab function are used throughout the experiments unless otherwise stated.

- *armax* and *oe* both employ a robustified quadratic prediction error criterion minimised using an iterative Gauss-Newton algorithm. Initial parameter estimates are provided by a four-stage least-squares IV algorithm. There are differences between the two functions relating to the calculation of prediction errors and gradients.
- *arx* is solved by the least squares estimate from an overdetermined set of linear equations.
- *iv4* estimates the parameters of an ARX model using an approximately optimal four-stage instrumental variables (IV) procedure.
- *ar* estimates the parameters of an AR model for a scalar time series using variants of the least-squares method.

Model	Order	Input-Output Delay	MatLab functions	Error Criterion	MatLab *predict* method
ARMAX1	12	0	armax(z,[n (n+1) n 0])	PEM	1
ARMAX2	12	1	armax(z,[n n n 1])	PEM	1
ARMAX3	13	1	armax(z,[n (n+1) n 1])	PEM	1
N4SID1	12	1	n4sid(z,n,1,[],[0 1 0])	N4SID	2
N4SID2	12	1	n4sid(z,n,1,[],[0 1 1])	N4SID	2
N4SID3	12	0	n4sid(z,n,1,[],[1 1 1])	N4SID	2
N4SID4	12	0	n4sid(z,n,1,[],[1 1 0])	N4SID	2
ARX1	12	0	arx(z,[n (n+1) 0])	PEM	1
ARX2	12	1	arx(z,[n n 1])	PEM	1
ARX3	13	1	arx(z,[n (n+1) 1])	PEM	1
IV41	12	0	iv4(z,[n (n+1) 0])	IV	1
IV42	12	1	iv4(z,[n n 1])	IV	1
IV43	13	1	iv4(z,[n (n+1) 1])	IV	1
OE1	12	0	oe(z,[(n+1) n 0])	PEM	1
OE2	12	1	oe(z,[n n 1])	PEM	1
OE3	13	1	oe(z,[(n+1) n 1])	PEM	1
ARMA	12		armax(z(:,1),[n n])	PEM	1
AR1	12		ar(z(:,1),n,'ls','ppw')	PEM	1
AR2	12		ar(z(:,1),n,'ls','now')	PEM	1
AR3	12		ivar(z(:,1),n)	IV	1

Table 3: Models analysed during the experiments with the corresponding MatLab functions, parameter estimation techniques and the method used for one-step ahead prediction (refer to section 4.1), $n = 12$

- *ivar* estimates the parameters of an AR model using an approximately optimal choice of instrumental variable procedure.

- *n4sid* employs Van Overschee and De Moor's N4SID algorithm, which is a numerical algorithm for subspace state space system identification as described in section 3.3. Refer to the MatLab manual [22] and [39] for N4SID algorithm details.

5.3 Analysis of Results

The results from experiment sets 1 to 4 are analysed in terms of the prediction errors, the shape and consistency of the spectrograms and the perceptual quality of the one-step-ahead predicted waveforms during informal listening tests. During this report, particularly in the appendices, results are presented using abbreviations for the sake of brevity. In this section these abbreviations are explained. Table 3 lists the terminology for all models referred to throughout the report. Evaluation results for each model are presented using the following abbreviations:

- **predict** These are the one-step ahead prediction errors.

- **filter** These are the errors between original clean and Kalman filtered waveforms.

- **sse.** This is the median sum squared error between original clean and reconstructed (predicted or filtered) output waveform, averaged over all frames. The error is calculated for each frame *before* resynthesis using the overlap-add method. The median average is employed. This is to make the average more robust against spurious results which occur occasionally due to numerical instability. Refer to section 4.1.

- **whiteness.** This is a measure of the whiteness of the predicted or filtered errors, averaged over all frames. The whiteness is measured for each frame before resynthesis. Whiteness is the percentage of points in the biased autocovariance sequence which lie outside the 95 % confidence interval for a white sequence. Refer to section 4.3. The median average is employed.

- **zero-mean.** This is a measure of the zero-mean property of the predicted or filtered errors, averaged over all frames. *Zero-mean* is the p-value of the test statistic under the null hypothesis; large p-values indicate better zero-mean properties. Refer to section 4.4. The p-values are expressed as percentages. The median average is employed.

- **specDiff.** This is the mean squared log spectral difference between two model spectrograms calculated according to the method and measure described in section 4.2.

This terminology is adopted to explain tabular results throughout the report and appendices. Two types of graphical results are also presented.

- Parametric spectrograms of the noise model (the grey-scale is logarithmic).

- Graphs showing how the sum-squared one-step ahead prediction errors vary from one frame to the next, for various models and parameter estimation techniques.

6 Results and Discussion

Inaccurate modelling may be due to two causes: either a deficiency in the model structure or a deficiency in the parameter estimation technique. Results and discussion are separated into four parts. In the first, deficiency in the model structure is investigated for voiced speech. In the second, deficiency in the parameter estimation technique for voiced speech is investigated. The third part investigates the modelling of non-voiced speech. In the final section, the results from previous experiments are applied to the speech enhancement problem.

When comparing results and determining the accuracy of a model, three criteria are used. The first is the magnitude, whiteness and zero-mean of the prediction errors. The second is the shape of the noise model spectrograms. As explained by Burrows [4] it is also vital to consider the frequency distribution of modelling errors, because the ear is more perceptive to errors at certain frequencies than others and can take advantage of masking effects. In this report this is achieved by comparing noise model spectrograms with non-parametric spectrograms and a priori knowledge. Related to this is the third criterion: the predicted waveform. Informal listening tests and non-parametric spectrograms of the predicted waveform can provide important and perceptually relevant information concerning model accuracy. Often researchers pay attention to the first criterion only. This is particularly dangerous because models with small prediction errors may not give perceptually the best model.

6.1 Results Comparing Different Models For Voiced Speech

Inaccuracy due to a deficiency in model structure is investigated using *voiced* speech only. Results from the following experiments are compared: ARMAX1-3, ARX1-3, ARMA, OE1-3 and AR1-2. Terminology is explained in table 3. Because all these models employ a common parameter estimation technique, that is PEM, it can be assumed that differences in performance are due to differences in model structure only. In practise however, differences in PEM algorithm implementation between different model structures may contribute to model differences also. For example, PEM is sometimes iterative, sometimes non-iterative. However the effects of these algorithm differences are assumed small during a first analysis.

Figures 2 and 3 show how the sum-squared prediction errors per frame change with frame number for various models, when applied to the analysis of clean and noisy speech respectively. Tables 4 and 5 present a summary of the results. Models are ranked in order of increasing median average sum-squared prediction errors. Median whiteness and zero-mean results are also presented.

Rank	model	$\log_{10}$sse	whiteness (%)	zero-mean (%)
1	ARMAX1	5.75	1.26	46.51
2	ARX1	5.96	4.18	46.07
3	ARMA	6.07	1.26	93.37
4	AR2	6.15	4.18	93.00
5	OE1	7.74	28.87	5.27

Table 4: Models for clean voiced speech (in order of increasing prediction errors). Prediction errors are analysed for sum-squared error, whiteness and zero-mean, with median averages over all frames presented.

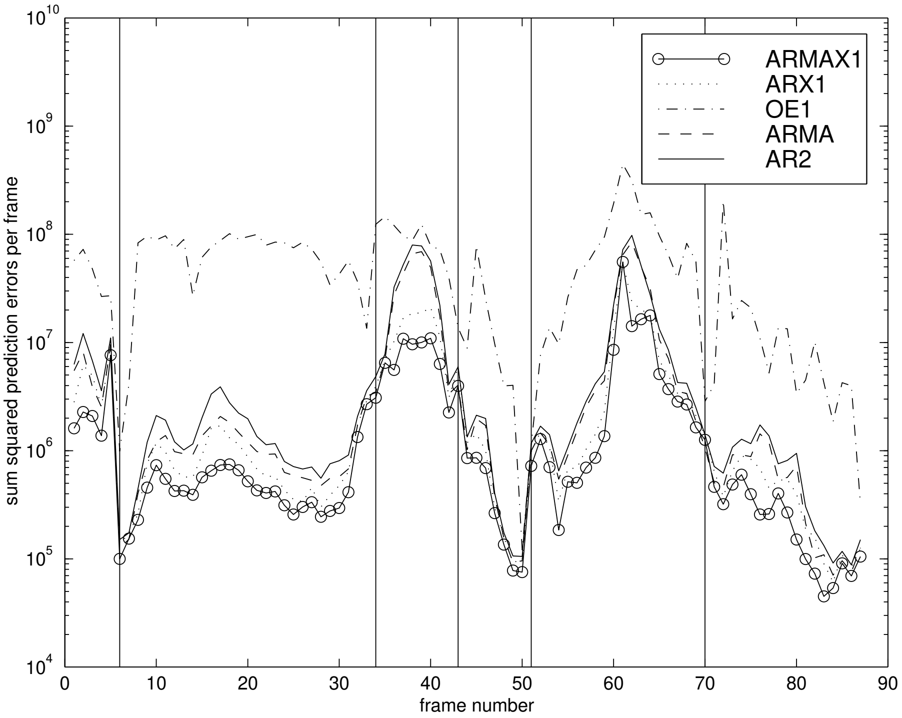

Figure 2: Prediction errors for PEM models for clean voiced speech. They represent the voiced sounds in “*a small set of letters*”.

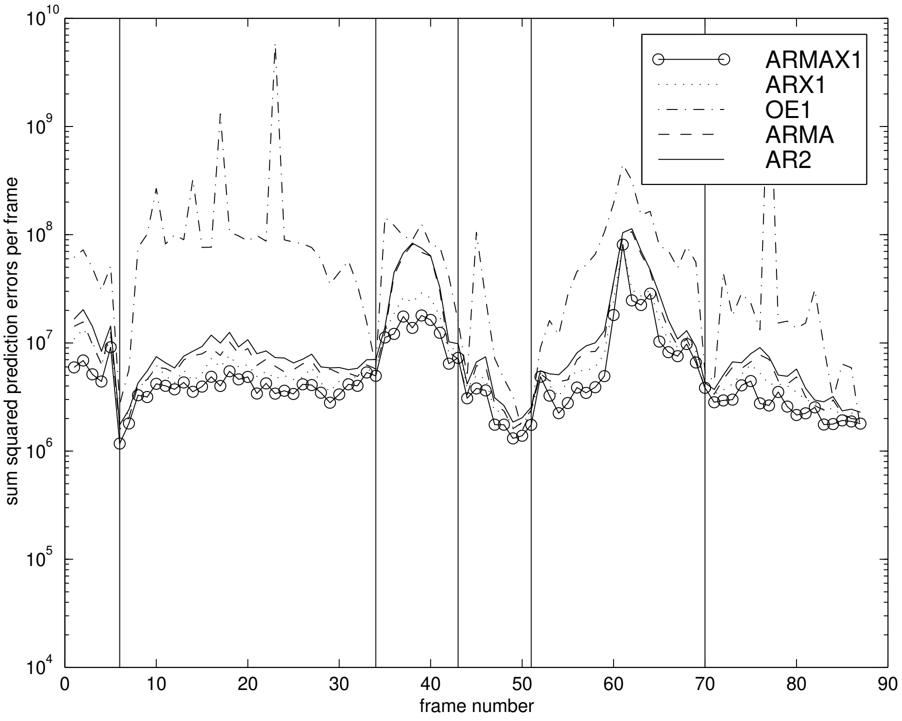

Figure 3: Prediction errors for PEM models for noisy voiced speech. They represent the voiced sounds in “*a small set of letters*”.

Rank	model	$\log_{10}$sse	whiteness (%)	zero-mean (%)
1	ARMAX1	6.60	1.26	44.31
2	ARX1	6.71	2.51	41.38
3	ARMA	6.79	0.84	92.71
4	AR2	6.87	2.93	94.18
5	OE1	7.74	25.10	2.73

Table 5: Models for noisy voiced speech (in order of increasing prediction errors). Prediction errors are analysed for sum-squared error, whiteness and zero-mean, with median averages over all frames

In this section only one model from each model set is presented. For ARMAX, ARX and OE this is the order 12 zero-sample delay model. For AR this is the model with smallest prediction errors and is the covariance estimate. Refer to Appendix 4 for more complete results. The following observations are made:

- In order of increasing prediction errors models are ARMAX1, ARX1, ARMA, AR2 and OE1. This is true for both clean and noisy speech. Both tabular and graphical results agree. Prediction errors generally increase in the presence of noise as expected.
- All models show prediction errors which are sufficiently white (less than 5 %) and sufficiently zero-mean (greater than 5 %), which means that the whiteness and zero-mean hypotheses are not rejected, except for the OE model. This indicates that the OE model does not model sufficient speech dynamics.
- ARMAX1, ARX1 and OE1 all model a deterministic glottal waveform input. ARMAX1 and ARX1 perform well, whereas OE1 performs poorly. Therefore the inclusion of deterministic input alone is not sufficient for small prediction errors.
- ARMAX1 and ARMA give a moving average (MA) structure to the noise, allowing both process and observation noise to be modelled. AR2 and ARX1 model process noise only, whereas OE1 models observation noise only. Results in clean conditions suggest that it is more important to model process noise than observation noise. Whether this is the same for noisy speech would depend on the signal-to-observation noise ratio. Smallest prediction errors are given when both process and observation noise are modelled together.
- Models with no deterministic input (ARMA and AR2) give largest p-values for the zero-mean hypothesis.
- Results were inconclusive as to whether a zero-sample delay or unit-sample delay reduced prediction errors or not. Refer to Appendix 4 for results. There are two possible reasons why a zero-delay between input and output may exist. Firstly, it is physically possible for a speech process because experiments consider a sampled representation of a continuous time process. Secondly, the glottal and speech waveforms may be misaligned.

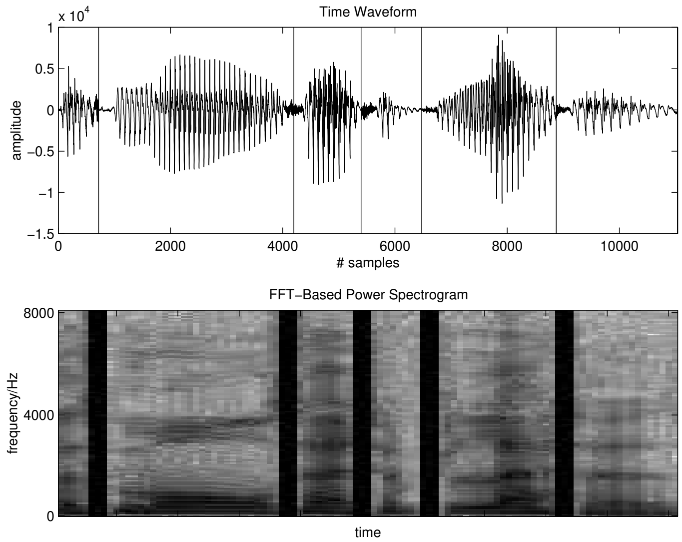

Figure 4: Waveform and MatLab FFT-based spectrogram for clean voiced speech. They represent the voiced sounds in "*a small set of letters*".

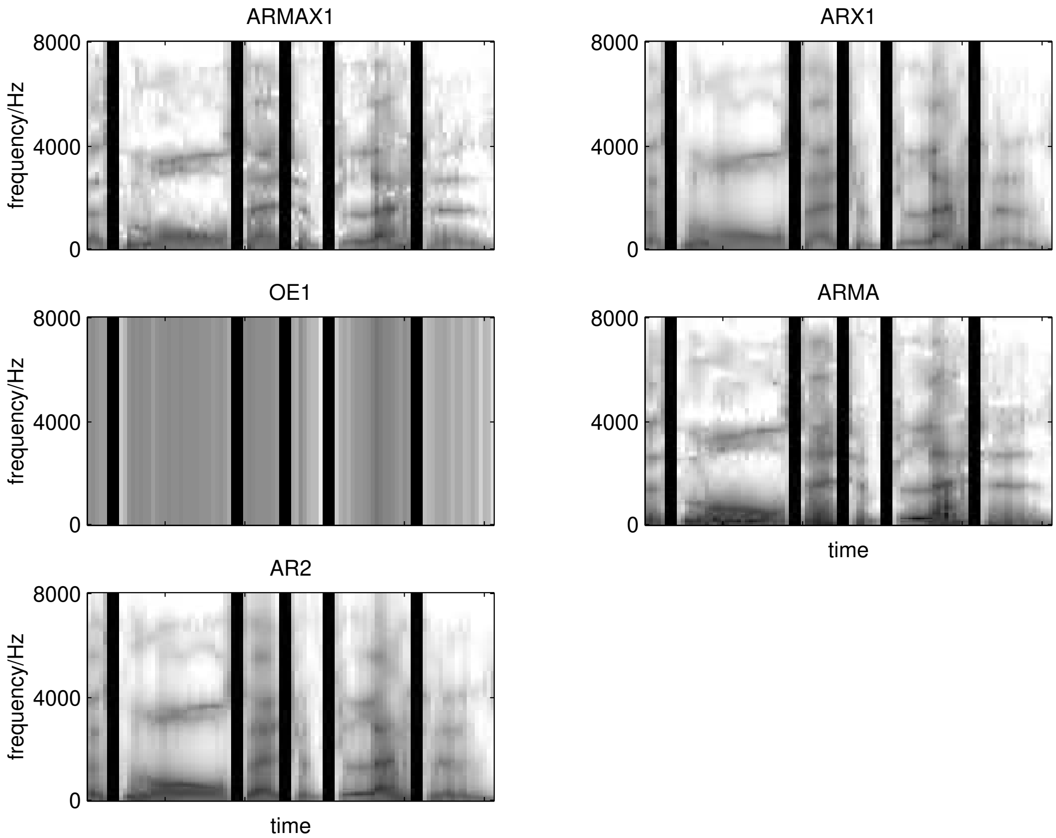

Figure 5: Noise model spectrograms comparing PEM models for clean voiced speech (same energy scaling). They represent the voiced sounds in "*a small set of letters*".

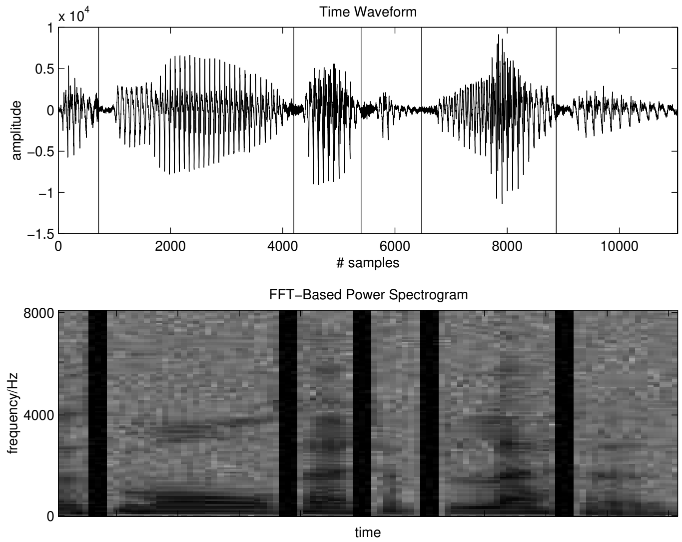

Figure 6: Waveform and MatLab FFT-based spectrogram for noisy voiced speech. They represent the voiced sounds in "*a small set of letters*".

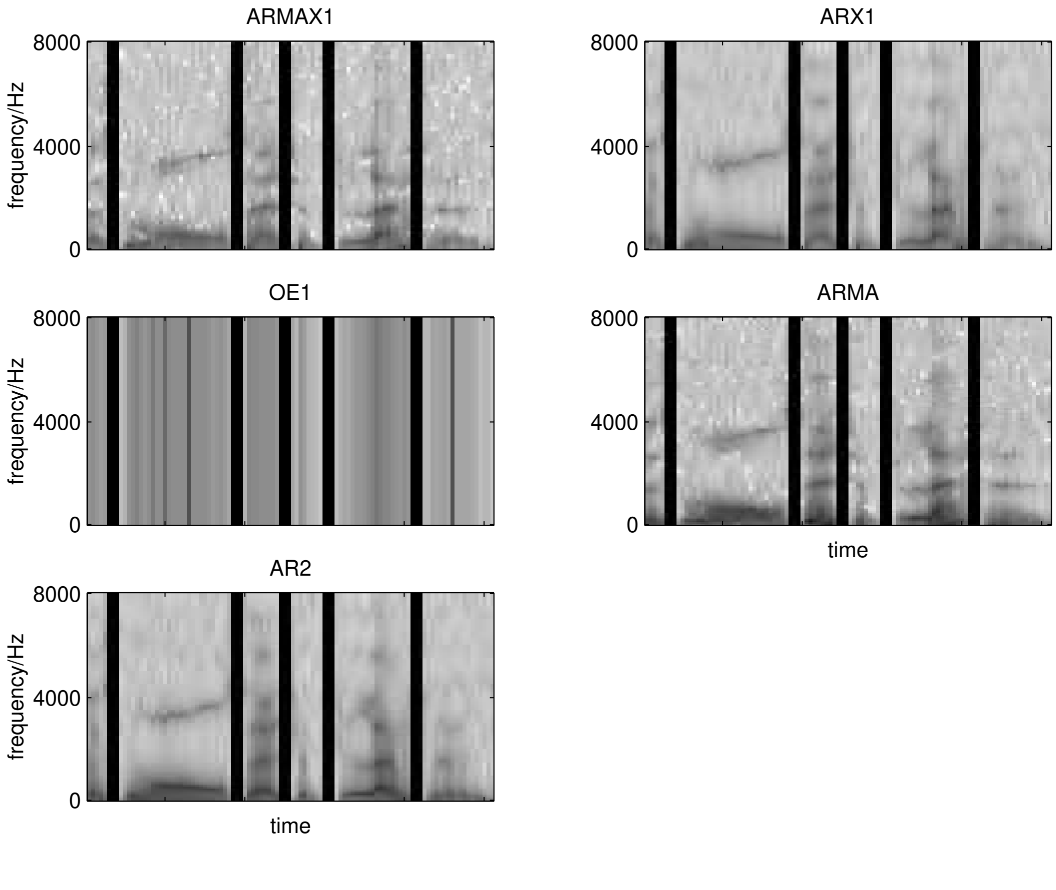

Figure 7: Noise model spectrograms comparing PEM models for noisy voiced speech (same energy scaling). They represent the voiced sounds in "*a small set of letters*".

Figures 5 and 7 show noise model parametric power spectrograms for clean and noisy speech. For the sake of comparison, time-domain waveforms and non-parametric output spectrograms are also shown in figures 4 and 6. All parametric spectrograms are on the same energy scale. The non-parametric spectrograms are determined using the MatLab *specgram* function and then converted to power spectra. These spectrograms are however on a different scaling to the parametric ones because they show output spectra rather than noise model spectra; refer to section 4.2 for more details. Dark regions indicate high energy. Dark continuous bands denote formant tracks which are evident on most voiced spectrograms. The following observations are made

- All methods except OE1 (where the noise model for each frame is flat) produce similar spectrograms. However ARMA and ARMAX1 are better at identifying the two close formant tracks between 3kHz and 4kHz in the *all* sound. This shows the advantage of a more general noise model.

- Although the OE1 noise model spectrogram shows no detail, the transfer function spectrogram does show spectral detail and formant structure.

- Among all the noise model spectrograms considered, ARMAX1 and ARX1 are most similar in the mean-squared log spectral difference sense. Numerical values are detailed in Appendix 4.

The one-step-ahead predicted audio waveforms from the different models are compared through informal listening tests and inspection of non-parametric spectrograms using the *xwaves* software. They give some very interesting results.

- In both clean and noisy conditions, ARMAX1 and OE1 waveforms contain high pitch background musical noise, and sometimes ARMA also. Moreover their non-parametric spectrograms contain relatively much high frequency energy. Further investigation is required but the reason for this musical noise is probably as given by Burrows [4]. She attributes the musical noise to the frame-to-frame fluctuation in high-frequency formant estimates. In turn this fluctuation is caused by the iterative parameter estimation algorithms converging on local optima rather than global optima, and these local optima vary between frames. (A side effect of these convergence problems is that high frequency formants have unnaturally narrow bandwidths.) In summary, the root cause of the musical noise seems to be due to a deficiency in the parameter estimation algorithms rather than the models themselves. This is supported by the fact that models with non-iterative PEM algorithms ie AR and ARX, contain less or no musical noise and their formant tracks appear much steadier at high frequencies; refer to figure 8. (Formant tracks are derived from the phase of the complex poles of the model.)

- Although the ARMAX1 model gives smallest prediction errors, the predicted waveform is perceptually poor. This demonstrates the danger of considering prediction error as the only goodness-of-fit criterion for a model.

- AR2 and ARX1 give predicted waveforms which perceptually differ little from the original waveform.

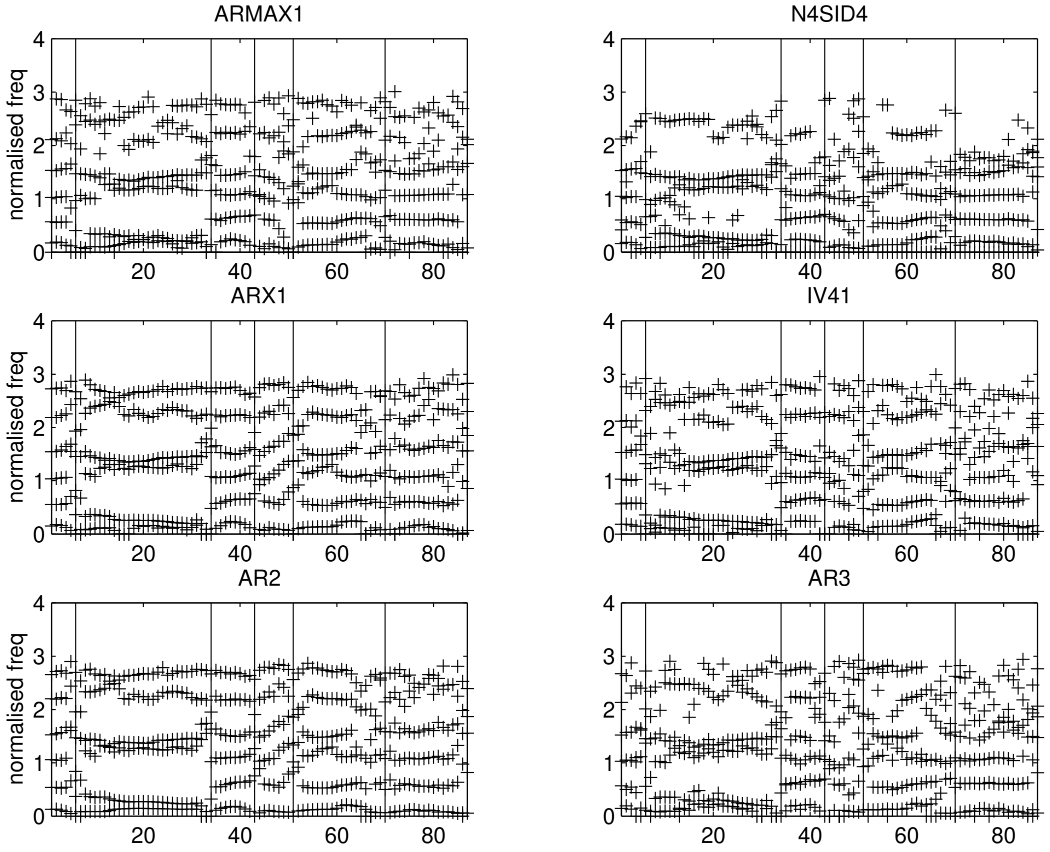

Figure 8: Formant frequency tracks comparing PEM models for clean voiced speech. They represent the voiced sounds in "*a small set of letters*".

Many large vocabulary speech recognition systems model speech using an AR model and extract LP (linear prediction) or PLP (perceptual linear prediction) coefficients. These results suggest that some useful information is lost by adopting such a model, particularly a loss of sharpness and detail in the frequency domain. Alternative models with more general noise models and/or with glottal waveform as input would be advantageous (although computation would also increase). However care must be taken when using iterative PEM algorithms as these may lead to musical noise.

6.2 Results Comparing Different Parameter Estimation Techniques for Voiced Speech

In the previous section, inaccuracy due to model deficiency was discussed. In this section inaccuracy due to a deficiency in the parameter estimation technique is discussed. Results from the following experiments are compared: ARMAX1 with N4SID4, ARX1 with IV41, and AR1-2 with AR3. Terminology is explained in table 3. These experiments allow comparison between PEM and 4SID, and PEM and IV techniques. These are compared according to the three criteria listed in section 6.1: prediction errors, noise model spectrograms and perceptual quality of the predicted waveform. Noise model spectrograms, time-domain waveforms and non-parametric spectrograms are shown in figures 9 to 11 in a similar manner to section 6.1. Parametric noise model spectrograms are all on the same energy scale, but the non-parametric output spectrogram is on a different energy scale; refer to section 4.2 for the reason.

6.2.1 PEM and 4SID

ARMAX and N4SID both model process noise, observation noise and glottal waveform input. Differences in performance are therefore primarily due to differences in parameter estimation techniques: PEM and 4SID respectively. Only results for ARMAX1 and 4SID4 are shown; these both correspond to an order 12 zero-sample delay model with zero initial conditions. More complete results are given in Appendix 4.

Figures 13 and 14 show how the sum-squared prediction errors per frame change for ARMAX1 and N4SID4 models for clean and noisy speech respectively. Table 6 presents a summary of these results.

Environment	model	$\log_{10}$sse	whiteness (%)	zero-mean (%)
Clean	ARMAX1	5.75	1.26	46.51
Clean	N4SID4	6.06	2.51	66.95
Noisy	ARMAX1	6.60	1.26	44.31
Noisy	N4SID4	6.75	2.09	61.04

Table 6: Comparing PEM and 4SID for clean and noisy voiced speech. Prediction errors are analysed for sum-squared value, whiteness and zero-mean, with median averages over all frames presented.

- ARMAX1 has smaller prediction errors than N4SID4. Both tabular and graphical results agree.
- ARMAX1 and N4SID4 both have prediction errors which are sufficiently white (less than 5%) and sufficiently zero-mean (greater than 5 %), which means that the whiteness and zero-mean hypotheses are not rejected.
- Both techniques show spectrograms with little spectral difference. Spectrograms for N4SID4 appear to show better formant tracks at low frequencies. This may be due to a different frequency-domain weighting of modelling errors.
- The ARMAX1 predicted waveform contains high pitch musical noise, whereas the N4SID4 waveform does not contain musical noise. A possible reason for this is as discussed in section 6.1.
- In order of increasing median prediction errors, models are N4SID3, N4SID2, N4SID4, N4SID1. Refer to Appendix 4 for complete results. These results show that it is better to allow non-zero initial conditions on the state vector.

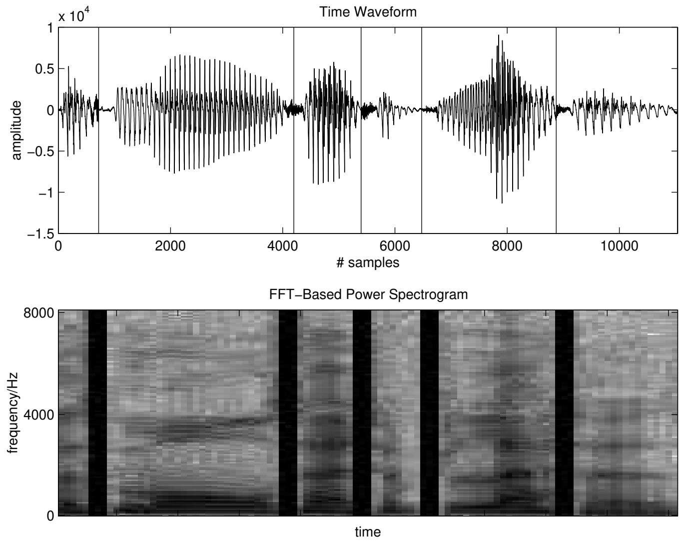

Figure 9: Waveform and MatLab FFT-based spectrogram for clean voiced speech. They represent the voiced sounds in "*a small set of letters*".

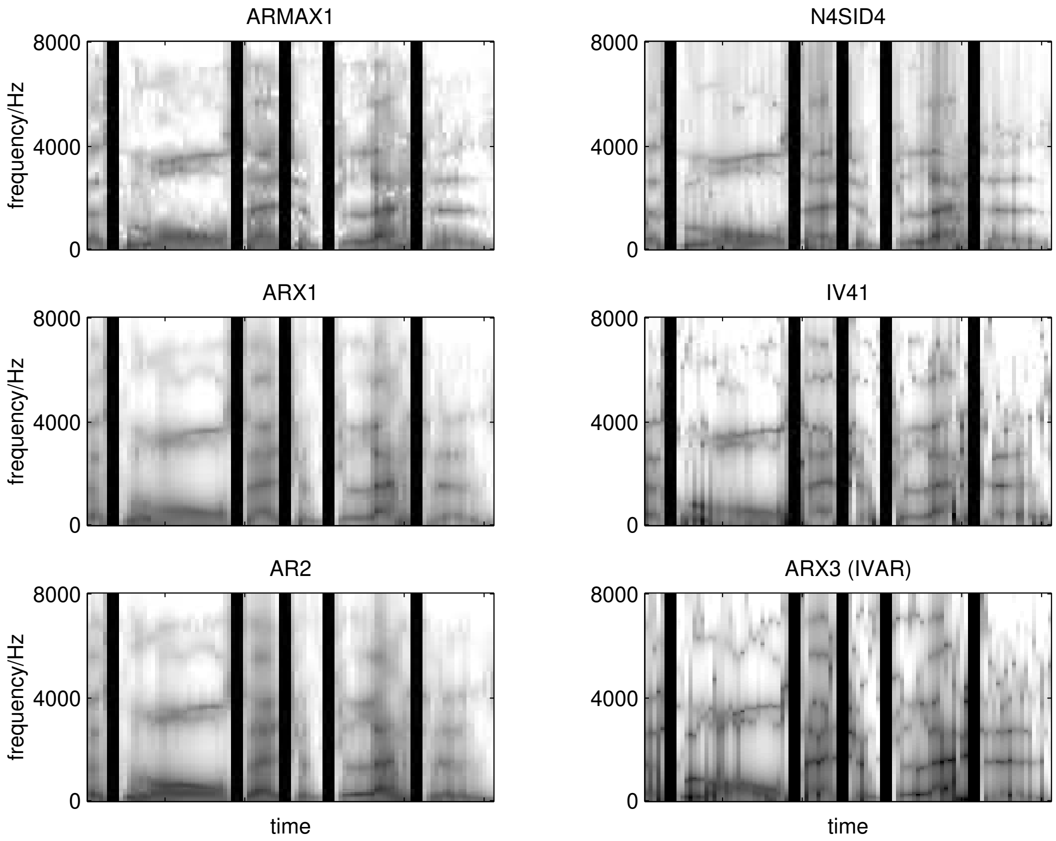

Figure 10: Noise model spectrograms comparing PEM, 4SID and IV estimation methods for clean voiced speech (same energy scaling). They represent the voiced sounds in "*a small set of letters*".

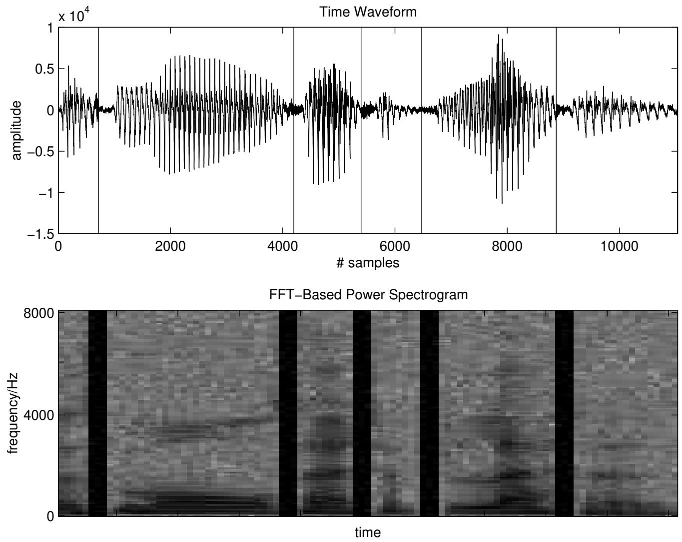

Figure 11: Waveform and MatLab FFT-based spectrogram for noisy voiced speech. They represent the voiced sounds in "*a small set of letters*".

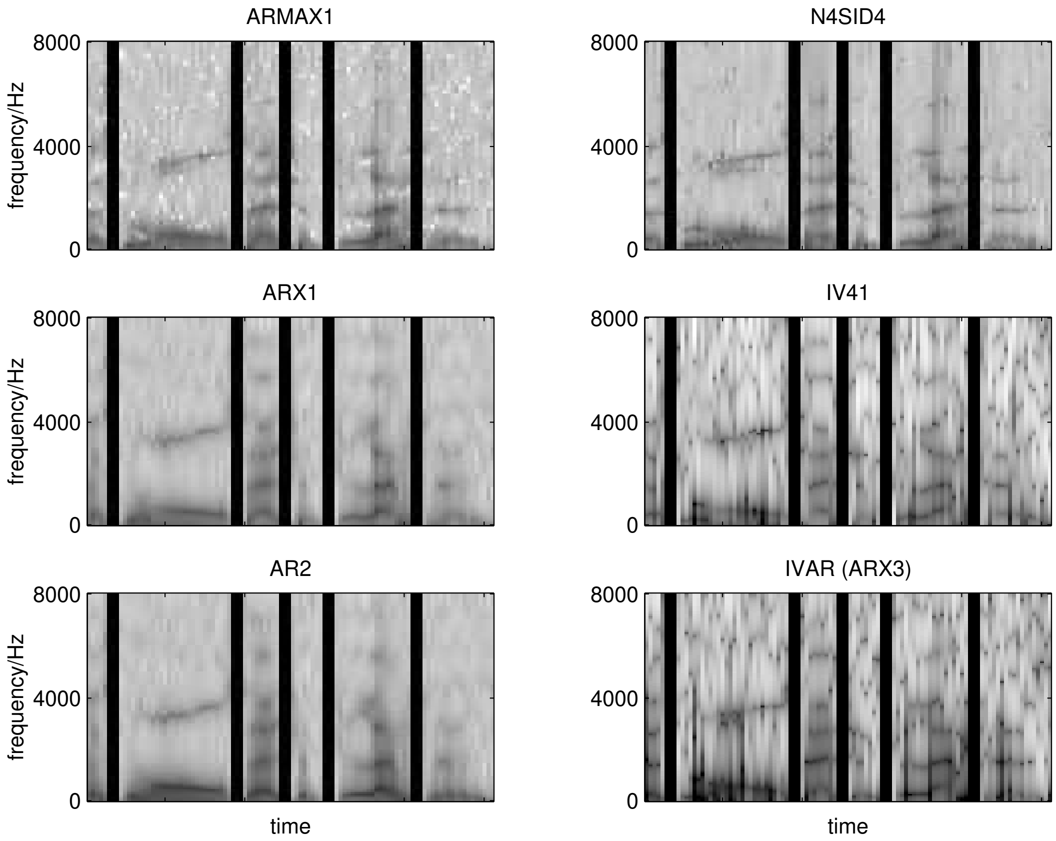

Figure 12: Noise model spectrograms comparing PEM, 4SID and IV estimation methods for noisy voiced speech (same energy scaling). They represent the voiced sounds in "*a small set of letters*".

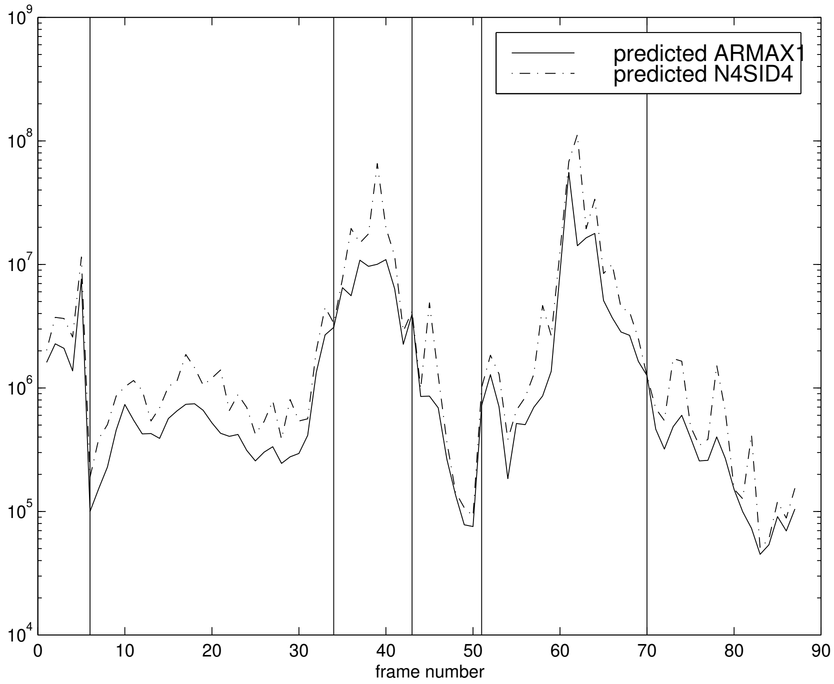

Figure 13: Prediction errors for PEM and 4SID estimation methods for an ARMAX model on clean voiced speech. They represent the voiced sounds in "*a small set of letters*".

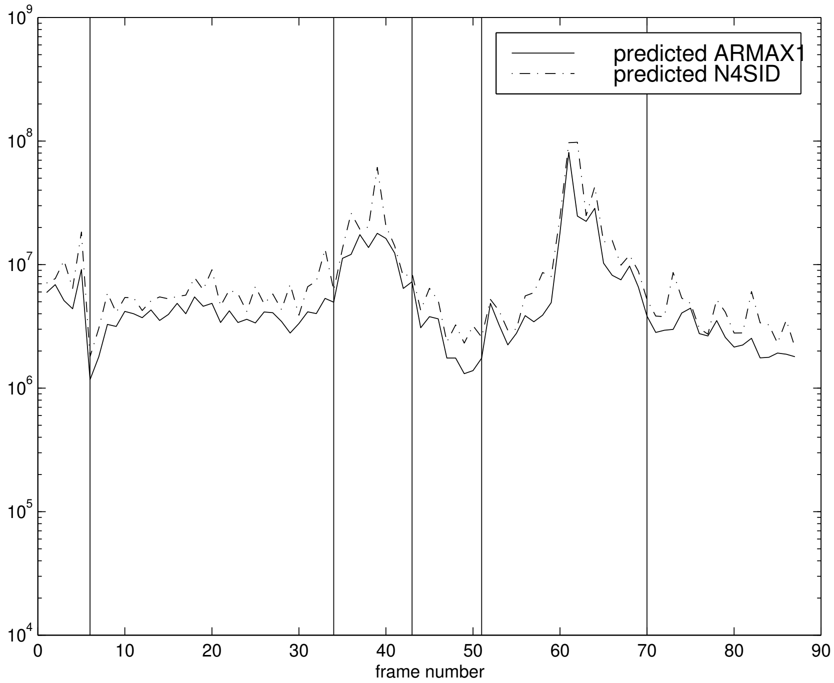

Figure 14: Prediction errors for PEM and 4SID estimation methods for an ARMAX model on noisy voiced speech. They represent the voiced sounds in "*a small set of letters*".

A detailed comparison between PEM and 4SID, especially concerning modelling errors and how these relate perceptually, will be left for a future report. Here some brief comments concerning the differences are given.

Iteration. 4SID methods are single-step and non-iterative. PEM methods for ARMAX, OE and ARMA models are iterative. Iterative methods suffer from the usual problems of convergence to local rather than global optima, slow or lack of convergence, sensitivity to initial conditions etc. Iterative algorithms seem to lead to high pitch musical noise in the one-step-ahead predicted waveforms as evident from these experiments.

Error Criterion. PEM methods minimise one-step-ahead prediction errors. 4SID methods however minimise a weighted combination of 1-step to i-step prediction errors, where i is the block size. In practice this means that 4SID methods distribute modelling errors in the frequency domain in a different manner to PEM, and tend to model low-frequencies better.

Numerical Stability. 4SID methods tend to use numerically robust algorithms such as the SVD.

State Space Basis. 4SID methods operate on a state space basis which is frequency-weighted balanced, whereas polynomial models operate on a companion parametrisation state space basis. Refer to section 2.8 for further details.

Bias. PEM gives unbiased estimates of parameters. 4SID may give solutions which are unbiased or show a little bias, depending on the actual 4SID algorithm implemented.

Instrumental Variables. 4SID methods remove the effects of noise by projecting the model onto subspaces orthogonal to the noise subspace and show similarities to instrumental variable (IV) methods [41]. 4SID therefore shows some properties of both PEM and IV methods.

6.2.2 PEM and IV

ARX and IV4 both use the same autoregressive with exogenous inputs model. IV4 estimates parameters using a 4-stage IV algorithm. ARX on the other hand uses a least-squares solution to an overdetermined set of linear equations. Differences in performance are therefore primarily due to differences in the parameter estimation technique.

AR1-2 and AR3 both use the same autoregressive model. Differences in performance are primarily due to differences in parameter estimation technique: AR1 uses a least-squares autocorrelation method, AR2 uses a least-squares covariance method, and AR3 uses an approximately optimal instrumental variable procedure to estimate the AR-part of a time series.

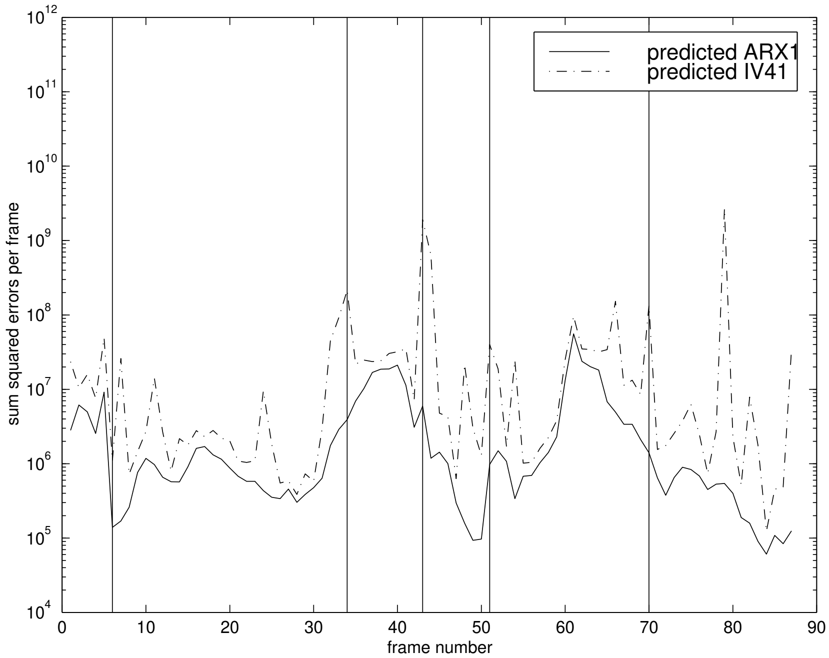

Figure 15: Prediction errors for PEM and IV estimation methods for an ARX model on clean voiced speech. They represent the voiced sounds in "*a small set of letters*".

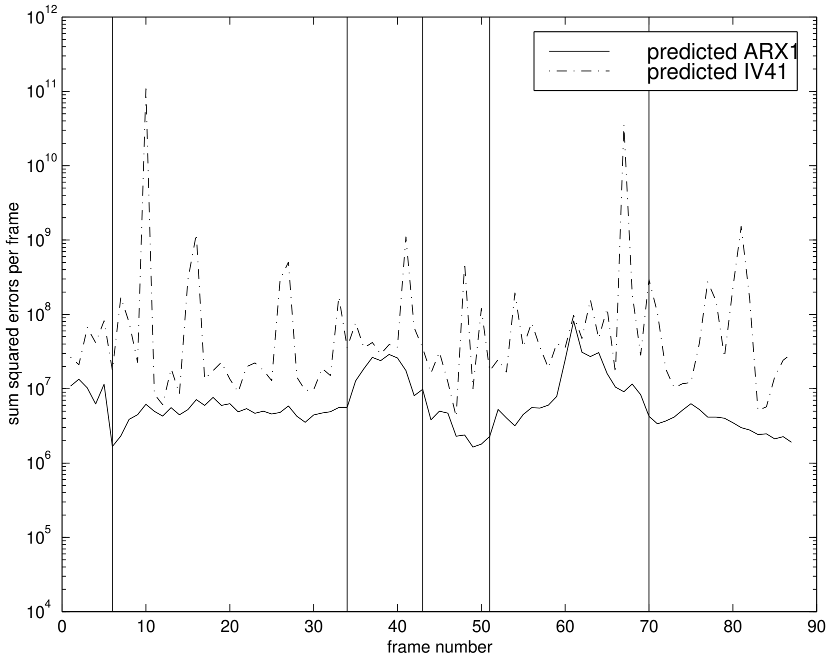

Figure 16: Prediction errors for PEM and IV estimation methods for an ARX model on noisy voiced speech. They represent the voiced sounds in "*a small set of letters*".

Environment	model	$\log_{10}$sse	whiteness (%)	zero-mean (%)
Clean	ARX1	5.96	4.18	46.07
Clean	IV41	6.57	13.39	72.43
Clean	AR2	6.15	4.18	93.00
Clean	AR3	6.79	9.62	95.05
Noisy	ARX1	6.71	2.51	41.38
Noisy	IV41	7.49	11.72	81.28
Noisy	AR2	6.87	2.93	94.18
Noisy	AR3	7.62	8.79	96.79

Table 7: Comparing PEM and IV for clean and noisy voiced speech. Prediction errors are analysed for sum-squared error, whiteness and zero-mean, with median averages over all frames presented.

Noise model spectrograms are displayed in figures 10 and 12. The variation of sum-squared prediction error with frame number is shown in figures 15 and 16 (for ARX) and figures 17 and 18 (for AR methods). Table 7 presents median average prediction errors, whiteness and zero-mean results in clean and noisy conditions.

- According to both tabular and graphical results, PEM methods give smaller prediction errors than IV methods. This is because PEM methods aim to minimise prediction errors, whereas IV methods aim to decorrelate prediction errors and not necessarily minimise them; refer to section 3.2.
- Although PEM and IV both give prediction errors which are sufficiently zero-mean (greater than 5 %), only PEM gives prediction errors which are sufficiently white (less than 5 %). This means that the whiteness hypothesis for IV is rejected. IV does not therefore model all the important dynamics in the speech.
- Results from AR1 and AR2 show that there is little difference between the performance of the autocorrelation and covariance methods. Refer to Appendix 4 for detailed results. Both methods are similar in that they use PEM and determine $A(q)$ according to

$$A(q) \quad = \quad \arg_{A(q)} \min \sum_{j=t_1}^{t_2} A(q)y(t) \tag{56}$$

where L is the frame length, n_a is the order and an output sequence $\{y(1),\ldots,y(L)\}$ is considered. However the autocorrelation and covariance methods differ with respect to the summation limits. For the covariance method, $t_1 = (n_a + 1)$, $t_2 = L$. For the autocorrelation method, $t_1 = 1$, $t_2 = (L + n_a)$, which is achieved by assuming samples before and after the frame are zero. Therefore the covariance and autocorrelation methods differ only with respect to end-effects. During these experiments, the frame size (240 samples) is relatively large, and the frame is hamming windowed prior to analysis, meaning end-effects are small. Therefore the two methods give similar results.

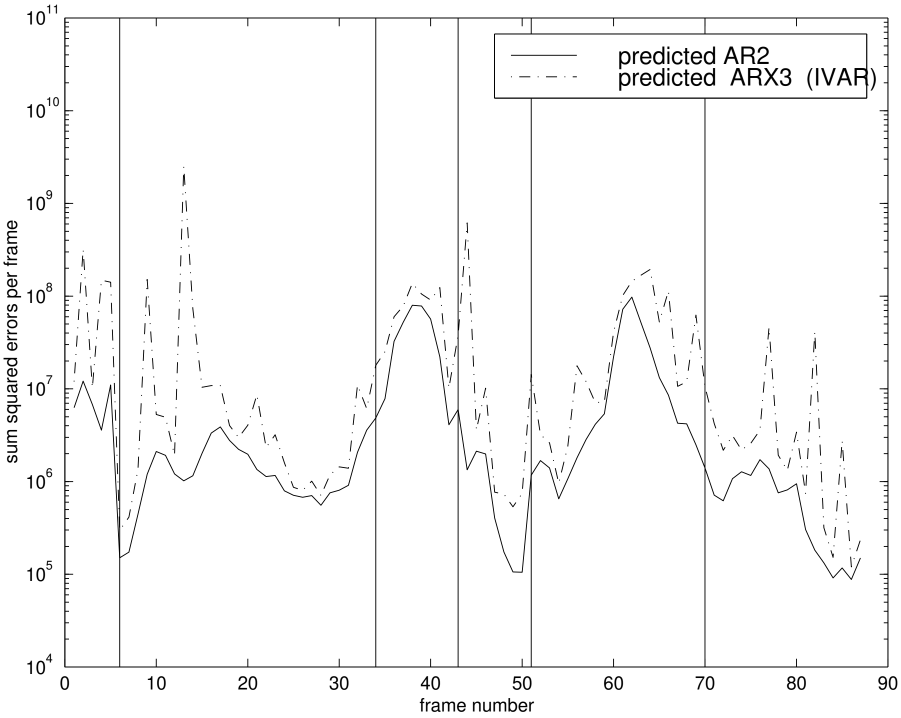

Figure 17: Prediction errors for PEM and IV estimation methods for an AR model on clean voiced speech. They represent the voiced sounds in "*a* sm*all* s*e*t *o*f l*e*tt*er*s".

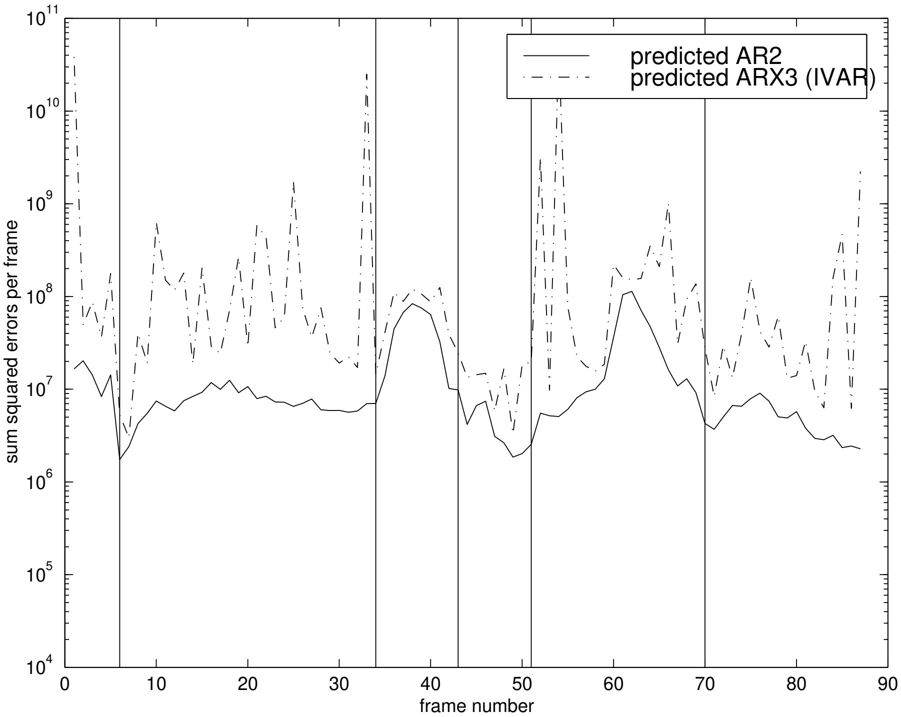

Figure 18: Prediction errors for PEM and IV estimation methods for an AR model on noisy voiced speech. They represent the voiced sounds in "*a* sm*all* s*e*t *o*f l*e*tt*er*s".

- There is appreciable differences between ARX and IV4 spectrograms, and AR1-2 and AR3 spectrograms as evident from the spectral differences listed in Appendix 4 and the spectrograms displayed in figures 10 and 12. The PEM methods produce much more consistent formant tracks than IV, although IV produces sharper tracks. Both ARX and AR1-2 have spectrograms which are more similar to ARMAX than IV4 and AR3 spectrograms.

- When considering the predicted waveforms for both clean and noisy speech, IV methods tend to produce waveforms with very large occasional "clicks", relatively high frequency energy noise, and high pitch musical noise. Further investigation is required to determine the reasons for these phenomena, but may be due to algorithm convergence problems also.

- When derived from noisy speech, the one-step-ahead predicted waveform for the AR2 model (covariance method) is "buzzy". This demonstrates the well-known fact that the AR covariance and autocorrelation methods are not robust to noise [17, 19, 29].

6.3 Experiments for Non-Voiced Speech

Non-voiced speech includes both unvoiced speech and silence. During the analysis of non-voiced speech, only two models are assessed: ARMA and AR. Other models are not possible due to an absence of deterministic input. ARMA and AR1-2 employ a PEM parameter estimation technique, whereas AR3 employs an instrumental variable technique.

A time-domain waveform, MatLab derived output power spectra and noise model spectrograms are illustrated in figures 19 and 20. Energy scalings for the parametric spectrograms are the same. However the scaling for the non-parametric spectrogram is different because it measures output rather than noise models; refer to section 4.2 for further details. Notice that the "t"s are stops and are preceded by a short silence. Figures 21and 22 show the variation in sum-squared prediction errors with frame number in clean and noisy conditions. Models are also presented in tables 8 and 9 in order of increasing median average prediction errors. Median average whiteness and zero-mean results are also given. More complete results and spectral differences are presented in Appendix 4.

rank	model	$\log_{10}$sse	whiteness (%)	zero-mean (%)
1	ARMA	6.33	0.84	47.79
2	AR2	6.36	2.09	38.33
3	AR1	6.36	2.09	38.47
4	AR3	7.36	7.53	69.18

Table 8: Results for clean non-voiced speech. Prediction errors are analysed for sum-squared error, whiteness and zero-mean, with median averages over all frames presented.

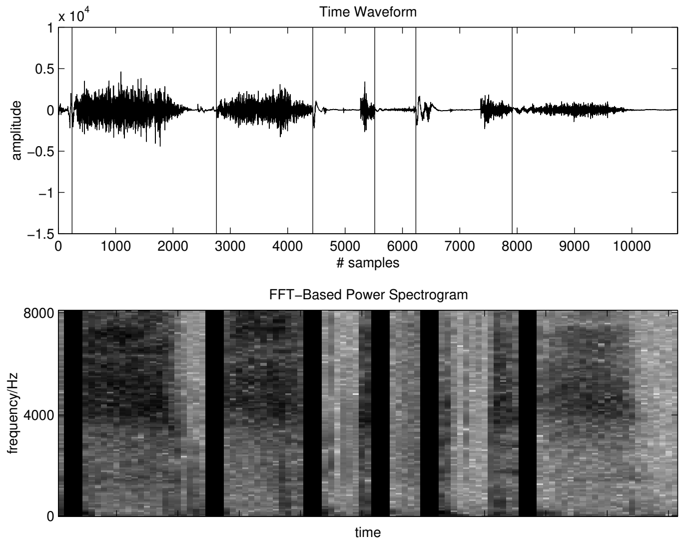

Figure 19: Waveform and MatLab FFT-based spectrogram for clean non-voiced speech. They represent the non-voiced sounds in "*a small set of letters*".

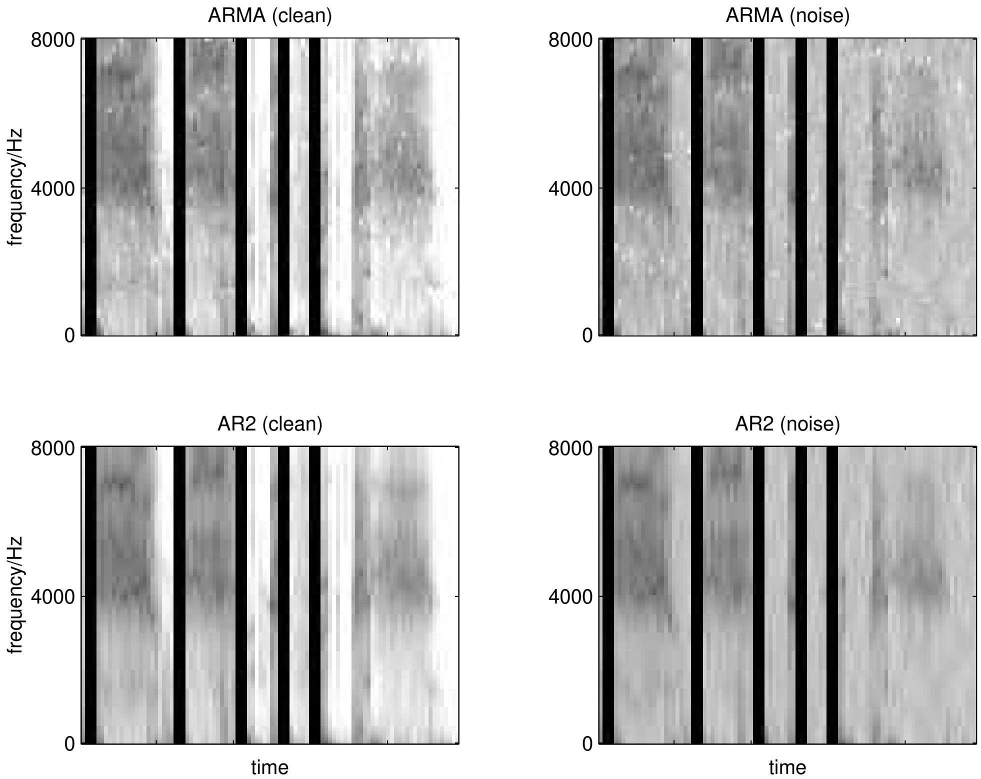

Figure 20: Noise model spectrograms for non-voiced speech (same energy scaling).They represent the non-voiced sounds in "*a small set of letters*".

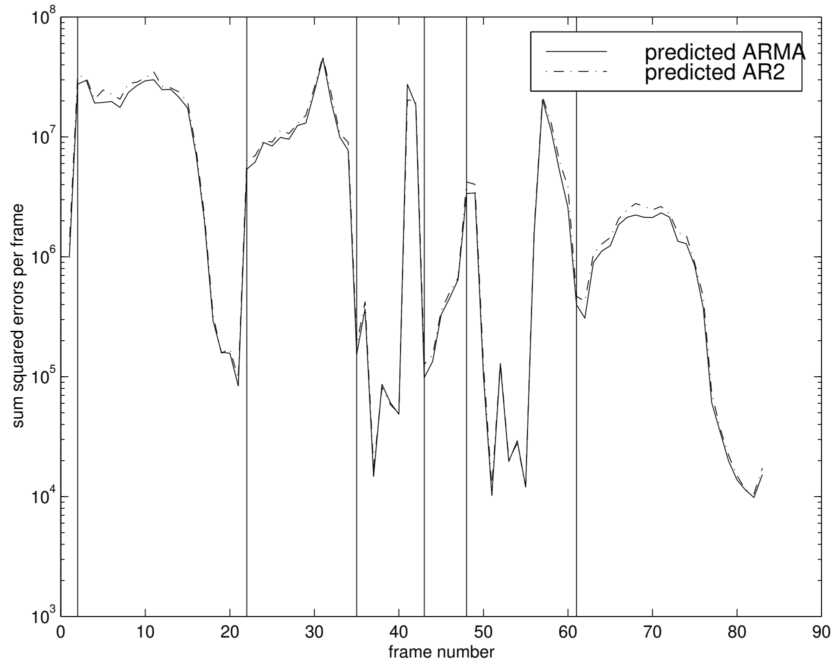

Figure 21: Prediction errors for clean non-voiced speech. They represent the non-voiced sounds in "*a small set of letters*".

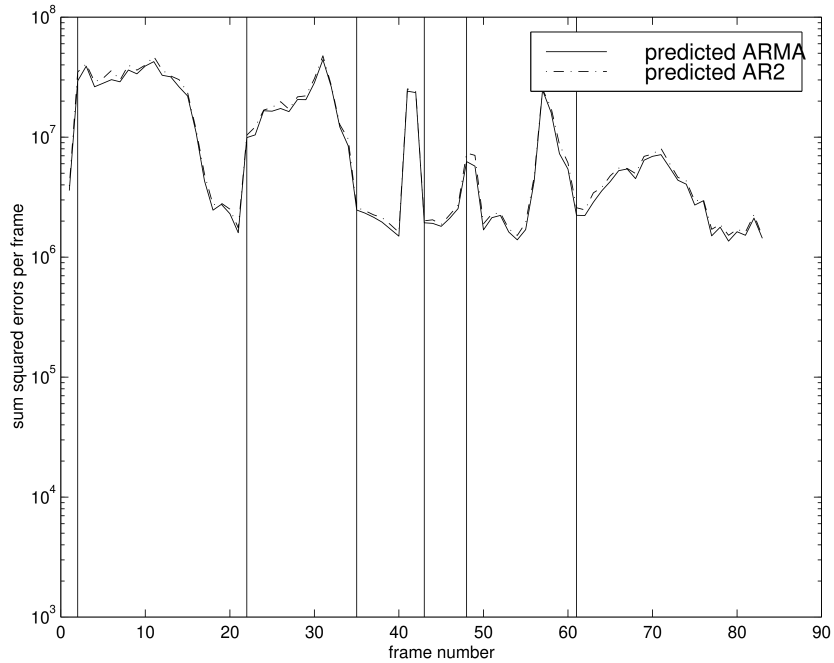

Figure 22: Prediction errors for noisy non-voiced speech. They represent the non-voiced sounds in "*a small set of letters*".

rank	model	$\log_{10}$sse	whiteness	zero-mean
1	ARMA	6.72	1.26	39.48
2	AR2	6.74	1.67	36.04
3	AR1	6.74	1.67	36.00
4	AR3	7.49	7.53	75.33

Table 9: Results for noisy non-voiced speech. Prediction errors are analysed for sum-squared error, whiteness and zero-mean, with median averages over all frames presented.

- Prediction errors for ARMA are slightly smaller than those for AR1-2 showing the benefits of a more general noise model. As expected, prediction errors for both models are smallest for silence regions!
- ARMA and AR1-2 all show prediction errors which are sufficiently white (less than 5%) and sufficiently zero-mean (greater than 5%), for the whiteness and zero-mean hypotheses not to be rejected. However the whiteness hypothesis is rejected for AR3. AR3 therefore does not model sufficient speech dynamics.
- Spectrograms generally show a lack of formant structure. Instead, for unvoiced speech, energy seems to exist as high-frequency energy bands.
- In terms of the predicted waveforms, there is very little perceptual difference between the ARMA and the AR2-3 model. The ARMA model does not seem to suffer from musical noise during non-voiced speech.
- Both ARMA and AR1-2 have similar shaped spectrograms, whereas AR3 has a large spectral difference. Refer to Appendix 4 for numerical values.
- AR1-2 shows smaller prediction errors than AR3. This agrees with previous results that PEM minimises prediction errors better than the IV method.

These results for non-voiced speech are in general agreement with those for voiced speech. When comparing results for voiced and non-voiced speech, errors for non-voiced speech are greater, especially in clean environments. This shows the difficulties in modelling the random nature of non-voiced speech. However it is more difficult to compare directly non-voiced and voiced speech modelling in noisy conditions. This is because non-voiced regions usually have lower local SNRs than voiced regions when artificial noise is added uniformly across the whole spoken phrase (as done in these experiments), because non-voiced regions are generally lower in energy.

6.4 Results For Speech Enhancement

Speech corrupted by additive white Gaussian noise is filtered to remove the effects of this noise using a Kalman filter. The algorithm is as follows.

1.	Select a model set and estimate model parameters using the noisy speech.
2.	Use the model parameters to initialise a Kalman filter.
3.	Filter the noisy speech using the Kalman filter.

In this section, results from previous experiments are applied to the practical problem of enhancing speech corrupted by 20 dB SNR additive white Gaussian output noise. Parameter estimates for models derived during previous experiments on the *noisy* speech are used to initialise Kalman filters on a frame-by-frame basis, which are then used to filter the noisy speech to produce noise-free estimates.

Kalman filters are discussed extensively in [2, 5, 10, 12]. Briefly, a Kalman filter assumes a state space model in forward innovations form.

$$\begin{aligned} x(t+1) &= \mathbf{A}x(t) + \mathbf{B}u(t) + \mathbf{K}e(t) \\ y(t) &= \mathbf{C}x(t) + \mathbf{D}u(t) + e(t) \end{aligned}$$

Given system matrix estimates $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$, process and observation noise covariance estimates $(\mathbf{Q}, \mathbf{R})$, and initial condition estimates on the state vector and state covariance matrix $(\mathbf{x_0}, \mathbf{P_0})$, the Kalman filter estimates an optimal state sequence $\{\hat{x}_0(t), \ldots, \hat{x}(N)\}$, where N is the number of samples. This is optimal in the sense that it is the minimum variance estimate. The output waveform is reconstructed using $\hat{y}_{KF} = \mathbf{C}x(t) + \mathbf{D}u(t)$. The accuracy of the reconstruction is measured using the sum-squared filtered (innovations) error, and the whiteness and zero-mean of these filtered errors.

During previous experiments, errors analysed are one-step-ahead prediction errors: $y(t) - \hat{y}(t|t-1)$. These are generated by the MatLab *predict* function. During enhancement, filter errors are also considered: $y(t) - \hat{y}(t|t)$. These are generated by the Kalman filter. Kalman filter errors are generally smaller than the one-step ahead prediction errors, because the Kalman filter uses additional information at time t to improve the estimate $\hat{y}$. Filter errors are up to one order of magnitude smaller approximately than prediction errors.

During these experiments two criteria are used for comparison. The first criterion is the magnitude, whiteness and zero-mean properties of the innovations. Refer to table 10 for analyses of prediction and filter errors for significant models from previous experiments. The second criterion is the perceptual quality of the filtered speech, through informal listening tests and inspection of filtered speech non-parametric output spectrograms. Spectrograms of clean, noisy and enhanced speech using a Kalman filter initialised using AR2 are show in figures 23, 24 and 25. These spectrograms are FFT-based and are generated using the *xwaves* software. The following observations are made

- Kalman filtering tends to reduce differences in performance between different models and parameter estimation techniques. Whereas models have a clear hierarchy of ARMAX, ARX, ARMA, AR and OE for the one-step-ahead-predictor, no such clear hierarchy exists for the filter. Why the use of the filter reduces the variation of reconstruction error among different model types is not known. OE however is a noticeable exception. This is because the OE model assumes no process noise, and therefore gives poor estimates of system matrices for the Kalman filter.

- The Kalman filter tends to decrease the zero-mean and whiteness properties of the residual. The reason for this is not yet known. When the whiteness exceeds 5 % and the zero-mean reduces to below 5 %, then the whiteness and zero-mean hypotheses respectively can be rejected at the 5 % significance level.

- When listening to the filtered waveforms, Kalman filtering generally removes the observation noise well, while introducing little or no perceptual distortion into the signal. The filter removes much of the broadband high frequency noise especially. Musical noise evident in the one-step-ahead predicted waveform is no longer present in the filtered waveform. Exceptions are OE1 and AR3 models which agree with numerical results listed in table 10.

Small reconstruction errors suggests possible applications of the technique to speech coding. Although in its present set-up the Kalman filter is suited to the removal of *white* observation noise only, the state space can be augmented to allow *coloured* observation noise to be considered also [10].

Environment	Model	$\log_{10}$(sse)		whiteness (%)		zero-mean (%)	
		predict	filter	predict	filter	predict	filter
clean	ARMAX1	5.75	5.09	1.26	4.18	46.51	39.97
clean	N4SID3	5.95	4.95	3.35	8.37	57.76	39.58
clean	ARX1	5.96	5.16	4.18	7.11	46.07	36.22
clean	IV41	6.57	5.04	13.39	12.97	72.43	24.24
clean	OE1	7.74	7.72	28.87	29.71	5.27	6.53
clean	ARMA	6.07	5.03	1.26	5.44	93.37	81.53
clean	AR2	6.15	5.16	4.18	10.04	93.00	77.96
noisy	ARMAX1	6.60	6.21	1.26	5.44	44.31	30.00
noisy	N4SID3	6.72	6.19	2.09	6.69	55.84	48.05
noisy	ARX1	6.71	6.25	2.51	6.69	41.38	23.80
noisy	IV41	7.49	6.31	11.72	4.18	81.28	38.38
noisy	OE1	7.74	7.74	25.10	25.94	2.73	3.54
noisy	ARMA	6.79	6.25	0.84	6.69	92.71	46.00
noisy	AR2	6.87	6.27	2.93	7.95	94.18	50.44

Table 10: Results when analysing clean and noisy voiced speech. Prediction and Kalman filter innovations are analysed for sum-squared error, whiteness and zero-mean, with median averages over all frames considered.

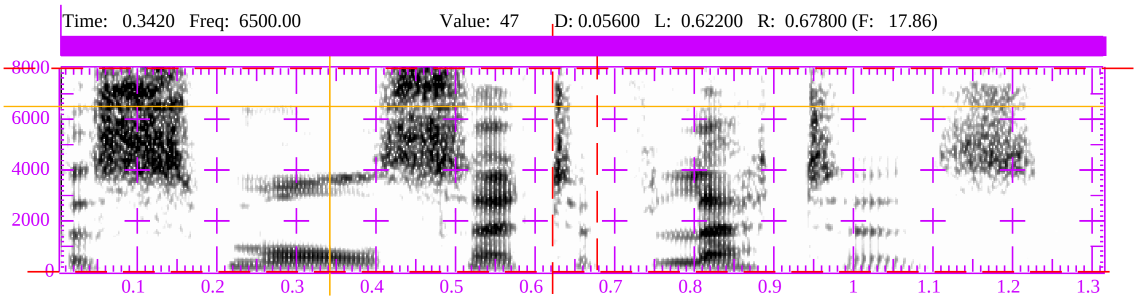

Figure 23: *xwaves* spectrogram for *a small set of letters* (clean speech).

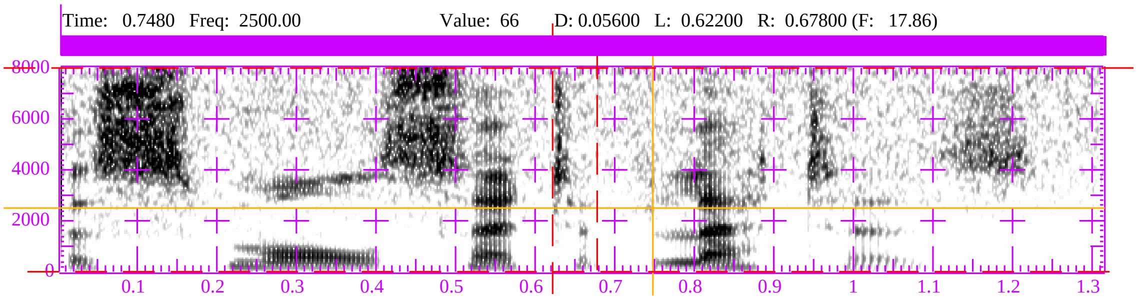

Figure 24: *xwaves* spectrogram for *a small set of letters* (noisy speech with global 20dB SNR). Noise is additive white Gaussian.

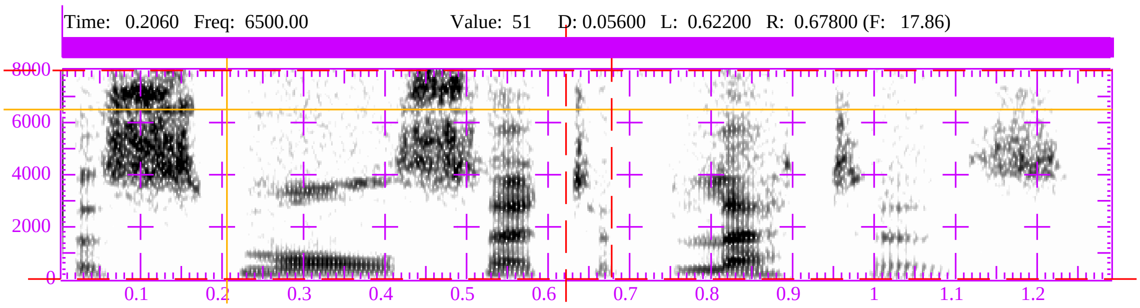

Figure 25: *xwaves* spectrogram for *a small set of letters* (enhanced speech using a Kalman filter initialised with the AR2 model from the noisy speech).

7 Future Work

Results from these experiments show that knowledge of the human speech production and speech perception processes is helpful in model set selection and parameter estimation respectively. Future research will investigate speech production and speech perception in more detail to determine how such knowledge can best be used in the system identification problem. On the frame level, particular modelling challenges include modelling of rapidly changing sounds such as plosives, glides and onsets. Multi-modelling approaches for the separate modelling of different sound classes may be of interest. On the phoneme level, particular modelling challenges include the introduction of time constraints on the time evolution of formants and spectrograms to reduce musical noise, perhaps by means of a high-level speech model. Particular perception challenges include weighting modelling errors in the frequency domain so as to take advantage of masking, the cochlea filter-banks perception model, and the variation of the sensitivity of the ear to frequency.

Therefore it is also necessary to understand the frequency distribution of modelling errors for PEM, IV and 4SID, and the factors which affect these distributions. Factors include but may not be limited to prefiltering, input signal power spectra, prediction horizon and sampling rate [21, 42]. The effects and merits of preemphasis, a popular speech preprocessing procedure, in relation to these techniques and various models should also be understood.

Of particular interest are the 4SID algorithms. These have the advantage that they model an ARMAX model, allow state vector initial conditions to be readily estimated as part of the N4SID algorithm, allow noise or insignificant signal modes to be rejected in a better fashion, and do not appear to suffer from musical noise problems. This may be due to the fact that these algorithms tend to locate global optima better than PEM methods, or that they are non-iterative or that they use a different weighting on prediction errors.

The Kalman filter (initialised using PEM, IV or 4SID) has been successfully used for the speech enhancement problem. However the performance of the filter appears to be fairly insensitive to the choice of initialisation algorithm, provided it is reasonable. Further investigation is required. Also 4SID parameter estimation techniques have an elegant relationship with Kalman filter theory [35, 39], which would be worthwhile considering further in the speech enhancement problem.

Further investigation into the effects of improved speech modelling on the enhancement of coloured or non-stationary noise, speech coding, feature extraction and speech recognition is also worthwhile.

8 Conclusions

It is useful to consider both polynomial and state space models within a common state space framework because this makes explicit the assumptions which models make regarding process noise, observations noise and the structure of system matrices. Results can be divided into those related to the model, and those related to the parameter estimation technique.

Models in decreasing performance are ARMAX, ARX, ARMA, AR and OE where performance is measured in terms of the prediction errors and noise model spectrograms. Glottal waveform input, a general noise model and non-zero initial conditions on the state vector all reduce prediction errors. The lack of a general noise model (such as for AR, ARX and OE) increases prediction errors and reduces the clarity of spectrograms. However there is little evidence to suggest whether modelling is improved by having a zero-sample time delay between input and output.

PEM, 4SID and IV parameter estimates are compared. PEM methods give smaller prediction errors than 4SID and IV. But 4SID predicted waveforms have better perceptual qualities and noise model spectrograms show more detail at low frequencies (this may be due to a different frequency domain weighting of modelling errors). Also 4SID has the advantage that it is non-iterative, can readily estimate non-zero initial conditions and lends itself in a better manner to low order approximation and rejection of insignificant signal modes or noise modes by simple truncation of system matrices. IV tends to give sharper spectrograms but spectra show inconsistency from one frame to the next. Iterative PEM algorithms and IV seem to suffer from high pitch musical noise effects.

Results for non-voiced and voiced speech in clean conditions are similar, except prediction errors are generally larger for non-voiced speech due to the random nature of the system input.

Speech modelling is applied to the practical problem of speech enhancement in white additive Gaussian observation noise, using a Kalman filter initialised from noisy speech. The filter produces similar reconstruction errors for all models and parameter estimation techniques except IV techniques and the OE model (probably due to unreasonable initial parameter estimates due to the deficiency of the OE model). The enhanced speech shows good perceptual qualities and lack of distortion except for OE and IV.

Acknowledgements Gavin Smith is grateful for funding and support from the Schiff Foundation, Cambridge University. In addition we appreciate the use of the Kalman filter software of Zoubin Ghahramani *www.gatsby.ucl.ac.uk/˜zoubin/*.

9 Appendices

9.1 Appendix 1 – Polynomial Models in State Space Form

Consider a combined deterministic-stochastic system in forward innovations form, with state space order 3. The derivation herein can readily be extended to higher orders, but is kept to order 3 for simplicity.

$$\begin{aligned} x(t+1) &= \mathbf{A}x(t) + \mathbf{B}u(t) + \mathbf{K}e(t) \\ y(t) &= \mathbf{C}x(t) + \mathbf{D}u(t) + e(t) \end{aligned}$$

For the transition equation define matrices as

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ -a_3 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad \mathbf{K} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix}$$

For the observation or measurement equation define matrices as

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \quad \mathbf{D} = b_0$$

Now combine transition and observation equations together

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \end{bmatrix} = \begin{bmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ -a_3 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} u(t) + \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} e(t)$$

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \end{bmatrix} = \begin{bmatrix} -a_1 x_1(t) + x_2(t) \\ -a_2 x_1(t) + x_3(t) \\ -a_3 x_1(t) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} u(t) + \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} e(t)$$

From the observation equation

$$x_1(t) = y(t) - b_0 u(t) - e(t)$$

Substituting this into the above equation and eliminating $x_2(t)$ and $x_3(t)$ by repeated substitution yields the following:

$$\begin{aligned} \begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \end{bmatrix} &= \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ -a_2 & -a_3 & 0 \\ -a_3 & 0 & 0 \end{bmatrix} \begin{bmatrix} y(t) & - & b_0 u(t) & - & e(t) \\ y(t-1) & - & b_0 u(t-1) & - & e(t-1) \\ y(t-2) & - & b_0 u(t-2) & - & e(t-2) \end{bmatrix} \\ &+ \begin{bmatrix} b_1 & b_2 & b_3 \\ b_2 & b_3 & 0 \\ b_3 & 0 & 0 \end{bmatrix} \begin{bmatrix} u(t) \\ u(t-1) \\ u(t-2) \end{bmatrix} + \begin{bmatrix} k_1 & k_2 & k_3 \\ k_2 & k_3 & 0 \\ k_3 & 0 & 0 \end{bmatrix} \begin{bmatrix} e(t) \\ e(t-1) \\ e(t-2) \end{bmatrix} \end{aligned}$$

Note that the matrices are Toeplitz. Expanding the first line and substituting for $x_1(t+1)$ on the left-hand side of the equation gives

$$
y(t+1) - b_0 u(t+1) - e(t+1) = \begin{bmatrix} -a_1 & -a_2 & -a_3 \end{bmatrix} \begin{bmatrix} y(t) & - & b_0 u(t) & - & e(t) \\ y(t-1) & - & b_0 u(t-1) & - & e(t-1) \\ y(t-2) & - & b_0 u(t-2) & - & e(t-2) \end{bmatrix}
$$

$$
+ \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} \begin{bmatrix} u(t) \\ u(t-1) \\ u(t-2) \end{bmatrix}
$$

$$
+ \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \begin{bmatrix} e(t) \\ e(t-1) \\ e(t-2) \end{bmatrix}
$$

Reorganising this gives

$$
\begin{bmatrix} 1 & a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} y(t+1) \\ y(t) \\ y(t-1) \\ y(t-2) \end{bmatrix} = \begin{bmatrix} b_0 & (b_1 + b_0 a_1) & (b_2 + b_0 a_2) & (b_3 + b_0 a_3) \end{bmatrix} \begin{bmatrix} u(t+1) \\ u(t) \\ u(t-1) \\ u(t-2) \end{bmatrix}
$$

$$
+ \begin{bmatrix} 1 & a_1 + k_1 & a_2 + k_2 & a_3 + k_3 \end{bmatrix} \begin{bmatrix} e(t+1) \\ e(t) \\ e(t-1) \\ e(t-2) \end{bmatrix}
$$

This can be written in familiar polynomial form as

$$
A(q)y(t) = B(q)u(t) + C(q)e(t)
$$

where polynomials are defined as

$$
\begin{aligned}
A(q) &= 1 + \textstyle\sum_{i=1}^{3} a_i \\
B(q) &= b_0 + \textstyle\sum_{i=1}^{3} (b_i + b_0 a_i) \\
C(q) &= 1 + \textstyle\sum_{i=1}^{3} (k_i + a_i)
\end{aligned}
$$

In many cases of polynomial modelling, $\mathbf{D} = 0$, which means $b_0 = 0$. This simplifies the expression for the $B(q)$ polynomial especially.

9.2 Appendix 2 – The DOA Model

A brief overview of the DOA methods are given. One of the reasons for this is because DOA methods can be seen as a subset of the more general 4SID methods. Consider the state space system in block matrix form as described in section 2.6.

$$\begin{aligned}\mathbf{X}_f &= \mathbf{A}^i\mathbf{X}_p + \mathbf{\Delta}_i^d\mathbf{U}_p + \mathbf{\Delta_i}^w\mathbf{E}_p \\ \mathbf{Y}_p &= \mathbf{\Gamma}_i\mathbf{X}_p + \mathbf{H}_i^d\mathbf{U}_p + \mathbf{H}_i^w\mathbf{E}_p \\ \mathbf{Y}_f &= \mathbf{\Gamma}_i\mathbf{X}_f + \mathbf{H}_i^d\mathbf{U}_f + \mathbf{H}_i^w\mathbf{E}_f\end{aligned}$$

The DOA model is a state space system with no deterministic input and no process noise. These equations can therefore be simplified further, noting that $\mathbf{H}_i^w$ reduces to the identity matrix and $\mathbf{\Delta}_\mathbf{i}^\mathbf{w}$ reduces to zero.

$$\begin{aligned}\mathbf{X}_f &= \mathbf{A}^i\mathbf{X}_p \\ \mathbf{Y}_p &= \mathbf{\Gamma}_i\mathbf{X}_p + \mathbf{E}_p \\ \mathbf{Y}_f &= \mathbf{\Gamma}_i\mathbf{X}_f + \mathbf{E}_f\end{aligned}$$

Consider now the sample correlation matrix of $\mathbf{Y}_p$, where the Hermitian transpose ^{H} is used because matrices may contain complex components.

$$\begin{aligned}\mathbf{Y}_p\mathbf{Y}_p^H &= \mathbf{\Gamma}_i\mathbf{X}_p(\mathbf{\Gamma}_i\mathbf{X}_p + \mathbf{E}_p)^H \\ &\quad +\mathbf{E}_p(\mathbf{\Gamma}_i\mathbf{X}_p + \mathbf{E}_p)^H\end{aligned}$$

If innovations are assumed white and uncorrelated with the states and the innovations, then $\mathbf{E}_p\mathbf{E}_p^H = \sigma_e^2\mathbf{I}$ and $\mathbf{X}_p\mathbf{E}_p^H = 0$ respectively, where σ_e^2 is the variance of the innovations process or observation noise. Therefore

$$\mathbf{Y}_p\mathbf{Y}_p^H = \mathbf{\Gamma}_i\mathbf{X}_p\mathbf{X}_p^H\mathbf{\Gamma}_i^H + \sigma_e^2\mathbf{I}$$

This final equation can be recognised as the fundamental equation for the signal processing algorithms to solve the direction-of-arrival problem, and in estimating damped sinusoids in noisy conditions [20, 26, 34]. These include algorithms such as ESPRIT, MUSIC, MODE and WSF.

Furthermore the correlation matrix can be decomposed such that the observability matrix $\mathbf{\Gamma}_i$ has Vandermonde structure. This corresponds to a uniform and linear array for the direction-of-arrival problem.

$$\begin{aligned}\mathbf{Y}_p &= \begin{bmatrix} 1 & 1 & \dots & 1 \\ e^{z_1} & e^{z_2} & \dots & e^{z_p} \\ e^{2z_1} & e^{2z_2} & \dots & e^{2z_p} \\ \vdots & \vdots & & \vdots \\ e^{(i-1)\ z_1} & e^{(i-1)\ z_2} & \dots & e^{(i-1)\ z_p} \end{bmatrix} \begin{bmatrix} x(0) & x(1) & \dots & x(j-1) \end{bmatrix} \\ &= \mathbf{\Gamma}_{v,i}\mathbf{X}_{v,p} + \mathbf{E}_p\end{aligned}$$

where i is the block size and the $_v$ subscript denotes matrices relating to the Vandermonde structure. The output can be written as a sum of decaying exponentials in additive observation noise.

$$y(t) = \sum_{i=1}^{p} \mathbf{X}_{v,p}(i,t)e^{z(i-1)} + e(t)$$

$z_i = (j\omega_i - \alpha_i)$ is a system pole where ω_i and α_i are the frequency and one-sided bandwidth (or damping rate) respectively. $\mathbf{X}_{v,p}$ is a state sequence which contains the amplitudes of the decaying exponentials. In this example, $\mathbf{C} = [1, 1, \ldots, 1]$ and $\mathbf{A} = diag(e^{z_i}, e^{z_{2i}}, \ldots, e^{z_p})$

Therefore traditional DOA methods operate on the block matrix form of a state space system, where process noise and deterministic inputs are zero. DOA methods can therefore be considered a subset of the more general 4SID methods. Now, different parameter estimation algorithms will be briefly considered.

Parameter Estimation Algorithms

There are a variety of subspace algorithms to solve the DOA problem. Large sample realizations of the maximum likelihood (ML) estimate are given in table 11 where noise is assumed white and temporally and spatially white. However these ML objective functions are highly nonlinear and multimodal. Therefore assumptions and approximations can be made to these criteria which results in many of the popular DOA algorithms such as MUSIC, ESPRIT, WSF and MODE. Refer to [31, 32, 40] for derivations and [29] for an overview.

Criterion	Optimisation	Citation
1a	$\theta = arg_\theta\ min\|\|\mathbf{Y}_p - \mathbf{\Gamma}_{v,i}(\theta)\mathbf{X}_{v,p}(\theta)\|\|_F^2$	[40]
1b	$\theta = arg_\theta\ max\ tr\{\mathbf{\Pi}_{\mathbf{\Gamma}_{v,i}}(\theta)\hat{\mathbf{R}}_{YY}\},\ \mathbf{X}_{v,p} = \mathbf{\Gamma}_{v,i}^{\dagger}\mathbf{Y}_{v,p}$	[40]
2	$\theta = arg_\theta\ min\ tr\{\mathbf{\Gamma}_{v,i}(\theta)^H\hat{\mathbf{U}}_n\hat{\mathbf{U}}_n^H\mathbf{\Gamma}_{v,i}(\theta)\hat{\mathbf{R}}_{XX}(\theta)\}$	[31]

Table 11: DOA Maximum Likelihood Estimation Criteria

Matrices are defined as

- $\hat{\mathbf{R}}_{XX}(\theta) = \frac{1}{N}\mathbf{X}_{v,p}\mathbf{X}_{v,p}^H$ is the sample correlation matrix of the state sequence.
- $\hat{\mathbf{R}}_{YY} = \frac{1}{N}\mathbf{Y}_p\mathbf{Y}_p^H$ is the sample correlation matrix of the output.
- $\hat{\mathbf{U}}_n$ is a matrix whose columns are eigenvectors of $\hat{\mathbf{R}}_{YY}$
- $\mathbf{\Pi}_{\mathbf{\Gamma}_{v,i}}(\theta) = \mathbf{\Gamma}_{v,i}(\theta)\mathbf{\Gamma}_{v,i}(\theta)^{\dagger} = \mathbf{\Gamma}_{v,i}\{\mathbf{\Gamma}_{v,i}^H(\theta)\mathbf{\Gamma}_{v,i}(\theta)\}^{-1}\mathbf{\Gamma}_{v,i}^H(\theta)$ is the projection matrix onto the column space of $\mathbf{\Gamma}_{v,i}$.

Criterion 1a attempts to align the column space of the extended observability matrix with the column space of the data, and is therefore a subspace fitting problem. Joint optimisation over both $\mathbf{\Gamma}_{v,i}$ and $\mathbf{X}_{v,p}$ is required. As with most 4SID problems, this optimisation is separable. Setting $\mathbf{X}_{v,p} = \mathbf{\Gamma}_{v,i}^{\dagger}\mathbf{Y}_p$ means that the optimisation can be separated into two stages which gives criterion 1b. Finally criterion 2 attempts to make the extended observability matrix column space as orthogonal as possible to the output sample correlation matrix noise subspace.

9.3 Appendix 3 – Projection Theory

The first step in subspace algorithms is to project a matrix onto a given subspace. Two projectors are used depending on the algorithm, termed the orthogonal and oblique projectors. † denotes the Moore-Penrose inverse, and $^\perp$ the orthogonal space. Consider matrices $\mathbf{S}$ and $\mathbf{R}$ with row spaces $\mathcal{S}$ and $\mathcal{R}$ respectively.

- $\mathbf{A}/\mathfrak{S}$ is the *orthogonal* projection of $\mathbf{A}$ onto $\mathcal{S}$.

$$\begin{aligned}\mathbf{A}/\mathfrak{S} &= \mathbf{A}\mathbf{S}^T(\mathbf{S}\mathbf{S}^T)^\dagger\mathbf{S} \\ &= \mathbf{A}\mathbf{\Pi}_\mathbf{S}\end{aligned}$$

- $\mathbf{A}/_R\mathfrak{S}$ is the *oblique* projection of $\mathbf{A}$ onto $\mathcal{S}$ along $\mathcal{R}$.

$$\mathbf{A}/_\mathbf{R}\mathfrak{S} = (\mathbf{A}/\mathfrak{R}^\perp)(\mathbf{S}/\mathfrak{R}^\perp)^\dagger\mathbf{S}$$

When $\mathbf{R} = 0$ or when $\mathcal{R}$ is orthogonal to $\mathcal{S}$, the oblique projection reduces to an orthogonal projection.

$$\mathbf{A}/_\mathbf{R}\mathfrak{S} = \mathbf{A}/\mathfrak{S}$$

9.4 Appendix 4 – Further Experimental Results

Experiments on Clean Voiced Speech

Model	$\log_{10}$(sse)		whiteness (%)		zero-mean (%)	
	predict	filter	predict	filter	predict	filter
ARMAX1	5.75	5.09	1.26	4.18	46.51	39.97
ARMAX2	5.86	5.05	1.26	5.44	30.83	15.18
ARMAX3	5.87	5.05	1.26	5.02	28.28	19.46
N4SID1	6.10	5.27	2.51	6.28	47.87	35.82
N4SID2	6.04	5.10	2.93	7.11	44.27	28.21
N4SID3	5.95	4.95	3.35	8.37	57.76	39.58
N4SID4	6.06	5.31	2.51	6.28	66.95	48.39
ARX1	5.96	5.16	4.18	7.11	46.07	36.22
ARX2	6.02	5.13	3.77	8.37	31.76	13.26
ARX3	6.02	5.13	3.77	7.95	39.39	18.29
IV41	6.57	5.04	13.39	12.97	72.43	24.24
IV42	6.53	5.05	12.97	12.55	53.40	13.59
IV43	6.52	4.96	12.13	12.97	48.64	13.60
OE1	7.74	7.72	28.87	29.71	5.27	6.53
OE2	7.75	7.73	28.87	29.29	2.73	2.51
OE3	7.70	7.68	28.87	28.87	5.78	7.72
ARMA	6.07	5.03	1.26	5.44	93.37	81.53
AR1	6.19	5.20	4.60	10.04	93.00	78.24
AR2	6.15	5.16	4.18	10.04	93.00	77.96
AR3	6.79	5.18	9.62	15.90	95.05	69.64

Table 12: Results when analysing clean voiced speech. Prediction and Kalman filter innovations are analysed for sum-squared error, whiteness and zero-mean, with median averages over all frames presented.

model	specDiff						
	ARMAX1	N4SID3	ARX1	IV41	OE1	ARMA	AR2
ARMAX1		9.46	9.39	10.86	9.81	10.81	10.13
ARMAX2	9.60	9.78	9.72	10.88	9.98	10.81	10.12
ARMAX3	9.79	9.90	9.86	10.89	10.06	10.82	10.17
N4SID1	9.54	9.42	9.63	10.87	9.91	10.81	10.13
N4SID2	9.45	9.19	9.54	10.87	9.85	10.81	10.14
N4SID3	9.46		9.46	10.86	9.76	10.82	10.17
N4SID4	9.50	9.13	9.56	10.87	9.85	10.81	10.15
ARX1	9.39	9.46		10.86	9.73	10.82	10.14
ARX2	9.61	9.69	9.43	10.87	9.87	10.83	10.16
ARX3	9.65	9.72	9.50	10.87	9.89	10.83	10.17
IV41	10.86	10.86	10.86		10.87	11.13	10.92
IV42	11.18	11.18	11.18	11.28	11.18	11.32	11.21
IV43	10.73	10.73	10.72	11.06	10.75	11.05	10.79
OE1	9.81	9.76	9.73	10.87		10.85	10.26
OE2	10.10	10.07	10.06	10.90	9.80	10.88	10.39
OE3	9.85	9.80	9.77	10.87	8.98	10.85	10.27
ARMA	10.81	10.82	10.82	11.13	10.85		10.76
AR1	10.08	10.13	10.09	10.91	10.22	10.76	9.29
AR2	10.13	10.17	10.14	10.92	10.26	10.76	
AR3	11.25	11.25	11.25	11.38	11.26	11.33	11.24

Table 13: Results when analysing clean voiced speech. $\log_{10}$ mean squared spectral differences between noise model spectrograms.

Experiments on Noisy Voiced Speech

Model	$\log_{10}$(sse)		whiteness (%)		zero-mean (%)	
	predict	filter	predict	filter	predict	filter
ARMAX1	6.60	6.21	1.26	5.44	44.31	30.00
ARMAX2	6.62	6.23	1.26	5.86	33.54	20.17
ARMAX3	6.63	6.23	1.26	5.86	32.20	21.70
N4SID1	6.78	6.25	2.09	6.69	58.12	35.57
N4SID2	6.74	6.20	2.09	6.69	48.88	32.85
N4SID3	6.72	6.19	2.09	6.69	55.84	48.05
N4SID4	6.75	6.24	2.09	6.28	61.04	43.16
ARX1	6.71	6.25	2.51	6.69	41.38	23.80
ARX2	6.73	6.25	2.51	7.11	31.88	20.56
ARX3	6.73	6.26	2.93	7.53	36.81	20.65
IV41	7.49	6.31	11.72	4.18	81.28	38.38
IV42	7.59	6.29	11.30	4.60	72.75	32.01
IV43	7.39	6.30	10.88	5.02	71.24	26.10
OE1	7.74	7.74	25.10	25.94	2.73	3.54
OE2	7.78	7.76	26.36	27.20	2.34	5.08
OE3	7.76	7.74	28.03	29.71	6.92	6.37
ARMA	6.79	6.25	0.84	6.69	92.71	46.00
AR1	6.87	6.27	2.93	7.95	94.16	50.44
AR2	6.87	6.27	2.93	7.95	94.18	50.44
AR3	7.62	6.29	8.79	5.86	96.79	63.12

Table 14: Results when analysing noisy voiced speech. Prediction and Kalman filter innovations are analysed for sum-squared error, whiteness and zero-mean, with median averages over all frames presented.

model	specDiff						
	ARMAX1	N4SID3	ARX1	IV41	OE1	ARMA	AR2
ARMAX1		9.41	9.30	11.26	10.10	10.46	10.12
ARMAX2	9.37	9.62	9.52	11.26	10.15	10.45	10.11
ARMAX3	9.64	9.76	9.70	11.27	10.20	10.47	10.15
N4SID1	9.48	9.24	9.55	11.26	10.15	10.45	10.11
N4SID2	9.42	9.04	9.50	11.26	10.13	10.46	10.12
N4SID3	9.41		9.46	11.26	10.10	10.47	10.15
N4SID4	9.42	8.87	9.50	11.26	10.12	10.46	10.13
ARX1	9.30	9.46		11.26	10.09	10.46	10.12
ARX2	9.52	9.63	9.25	11.26	10.14	10.47	10.12
ARX3	9.56	9.66	9.35	11.26	10.15	10.48	10.14
IV41	11.26	11.26	11.26		11.28	11.31	11.27
IV42	11.24	11.25	11.24	11.17	11.26	11.29	11.25
IV43	11.42	11.42	11.42	11.63	11.43	11.45	11.43
OE1	10.10	10.10	10.09	11.28		10.60	10.38
OE2	10.01	10.02	10.00	11.28	9.77	10.58	10.35
OE3	9.78	9.78	9.76	11.27	9.88	10.53	10.26
ARMA	10.46	10.47	10.46	11.31	10.60		10.36
AR1	10.07	10.10	10.07	11.27	10.36	10.36	9.20
AR2	10.12	10.15	10.12	11.27	10.38	10.36	
AR3	12.07	12.07	12.07	12.05	12.08	12.08	12.07

Table 15: Results when analysing noisy voiced speech. $\log_{10}$ mean squared spectral differences between noise model spectrograms.

Experiments on Clean Non-Voiced Speech

Model	$\log_{10}$(sse)		whiteness (%)		zero-mean (%)	
	predict	filter	predict	filter	predict	filter
ARMA	6.33	5.71	0.84	4.18	47.79	52.63
AR1	6.36	5.78	2.09	5.44	38.33	52.77
AR2	6.36	5.78	2.09	5.44	38.47	52.71
AR3	7.36	5.39	7.53	9.21	69.18	56.90

Table 16: Results when analysing clean non-voiced speech. Prediction and Kalman filter innovations are analysed for sum-squared error, whiteness and zero-mean, with median averages over all frames presented.

Model	specDiff			
	ARMA	AR1	AR2	AR3
ARMA		8.79	8.79	10.41
AR1	8.79		8.19	10.41
AR2	8.79	8.19		10.41
AR3	10.41	10.41	10.41	

Table 17: Results when analysing clean non-voiced speech. $\log_{10}$ mean squared spectral differences between noise model spectrograms.

Experiments on Noisy Non-Voiced Speech

Model	log10(sse)		whiteness (%)		zero-mean (%)	
	predict	filter	predict	filter	predict	filter
ARMA	6.72	6.32	1.26	7.11	39.48	42.65
AR1	6.74	6.37	1.67	7.95	36.04	42.78
AR2	6.74	6.37	1.67	7.95	36.00	42.76
AR3	7.49	6.29	7.53	5.44	75.33	49.97

Table 18: Results when analysing noisy non-voiced speech. Prediction and Kalman filter innovations are analysed for sum-squared error, whiteness and zero-mean, with median averages over all frames presented.

Model	specDiff			
	ARMA	AR1	AR2	AR3
ARMA		8.81	8.82	10.94
AR1	8.81		8.21	10.94
AR2	8.82	8.21		10.94
AR3	10.94	10.94	10.94	

Table 19: Results when analysing noisy non-voiced speech. $\log_{10}$ mean squared spectral differences between noise model spectrograms.

References

[1] K.J Åstrom. Maximum likelihood and prediction error methods. *Automatica*, 16:551–574, 1980.

[2] Y. Bar-Shalom and X-R Li. *Estimation and Tracking: Principles, Techniques, and Software.* Artech House, Norwood, MA, USA, 1993.

[3] G.J. Borden, K.S. Harris, and L.J. Raphael. *Speech Science Primer - Physiology, Acoustics, and Perception of Speech.* Williams and Wilkins, third edition, 1994.

[4] T.L. Burrows. *Speech Processing with Linear and Neural Network Models.* PhD thesis, SVR Group, Cambridge University Engineering Department, 1996.

[5] J.V. Candy. *Signal Processing The Model-Based Approach.* McGraw-Hill Book Company, 1986.

[6] C. Chatfield. *Statistics For Technology.* Chapman and Hall, 11 New Fetter Lane, London, 3rd edition, 1983.

[7] N.L.C Chui. *Subspace Methods and Informative Experiments for System Identification.* PhD thesis, Control Group, Cambridge University Engineering Dept, Cambridge, UK, 1997. Also available at `http://www-control.eng.cam.ac.uk/nlcc/report/thesis.ps`.

[8] J.R. Deller, J.G. Proakis, and J.H.L Hansen. *Discrete-Time Processing of Speech Signals.* Macmillan Publishing Co, Englewood Cliffs, NJ, 1993.

[9] D. Enns. *Model Reduction For Control System Design.* PhD thesis, Dept. of Aeronautics and Acoustics, Stanford Univ, USA, 1984.

[10] A. Gelb, editor. *Applied Optimal Estimation.* MIT Press, Cambridge, MA, 1974.

[11] M. Grice and W. Barry. Multi-lingual speech input/output: Assessment, methodology and standardization. Technical report, University College, London, 1989. ESPRIT Project 1541 (SAM), extension phase final report.

[12] A.C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter.* Cambridge University Press, Cambridge, UK, 1989.

[13] C. Heiji. *Deterministic Identification of Dynamical Systems.* Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin, Germany, 1989.

[14] B. Jones. *Statistics Toolbox For Use With MatLab.* The MathWorks, Inc., 24 Prime Park Way, Natrick, MA, USA, 1991.

[15] J-N Juang. *Applied System Identification.* PTR Prentice Hall, Englewood Cliffs, NJ, 1994.

[16] T. Kailath. *Linear Systems.* Prentice-Hall, 1980.

[17] S.M. Kay. The effects of noise on the autoregressive spectral estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(5):478–485, 1979.

[18] S.W. Kim, B.D.O. Anderson, and A.G. Madievski. Error bound for transfer function order reduction using frequency weighted balanced truncation. *Systems and Control Letters*, 24(3):183–192, 1995.

[19] J.S. Lim and A.V. Oppenheim. All-pole modeling of degraded speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(3):197–210, 1978.

[20] K. Liu. Modal parameter estimation using the state space method. *Journal of Sound and Vibration*, 197(4):387–402, 1996.

[21] L. Ljung. *System Identification: Theory for the User.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1987.

[22] L. Ljung. *System Identification Toolbox For Use With MatLab.* The MathWorks, Inc., 24 Prime Park Way, Natrick, MA, USA, 1991.

[23] B.C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, 1981.

[24] B.C.J Moore. *An Introduction to the Psychology of Hearing.* Academic Press Ltd, 24/28 Oval Road, London, UK, 3rd edition, 1989.

[25] L.R. Rabiner and B. Juang. *Fundamentals of Speech Recognition.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[26] B.D Rao and K.S. Arun. Model based processing of signals: A state space approach. *Proceedings of the IEEE*, 80(2), 1992.

[27] A. Rosenberg. Effect of glottal pulse shape on the quality of natural vowels. *Journal of the Acoustical Society of America*, 49(2):583–590, 1971.

[28] S.T. Roweis and Z. Ghahramani. A unifying review of linear gaussian models. *Neural Computation*, 11(2):305–345, 1999.

[29] G.A. Smith, M. Niranjan, and A.J. Robinson. *Speech Enhancement Using Subspace Methods.* PhD thesis, SVR Group, Cambridge University Engineering Dept, Cambridge, UK., 1999.

[30] T. Soderstrom and P. Stoica. *System Identification.* Prentice-Hall, London, 1989.

[31] P. Stoica and A. Nehorai. Music, maximum likelihood and cramer-rao lower bound. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37:720–741, 1989.

[32] P. Stoica and K.C. Sharman. Maximum likelihood methods for direction-of-arrival estimation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38:1132–1143, 1990.

[33] P. Stoica, T. Söderström, and B. Friedlander. Optimal instrumental variable estimates of the ar-parameters of an arma process. *IEEE Transactions on Automatic Control*, 30:1066–1074, 1985.

[34] A.J. Van Der Veen, E.F. Deprettere, and A.L Swindlehurst. Subspace-based signal analysis using singular value decomposition. *Proceedings of the IEEE*, 81(9), 1993.

[35] P. Van Overschee and B. De Moor. Subspace algorithms for the stochastic identification problem. *Automatica*, 29(3):649–660, 1993.

[36] P. Van Overschee and B. De Moor. Choice of state-space basis in combined deterministic stochastic subspace identification. *Automatica*, 31(12):1877–1883, 1995.

[37] P. Van Overschee and B. De Moor. A unifying theorem for three subspace system identification algorithms. *Automatica*, 31(12):1853–1864, 1995.

[38] P. Van Overschee and B. De Moor. A unifying theorem for three subspace system identification algorithms. *Automatica*, 31(12):1853–1864, 1995.

[39] P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems: Theory, Implementation, Applications.* Kluwer Academic Publishers, Dordrecht, Netherlands, 1996.

[40] M. Viberg and B. Ottersten. Sensor array processing based on subspace fitting. *IEEE Transactions on Signal Processing*, 39:1110–1121, 1991.

[41] M. Viberg, B. Wahlberg, and B. Ottersten. Analysis of state space system identification methods based on instrumental variables and subspace fitting. *Automatica*, 33(9):1603–1616, 1997.

[42] B. Wahlberg and L. Ljung. Design variables for bias distribution in transfer function estimation. *IEEE Transactions on Automatic Control*, 31(2):134–144, 1986.