The Relationship Between Discrete Time and Continuous Time Linear Estimation

Brett M. Ninness and Graham C. Goodwin
Department of Electrical Engineering & Computer Science
University of Newcastle
Shortland, 2308 AUSTRALIA

Abstract

We examine the problem of discrete time system estimation while not ignoring the underlying continuous time system. This leads to the use of a new discrete time operator, the δ operator, which approximates the continuous time derivative operator $\frac{d}{dt}$. We use this to formulate system estimation algorithms, and discuss their significantly superior numerical properties when compared to the equivalent shift operator formulated algorithms. We provide an overview of this new δ operator and also discuss some practical considerations in recursive least squares parameter estimation.

1 Introduction

Discrete time system analysis is usually done using the q forward shift operator and associated discrete frequency variable z. Unfortunately, the discrete domains resulting are unconnected with the continuous domains which spawn them. This is because the underlying continuous domain descriptions cannot be obtained by setting the sampling period to zero in the discrete domain approximations to them. Furthermore, it is widely known that serious numerical problems arise using shift operator formulations of algorithms at high sampling rates relative to the natural frequencies of the system being estimated.

Here we address this problem by formulating our discrete time description for a process while not ignoring the underlying continuous time process generating it. This leads naturally to the specification of a new discrete time operator, the δ operator, which is a difference operator and thus is the equivalent of the continuous time $\frac{d}{dt}$ operator. The use of this operator will be shown to entail many advantages. Namely:

1. There is a close connection between continuous time results and those formulated using the δ operator in discrete time since setting $\Delta = 0$ in a δ operator result gives the corresponding continuous time result.

2. The δ operator provides more insight into discrete time system analysis because of the similarity between the continuous time description for the process and the δ operator discrete time description.

3. Improved numerical properties are achieved. There are numerous areas relevant to system estimation where δ operator parameterisations of discrete time algorithms are numerically superior to their equivalent conventional q operator implementations. Specifically, the improved numerical properties we consider are:

   - Digital Filtering
     - Finite word length effects
     - Frequency response sensitivity
     - Round-off noise
   - Optimal state estimation: Conditioning of Riccati equations
   - Least squares parameter estimation: Conditioning of covariance matrix

Note that there is a simple linear transformation between δ operator parameterisations and shift operator parameterisations and so no sacrifice in modelling flexibility or statistical efficiency is made in using the δ operator.

We begin our discussion with a presentation of the δ operator, an associated discrete frequency domain transform (the Γ transform), and a discussion of the advantages of the δ operator over the conventional q operator. We also present a new generalised notation that will allow continuous and discrete results to be derived simultaneously. A new derivation and analysis of recursive least squares (RLS) parameter estimation and its variants is later done to illustrate the use of this notation.

Using the δ operator, we also address the problem of state estimation and show how the Kalman filter can be formulated using the δ operator as well as highlighting the advantages of doing this. We conclude with a discussion of some practical aspects associated with RLS parameter estimation.

1.1 A New Discrete Time Operator

When describing a model for a dynamic system it is common to use operator notation. That is, if it is a continuous time system the model for its performance is usually a differential equation, can be more compactly expressed using the operator $\rho \equiv \frac{d}{dt}$. For example:

$$\begin{aligned} 2\frac{d^2x}{dt^2} + \frac{dx}{dt} + 7x &= \frac{dy}{dt} + 7y \\ \Rightarrow (2\rho^2 + \rho + 7)x &= (\rho + 7)y \end{aligned}$$

If it is a discrete time system, the model for its performance is usually a difference equation. This can be more compactly expressed using the forward shift operator q which is defined by:

$$qx_k \equiv x_{k+1}$$

For example:

$$\begin{aligned} 2x_{k+2} + x_{k+1} + 7x_k &= y_{k+1} + 7y_k \\ \Rightarrow (2q^2 + q + 7)x_k &= (q+7)y_k \end{aligned}$$

Unfortunately, the shift operator has no continuous time counterpart. Consequently, discrete time representations using the q operator do not converge smoothly to the underlying continuous time system as the sampling interval goes to zero. This motivates us to present a new discrete time operator called the δ operator that does have a continuous time counterpart. Using this operator continuous time systems are seen as the limiting case of a corresponding discrete time system as the sampling interval tends to zero.

2 The Delta Operator

The δ operator is defined as follows:

$$\begin{aligned} \delta &\triangleq \frac{q-1}{\Delta} \\ \Rightarrow \delta x_k &= \frac{(q-1)x_k}{\Delta} \\ &= \frac{x_{k+1} - x_k}{\Delta} \end{aligned} \tag{1}$$

This operator has been known in the numerical analysis field as the first divided difference operator [7]. From the above we see that the δ operator approximates the derivative:

$$\delta x_k \approx \left. \frac{dx}{dt} \right|_{x=x(k\Delta)}$$

with the approximation becoming better as the sampling interval tends to zero. Therefore, because the δ operator has the continuous time counterpart ρ, models for systems expressed in terms of the δ operator are very similar to models expressed with the differentiation operator ρ, or the Laplace transform variable s. Because of this the use of δ operators permits continuous time intuition and insights to be used in discrete time systems. Furthermore, it provides equivalent flexibility in the context of discrete time modelling as does the shift operator q.

2.1 Obtaining δ Operator Discrete time Models for Systems

Since (1) illustrates that there is a linear transformation between the shift operator and δ operator models for systems, the derivation of the δ operator discrete time models for continuous time systems is as straightforward as for the q operator case.

Specifically, suppose we have a continuous time model for a SISO system expressed in rational proper transfer function form using the differentiation operator ρ:

$$\frac{y(t)}{u(t)} = \frac{B(\rho)}{A(\rho)} = G(\rho)$$

$$\begin{aligned} A(\rho) &= \rho^n + a_{n-1}\rho^{n-1} + \ldots + a_1\rho + a_0 \\ B(\rho) &= b_m\rho^m + b_{m-1}\rho^{m-1} + \ldots + b_1\rho + b_0 \end{aligned}$$

This can be converted to an equivalent state space form using a canonical representation [6]:

$$\rho\vec{x}(t) = A\vec{x}(t) + Bu(t) \tag{2}$$
$$y(t) = C\vec{x}(t) + b_n u(t) \tag{3}$$

Note that if $G(\rho)$ is strictly proper then $b_n = 0$. Assuming zero order hold sampling, the equivalent shift operator state space description is:

$$q\vec{x}_k = M\vec{x}_k + Nu_k \tag{4}$$
$$y_k = S\vec{x}_k + Tu_k \tag{5}$$

Where M,N,S and T are given by [12]:

$$M = e^{A\Delta} \quad N = A^{-1}(e^{A\Delta} - I)B \quad S = C \quad T = b_n$$

Here $e^{A\Delta}$ is the matrix exponential [12] and Δ is the sampling period. Considering the definition in (1) then gives the δ operator state space representation from (4) and (5) by subtracting $\vec{x}_k$,and dividing by Δ on both sides of (4):

$$\delta\vec{x}_k = F\vec{x}_k + Gu_k \tag{6}$$
$$y_k = H\vec{x}_k + Ku_k \tag{7}$$

where

$$F = \Omega A; \quad G = \Omega B; \quad H = C; \quad K = b_n$$

and where we define Ω by:

$$\Omega \triangleq (e^{A\Delta} - I)\frac{A^{-1}}{\Delta} \qquad \text{If A non-singular} \tag{8}$$
$$= I + \frac{A\Delta}{2!} + \frac{A^2\Delta^2}{3!} + \frac{A^3\Delta^4}{4!} + \ldots \tag{9}$$
$$\tag{10}$$

Therefore, it can be seen that at high sampling rates relative to the system bandwidth, $A\Delta \to 0$ to give $\Omega \approx I$ and so the state space discrete time representation for a system using the δ operator will be very similar to the continuous time representation. Hence, it will be possible to assume $G(\delta) \approx G(s)$ if the sampling rate is sufficiently high. Empirically, sufficiently high has been found to mean sampling rates greater than ten times the plant bandwidth. Of course, it is always possible to work with the exact $G(\delta)$.

Also note that if the calculation of Ω is to be done via a computer, which is almost always the case, then the power series expression up to an appropriate number of terms should be used. This is because finite word length limits on floating point accuracy will more strongly affect the evaluation of the expression (8) then they will the expression (9). Finally, if desired, this δ operator discrete time state space description can be converted into a rational transfer function form by:

$$G(\delta) = \frac{y_k}{u_k} = \frac{det[\delta I - (F - GH)] + (K-1)det(\delta I - F)}{det(\delta I - F)}$$

2.2 Implementation of Filters

There are many ways in which to implement digital filters [2]. Here we present one way in order to illustrate the implementation of DSP algorithms formulated using the δ operator without converting back to shift operator form. This is essential to realising the improved numerical properties that are possible through δ operator parameterisation.

We assume without loss of generality that the filter is specified in δ operator state space form (6),(7). If it represented in δ operator transfer function form then a canonical representation may be used to obtain a representation as per (6), (7). Multiplying both sides of (6) by the inverse operation δ^{-1} then gives:

$$\begin{aligned} \vec{x}_k &= \delta^{-1}[F\vec{x}_k + Gu_k] & (11) \\ y_k &= H\vec{x}_k + Ku_k & (12) \end{aligned}$$

This suggests that the basic building block in the implementation of digital filters using δ operators is the function δ^{-1}. This compares with the usual building block, the backward shift function q^{-1}. The function δ^{-1} is defined to operate in the following way:

$$\alpha_k = \delta^{-1}\beta_k \qquad \Rightarrow \qquad \alpha_{k+1} = \alpha_k + \Delta\beta_k \tag{13}$$

The function δ^{-1} is thus seen to act as a discrete time integrator. This δ^{-1} building block can be manipulated in much the same way as integrators are manipulated in the implementation of continuous time models. Additionally, (13) shows that the hardware or software needed to implement this building block is only marginally more complex than that required for the more usual delay building block q^{-1} which it replaces. Considering (11) and (12),then gives that the required recursive updating of the state space equations to implement the filter are:

$$\begin{aligned} \vec{x}_{k+1} &= \vec{x}_k + \Delta F\vec{x}_k + \Delta Gu_k & (14) \\ y_{k+1} &= H\vec{x}_{k+1} + b_n u_{k+1} \end{aligned}$$

Note that (14) should be implemented exactly. The following difference equation:

$$\vec{x}_{k+1} = (I + \Delta F)\vec{x}_k + \Delta E u_k \tag{15}$$

which is mathematically equivalent to (14) should not be implemented since in practical terms it is not equivalent to the implementation of (14). This is due to finite word length effects in computer implementation which become apparent when sampling rates become high. In this case, Δ becomes small and so:

$$I + \Delta F \approx I \tag{16}$$

If only a finite word length is available, as is always the case in the computer implementation of the recursive update equation (15), then the approximation in (16) may in fact become equality. The required dynamics of the filter will thus not be realised at fast sampling rates. A major exacerbating factor of this problem is that much of the word length in the multiplication involved in (15) is taken up by registering the presence of the identity matrix I.

However, note that in the formulation of the update equation given in (14), the full word length of the computer is available to calculate the possibly small quantity ΔF. Eventually of course, as Δ becomes smaller, finite word length effects will make ΔF appear to be zero and once again the required filter dynamics will not be realised. However, it can be shown [12] that this will occur at much higher sampling rates than occurs for the formulation (14).

Therefore, implementation of the recursive difference equations to realise a digital filter exactly via the formulation (14) rather than the formulation (15) will result in higher sampling rates being possible before numerical difficulties degrade the performance of the filter.

2.3 Stability Region for δ Operator Models

It follows that since the poles of ρ and q domain input - output models for systems must lie in well known areas of the complex plane in order for the response of the system to be asymptotically stable, then a similar result should hold for the poles of δ domain models for systems.

Such a result is seen straightforwardly by noting that the Hurwitz stability region for the poles of q domain models of systems is the open unit disc centred on the origin [2]. Furthermore, the δ operator is defined in terms of the shift operator q via (1). This is a simple linear transformation, under which it is obvious that the stability region in q maps to the disc radius $\frac{1}{\Delta}$ and centre $\frac{-1}{\Delta}$. This then is the region in the complex plane that the poles of a δ operator model for a system must lie if the response is to be asymptotically stable. This region, and the mapping involved, is shown in figure 1. Note that as $\Delta \to 0$, the δ stability region expands to fill the whole open left half of the complex plane. This, of course, is the continuous time Hurwitz stability region and so once again it is seen that δ discrete time models converge towards continuous time models as the sampling interval tends to zero.

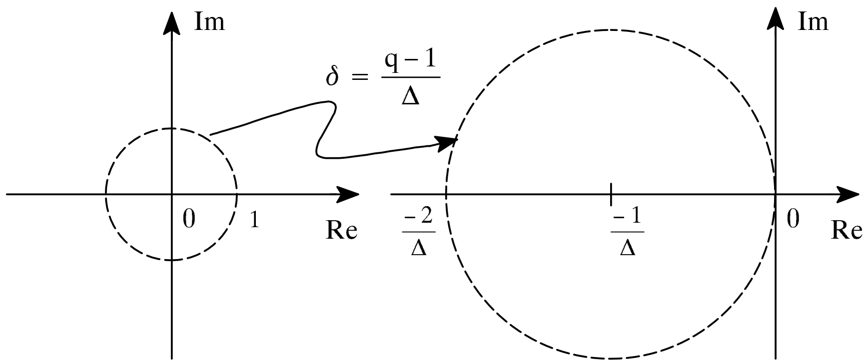

Figure 1: *Hurwitz stability regions for the poles of q and δ operator models*

2.4 A Generalised Notation

It has already been mentioned that the δ operator approximates a continuous time derivative operator as the sampling interval tends to zero. This suggests that the formulation of algorithms in terms of delta operators will not only be valid for discrete time implementation, but will tend to the equivalent continuous time formulations as the sampling interval tends to zero. This, in fact, is the case, but to make it clear some specialised notation needs to be introduced. The notation used has been drawn from [12]. We present it here for later use in the derivation of the discrete time least squares parameter estimation algorithm. We also present some elementary results formulated using this notation that we will require later. These results give a flavour for how this new notation is used.

2.4.1 The Generalised Derivative

Definition 1. Up to this point, the symbol ρ has been used to represent continuous time differentiation. This will remain the case, but will also be used to represent the δ operator in discrete time. Therefore, any formulations obtained in terms of ρ can be interpreted as discrete time expressions by replacing ρ with δ. More succinctly, if Δ represents the sampling interval:

$$\rho = \begin{cases} \frac{d}{dt} & \Delta = 0 \\ \delta & \Delta \neq 0 \end{cases}$$

2.4.2 Generalised Integration/Summation

Definition 2. The same symbol $\mathcal{S}$ will be used to represent Riemann integration in continuous time, and summation in discrete time. Specifically the definition is:

$$\mathcal{S}_{t_1}^{t_2} f(\tau)d\tau = \begin{cases} \int_{t_1}^{t_2} f(\tau)d\tau & \Delta = 0 \\ \Delta \sum_{k=\frac{t_1}{\Delta}}^{k=\frac{t_2}{\Delta}-1} f(k\Delta) & \Delta \neq 0 \end{cases}$$

2.4.3 Reciprocity of Integration/Differentiation

Lemma 1. *The results for generalised integration of a generalised derivative and generalised differentiation of a generalised integral follow as per the continuous time cases. Specifically:*

$$\begin{aligned} \rho\left[\mathcal{S}_{a}^{t} f(\tau)d\tau\right] &= f(t) \\ \rho\left[\mathcal{S}_{t}^{b} f(\tau)d\tau\right] &= -f(t) \\ \mathcal{S}_{a}^{b} [\rho f(\tau)]d\tau &= f(b) - f(a) \end{aligned}$$

Proof. These results may be trivially obtained by considering the continuous and discrete time cases separately. ▽▽▽

2.4.4 Differentiation of a Product

Lemma 2. *Consider two real scalar functions of a real scalar variable t: $f(t)$ and $g(t)$. For these functions, the product rule for generalised differentiation is:*

$$\rho(fg) = (\rho f)g + f(\rho g) + \Delta(\rho f)(\rho g)$$

Proof.

$$\begin{aligned} \rho(fg) &= \frac{f[(k+1)\Delta]g[(k+1)\Delta] - f(k\Delta)g(k\Delta)}{\Delta} \\ &= \frac{[f[(k+1)\Delta] - f(k\Delta)]\, g(k\Delta) + f(k\Delta)\, [g[(k+1)\Delta] - g(k\Delta)]]}{\Delta} + \\ &\quad \frac{\Delta\, [f[(k+1)\Delta] - f(k\Delta)]\, [g[(k+1)\Delta] - g(k\Delta)]]}{\Delta^2} \\ &= (\rho f)g + f(\rho g) + \Delta(\rho f)(\rho g) \end{aligned}$$

▽▽▽

Notice that as the sampling interval tends to zero, the result for generalised differentiation of a product of two functions tends to the well known product rule result for continuous time derivatives as expected.

2.4.5 Matrix Differencing Lemma

Lemma 3. *Consider a square invertible matrix A whose generalised derivative can be written via the vector B as:*

$$\rho A = BB^T$$

Then the generalised derivative of A^{-1} is given by:

$$\rho A^{-1} = \frac{-A^{-1}BB^TA^{-1}}{1 + \Delta B^T A^{-1} B}$$

Proof. The generalised derivative of A^{-1} is given by:

$$\begin{aligned}
\rho(A^{-1}) &= \frac{A^{-1}[(k+1)\Delta] - A^{-1}(k\Delta)}{\Delta} \\
&= \frac{A^{-1}[(k+1)\Delta]A(k\Delta)A^{-1}(k\Delta) - A^{-1}[(k+1)\Delta]A[(k+1)\Delta]A^{-1}(k\Delta)}{\Delta} \\
&= \frac{A^{-1}[(k+1)\Delta][A(k\Delta) - A[(k+1)\Delta]]A^{-1}(k\Delta)}{\Delta} \\
&= A^{-1}[(k+1)\Delta](\rho A)A^{-1}(k\Delta) \\
&= A^{-1}[(k+1)\Delta]BB^TA^{-1}(k\Delta) \qquad (17)
\end{aligned}$$

However, the generalised derivative of A is also given through definition by:

$$\begin{aligned}
\rho A &= \frac{A[(k+1)\Delta] - A(k\Delta)}{\Delta} \\
\Rightarrow A[(k+1)\Delta] &= A(k\Delta) + \Delta BB^T
\end{aligned}$$

Using the matrix inversion lemma [5] and substituting into (17) then gives the result.

∇∇∇

2.5 The Delta Transform and Frequency Response of δ Models

We now present a new transform, the Γ transform, that can be applied to δ operator models that will give the frequency response of both the continuous time model and the discrete time model for a plant simultaneously; the continuous frequency response being the limit of the discrete frequency response as the sampling interval tends to zero.

A heuristic justification of the Γ transform may be obtained by noting that for zero initial conditions, the z domain transfer function can be obtained by substituting $q = z$. Furthermore, the δ and q operators are related by definition in (1) which may be arranged to:

$$q = \Delta\delta + 1$$

This suggests a definition for the Γ transform, in terms of a new transform variable γ as:

$$\begin{aligned}
F_\Delta(\gamma) &= F(z)|_{z=\Delta\gamma+1} \\
&= \sum_{k=0}^{k=\infty} f_k(1+\Delta\gamma)^{-k}
\end{aligned}$$

Where $\Gamma[f(\delta)] = F_\Delta(\gamma)$. Middleton and Goodwin [12] modify this definition in order to ensure that $F_\Delta(\gamma) \to F(s)$ as $\Delta \to 0$ by scaling the above definition by Δ to arrive at the final definition for the Γ transform:

$$F_\Delta(\gamma) = \Delta F(z)\big|_{z=\Delta\gamma+1}$$

Furthermore, the following shift property holds for the Z transform:

$$Z(qf_k) = \mathrm{Z}\left[zf_k - f_0\right]$$

Therefore, the definition of the Γ transform provides the equivalent differentiation property:

$$\Gamma(\delta f_k) = \gamma\Gamma(f_k) - f_0(1+\Delta\gamma)$$

Applying this result to a δ domain model of a system expressed in state space form as:

$$\begin{aligned} \delta\vec{x}_k &= F\vec{x}_k + Gu_k \\ y_k &= H\vec{x}_k \end{aligned}$$

gives:

$$\begin{aligned} \gamma X(\gamma) - x_0(1+\Delta\gamma) &= FX(\gamma) + GU(\gamma) \\ Y(\gamma) &= CX(\gamma) \end{aligned}$$

Consequently, if we are only interested in the forced response of a system after transients due to initial conditions have died out, then the delta transform model of a plant modelled using the delta operator may be found by replacing δ by γ analagously to the previous cases of replacing q with z and ρ with s. That is:

$$G(\gamma) = \frac{Y(\gamma)}{U(\gamma)} = \left.\frac{B(\delta)}{A(\delta)}\right|_{\delta=\gamma}$$

Finally then, the frequency response of the delta operator model may be found by noting that:

$$\gamma = \frac{z-1}{\Delta} \tag{18}$$

and since the frequency response of z transform models is found by substituting $z = e^{j\omega\Delta}$, so the frequency response of $G(\gamma)$ at the angular frequency ω is found by:

$$\text{Frequency Response} = G(\gamma)\big|_{\gamma=\frac{e^{j\omega\Delta}-1}{\Delta}}$$

Morover, a Taylor series expansion for $\frac{e^{j\omega\Delta}-1}{\Delta}$ about $\omega = 0$ gives:

$$\frac{e^{j\omega\Delta}-1}{\Delta} = j\omega\left(1 + \frac{j\omega\Delta}{2!} - \frac{\omega^2\Delta^2}{3!} + \ldots\right)$$

Therefore, for sampling rates greater than 20 times any frequency of interest (ie. 20 times the system bandwidth) we have:

$$\frac{\omega\Delta}{2} < 0.15$$

and so the frequency response of a δ operator transfer function can be roughly found by substituting $\delta = j\omega$. This is obviously intuitively appealing.

2.6 Relationship between γ Domain Poles and System Response

The derivation of the Γ transform allows the mapping of poles from the s domain to the γ domain. Since the relationship between the location of s domain poles and plant response is well known, this mapping will provide insight into how the location of the γ domain poles of a system affect the response of a discrete time system. Note that [2] gives the mapping between the s and z domains as $z = e^{s\Delta}$. Considering (18) we find the mapping between the s and γ domains as:

$$\gamma = \frac{e^{s\Delta} - 1}{\Delta} \tag{19}$$

Note that this mapping is irrespective of the type of holding circuit used for the discrete time system. This is so because the poles of a system describe the natural response of a system when the input forcing signal is zero. Obviously, then the input holding circuit cannot affect the natural response of the system and therefore has no effect on the poles of the system. Such a simple situation does not hold for the zeroes of a system which describe non zero input forcing signals which cause zero output from the system. Obviously these zeroes will be intimately related to the type of input holding circuit used for the discrete time system, and there will be no simple mapping from continuous time zeroes to discrete time zeroes as there is for continuous time poles to discrete time poles [12]. The simple mapping proposed for the poles leads to three important conclusions:

1. $s = 0 \Rightarrow \gamma = 0$ and as $s \to -\infty$ along the real axis, $\gamma \to \frac{-1}{\Delta}$ along the real axis. This mapping is shown in figure 2. Therefore, poles in the γ domain near the real axis between the origin and the point $\gamma = \frac{-1}{\Delta}$ coincide with a well damped system response, with the response becoming quicker as the poles move to the left, analogous to the continuous time case. Furthermore, this mapping highlights the fact that there is a finite limit to how fast a sampled data system can respond. That is, it is obvious that it can respond no quicker than the sampling interval Δ.

2. Assume a continuous time pole at $s = -\alpha + j\beta$. Substituting this into (19) gives:

   $$1 + \Delta\gamma = e^{(-\alpha+j\beta)\Delta} = e^{-\alpha\Delta}(\cos\Delta\beta + j\sin\Delta\beta)$$

   Now suppose that γ is a complex number given by $\gamma = x + jy$. In this case:

   $$\begin{aligned} \cos\Delta\beta &= e^{\alpha\Delta}(1 + \Delta x) \\ \sin\Delta\beta &= e^{\alpha\Delta}(\Delta y) \end{aligned}$$

   $$\Rightarrow \left(x + \frac{1}{\Delta}\right)^2 + y^2 = \frac{1}{(\Delta e^{\alpha\Delta})^2} \tag{20}$$

   Therefore, the straight line locii $z = -\alpha + j\beta$ with α constant in the s domain maps to a circle centre $\frac{-1}{\Delta}$ and radius $\frac{1}{\Delta e^{\alpha\Delta}}$ in the γ domain. This is shown in figure 3.

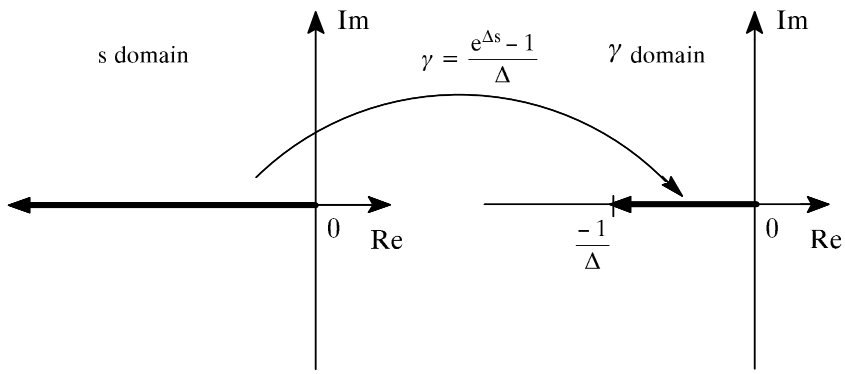

Figure 2: *Mapping of the negative real axis in the s domain to the γ domain*

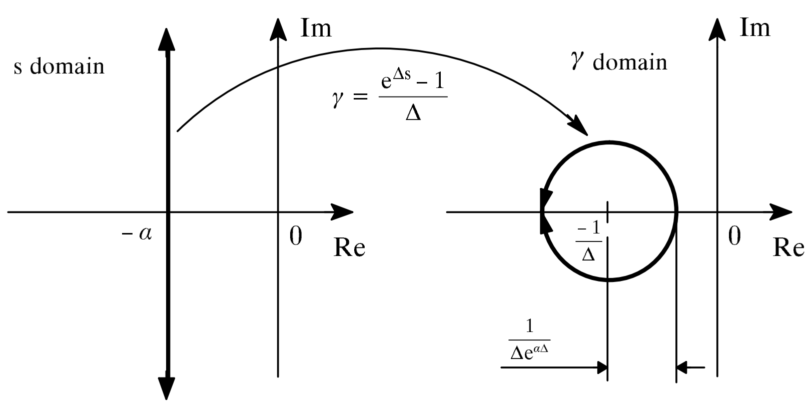

Figure 3: *Mapping the loci of poles with constant real part in the s domain to the γ domain*

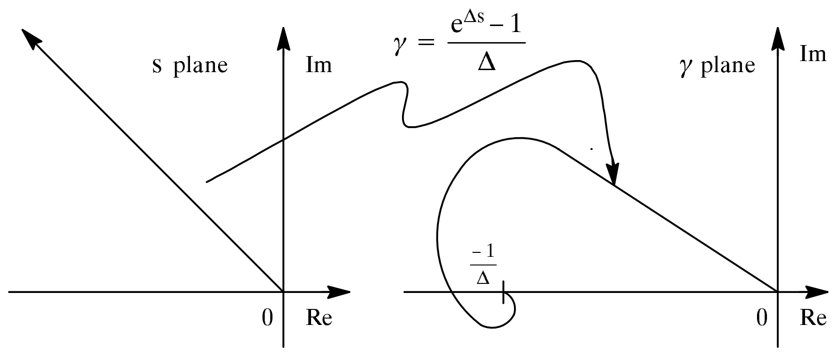

Figure 4: *Loci of poles with constant damping ratio in the s domain, and the loci they map to in the γ domain for a particular case.*

This highlights the interesting result that poles near the real axis in the γ domain can represent a very poorly damped system response if the pole is to the left of $\frac{-1}{\Delta}$. Furthermore, it is interesting to note how the locii of poles with a fixed damping ratio in the s domain map to the γ domain. The loci of poles in the s plane with constant damping ratio ζ is defined by the equation:

$$s = -\omega \cot \phi + j\omega \qquad \zeta = \cos \phi$$

Poles defined by this equation map to poles in the γ domain defined by:

$$\gamma = \frac{e^{-\Delta\omega \cot \phi} e^{j\Delta\omega} - 1}{\Delta}$$

This is an exponentially decaying spiral centred on $\gamma = -\frac{1}{\Delta}$ as shown in figure 4.

3. Finally, by substituting $\alpha = 0$ into (20) the s domain stability boundary is seen to map to the circle shown in figure 1. This provides agreement with the same result found earlier.

2.7 Advantages of δ Operators

By now, the intuitive advantages of using the δ operator to parameterise discrete time approximations to continuous time systems should be apparent. We now proceed to emphasise these advantages, as well as to highlight significant numerical benefits stemming from their use.

2.7.1 Insight Advantages

As has already been shown, the δ operator is a difference operator that approximates the derivative operator ρ. As a result of this, the discrete time model for a sampled continuous time system is very similar to the continuous time system model expressed in terms of the ρ operator. Thus continuous time insights can be used in discrete design. This is not true if we use discrete time models expressed in terms of the shift operator q.

For example, consider the continuous time system modelled using the ρ operator:

$$G(\rho) = \frac{-\rho + 5}{\rho^4 + 23\rho^3 + 185\rho^2 + 800\rho + 2500} \tag{21}$$

The zero order hold discrete time approximation to this system parameterised by the q operator and assuming a 100Hz sampling rate (rounded to 5 significant digits) is:

$$G(q) = \frac{-10^{-6}(0.15537q^3 + 0.41605q^2 - 0.47402q - 0.14200)}{q^4 - 3.7777q^3 + 5.3506q^2 - 3.3674q + 0.79453} \tag{22}$$

This is not at all similar to the continuous time model and the poles and zeroes of the above transfer function are not obviously related to those of the continuous time system. That is, a 'quick pole' in continuous time, ie. one in the far left hand of the complex ρ plane, does not map to one in the far left hand of the complex q plane. However, the δ operator form of the transfer function describing the discrete time domain response (rounded to 2 significant digits) is:

$$G(\delta) = \frac{-0.80\delta + 4.5}{\delta^4 + 22\delta^3 + 180\delta^2 + 760\delta + 2200} \tag{23}$$

This is quite similar to the continuous time model, with the poles and the zero being quite close to the continuous time poles and zero. This similarity between $G(s)$ and the δ operator discrete time approximation $G(\delta)$ is particularly important in the context of discrete time parameter estimation since it will be particularly easy to relate the estimated transfer function $\hat{G}(\delta)$ to the underlying continuous time process. Such is not the case for a q operator parameterised estimate $\hat{G}(q)$.

2.7.2 Finite Word Length Considerations

Many algorithms are better conditioned numerically using δ operator implementation than shift operator implementation. Most fundamentally, digital filtering operations are less prone to finite word length problems at higher sampling rates. This is due to the fact that as sampling rates increase, the poles and zeroes of models represented using shift operator notation tend to cluster about the point $q = 1$. Thus, the shift operator discrete time state transition matrix M tends to the identity matrix. This may be seen by noting that if A is the continuous time state transition matrix then M is given by:

$$M = e^{A\Delta} = 1 + \frac{A\Delta}{1!} + \frac{(A\Delta)^2}{2!} + \frac{(A\Delta)^3}{3!} + \dots$$

The dynamics of the system will be captured by the fractional part of the entries in M, but in floating point computer implementation much of the available word length will be used recognising the presence of the non-fractional part (i.e. 1) of M. Consequently, at a sufficiently high sampling rate the dynamics of the filter will be lost since the fractional part will be too small to be represented with the word length available.

The δ operator, being defined as $\delta = \frac{q-1}{\Delta}$, avoids this problem by shifting the point $q = 1$ to the point $\delta = 0$ and then scaling by the factor $\frac{1}{\Delta}$. Thus, the poles of δ operator models will tend to their continuous time values as the sampling rate increases. Again, this may be seen by considering the discrete time state transition matrix F. This time for a delta operator model:

$$F = A\left(I + \frac{A\Delta}{2!} + \frac{(A\Delta)^2}{3!} + \dots\right)$$

Therefore, the discrete time transition matrix will tend towards the continuous time transition matrix. Additionally, using δ operators the states of a filter are updated according to (14). Evidently the dynamics of the filter are contained in the matrix ΔF. This will become very small as the sampling rate increases, but with computer implementation, the full word length of the computer may be used to store the important fractional entries in ΔF. Eventually, as the sampling rate increases, finite word length effects will cause ΔF to appear as the zero matrix, just as F will eventually appear like the identity matrix using shift operators, and the dynamics of the filter will no longer be achieved. However, this will occur at a much higher sampling rate using δ operators [11]. In [11] it is shown that for a given word length, samplings rate five to ten times faster than achievable with shift operators can be sustained with δ operator implementation. The motivation for using fast sampling will be detailed presently.

To illustrate this numerical advantage, the step response of the continuous time system given by (21) was simulated over 10 seconds as is shown in figure 5. Also shown superimposed on this is the step response for the shift and δ operator discrete time approximations (22) and (23). The shift approximation is the upper trace in figure 5, and the continuous and δ approximation are the lower traces. As can be seen, the q operator implementation is not a very good approximation to the continuous time system, while the δ operator implementation is so close to the true continuous time response that it is difficult to tell the two apart. Notice too that the shift co-efficient representation in (22) involves 3 more significant digits than for the δ operator representation in (23) !

2.7.3 Frequency Response Sensitivity

There are other numerical advantages. Suppose we have a fixed precision in the representation of the coefficients of the denominator of a discrete time transfer function because

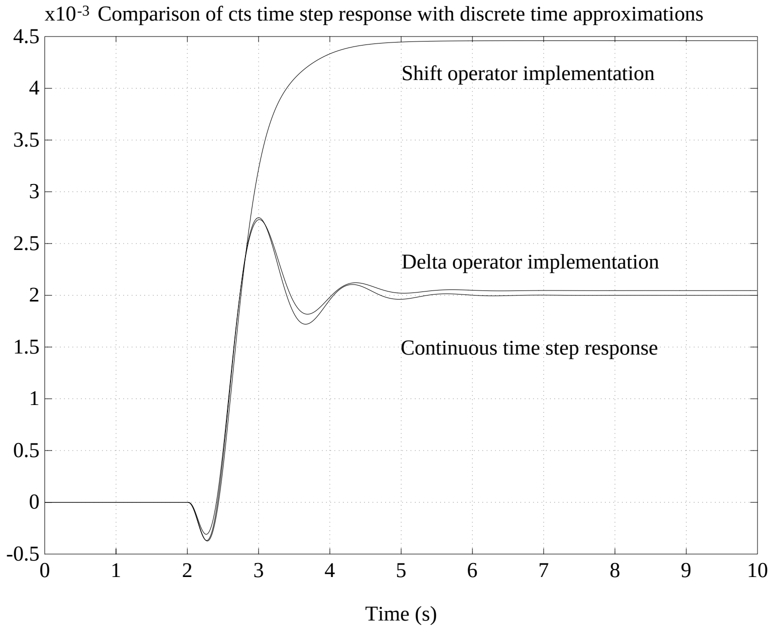

Figure 5: *Comparison of step response of continuous time system to the step responses of discrete time approximating systems using shift and δ operator implementations*

of finite word length effects; be it in δ or q form. This means that there is an error ϵ in the representation of the coefficients of a filter. Given this error in the denominator for example, there must also be an error η in the pole positions of the discrete time transfer function. It can be shown [11] that for a given ϵ, the upper bound for η is always smaller using a δ operator implementation than with a shift operator implementation. This is regardless of whether a fixed or a floating point implementation is used. Stated another way, to get within x % of required pole positions requires less bits of accuracy when using δ operator implementations rather than shift operator implementation. In general, to achieve an error of less then x % in pole positions, requires 5 to 7 bits less word length using δ operators [12].

2.7.4 Round Off Noise

In most finite word length implementations of DSP algorithms there will be errors introduced into the system due to the finite word length available for the calculation and storage of intermediate quantities. Under suitable conditions on the input to a discrete time system (namely that there be sufficient noise and/or input variation), the errors introduced may be considered as an almost stochastic process [18] [13]. Thus the term 'round off noise' has arisen. It is shown in [12] that at sampling rates high with respect to the system bandwidth, δ operator implementation of filters results in the introduction of less round off noise on the output than for shift operator implementation.

2.7.5 Fast Sampling Rates

Note that the δ operator will always outperform the shift operator in terms of intuitive insights and numerical performance. However, the difference in performance is most appreciable at sampling rates greater than twenty times the system bandwidth where, unless great care is taken, shift operator implementation of filters will simply fail (see figure 5). Furthermore, we have shown how at these frequencies $G(\delta) \approx G(s)|_{s=\delta}$ and the discrete frequency response is $\approx G(\delta)|_{\delta=j\omega}$.

We emphasise these advantages by enumerating other motivations for fast sampling in the context of adaptive control (a large application area of system estimation). Specifically, these advantages are;

1. Aliasing effects due to frequency folding are significantly reduced or eliminated, and consequently the specification of the front end anti-aliasing filters can be relaxed. Since these filters have to be taken into account in system estimation this is a major advantage.

2. There is a smoother progression in control input to the plant. If slow sampling is used then the control input can be a sequence of large step changes [6]. This can feed significant energy into high frequency mechanical resonances. Rapid sampling ensures a smooth sequence of smaller changes to achieve the same bandwidth.

3. The discrete time response is a better approximation to the desired continuous time response.

4. Higher closed loop bandwidths can be achieved.

Having introduced the δ operator and enumerated its virtues, we now go on to examine system estimation algorithms parameterised with the δ operator.

3 System Estimation

The vast majority of discrete time system estimation literature uses formulations involving the shift operator q. Consequently, the resultant estimates pertain to a discrete time system only, seemingly ignoring the underlying continuous time system. As discussed in [14],[4],[12] and [15] this causes state estimates to degenerate at high sampling rates to a constant irregardless of the underlying continuous time system, namely the trivial model $y_{k+1} = y_k$.

Consequently, we begin with a consideration of this problem. We will use the δ operator to establish a direct connection between the continuous and discrete formulations of the state estimation problem and show that this eliminates the Kalman filter degeneracy problem at high sampling rates. We will also compare the numerical robustness of the Kalman filter using the q and δ operators.

All the results here will, for the sake of brevity, be presented without proof. For readers interested in these proofs, and a more detailed discussion of the results, then [14],[4],[12] and [15] are the appropriate references.

3.1 State Estimation

Consider the continuous time SISO linear time invariant stochastic system given by:

$$dx(t) = Ax(t)dt + d\nu(t) \tag{24}$$
$$dz(t) = Cx(t)dt + d\omega(t) \tag{25}$$

where $\nu(t)$ and $\omega(t)$ are Wiener processes with incremental covariances:

$$\mathcal{E}\left\{\begin{bmatrix} d\nu(t) \\ d\omega(t) \end{bmatrix}\begin{bmatrix} d\nu^T(t) & d\omega^T(t) \end{bmatrix}\right\} = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} dt \tag{26}$$

and $\mathcal{E}$ denotes expectation over the underlying probability space. If we wish to find a state estimate $\hat{x}(t)$ minimising

$$J(t) = \mathcal{E}\{[x(t) - \hat{x}(t)][x(t) - \hat{x}(t)]^T\} \tag{27}$$

then the solution is well known [1] to be the continuous time Kalman filter

$$d\hat{x}(t) = A\hat{x}(t)dt + H(t)[dz(t) - C\hat{x}(t)] \tag{28}$$
$$H(t) = P(t)C^T R^{-1} \tag{29}$$

Where $P(t)$ satisfies the continuous time Riccati differential equation (CRDE):

$$\dot{P}(t) = P(t)A^T + AP(t) - P(t)C^T R^{-1} CP(t) + Q \tag{30}$$

Formulating the discrete time equivalent to the system in (24) and (25) presents the problem that sampling the output $\frac{dz}{dt}$ generates a system having output noise of infinite variance. This is overcome [15] by accounting for low pass filtering prior to sampling to arrive at a discrete time approximation to (24),(25):

$$x(k+1) = A_q x(k) + \nu_q(k) \tag{31}$$
$$y(k) = C_q x(k) + \omega_q(k) \tag{32}$$

with

$$\mathcal{E}\left\{\begin{bmatrix} \nu_q(k) \\ \omega_q(k) \end{bmatrix}\begin{bmatrix} \nu_q^T(k) & \omega_q^T(k) \end{bmatrix}\right\} = \begin{bmatrix} Q_q & S_q \\ S_q^T & R_q \end{bmatrix}$$

The optimal state estimate for a system described via this joint Markov model is well known [6] as:

$$\hat{x}(k+1) = A_q\hat{x}(k) + H_q(k)[y(k) - C_q\hat{x}(k)] \tag{33}$$
$$H_q(k) = [A_q P_q(k) C_q^T + S_q][C_q P_q(k) C_q^T + R_q]^{-1} \tag{34}$$

Where $P_q(k)$ satisfies the following discrete Riccati difference equation (DRDE):

$$P_q(k+1) = Q_q + A_q P_q(k) A_q^T - H_q(k)[C_q P_q(k) C_q^T + R_q] H_q^T(k) \tag{35}$$

However, it is noted [15] that the Kalman gain H_q in this solution has the property

$$\lim_{\Delta \to 0} H_q = 0$$

where Δ is the sampling interval. This is seen to stem from the fact that:

$$\lim_{\Delta \to 0} A_q = I \qquad \lim_{\Delta \to 0} Q_q = 0 \qquad \lim_{\Delta \to 0} R_q = \infty \tag{36}$$

These results apply for a large class of pre-sampling filters and irregardless of the dynamics of underlying continuous time process in (24),(25). This is the problem of Kalman filter degeneracy mentioned in the introduction and will obviously lead to numerical problems at high sampling rates. These problems can be overcome by formulating the discrete time state space approximation to (24) and (25) using the δ operator:

$$\delta x(t) = A_\delta x(k) + \nu_\delta(k) \tag{37}$$
$$y(k) = C_\delta x(k) + \omega_\delta(k) \tag{38}$$

where

$$A_\delta = \frac{1 - A_q}{\Delta} \qquad \nu_\delta(k) = \frac{\nu_\delta(k)}{\Delta}$$
$$C_\delta = C_q \qquad \omega_\delta(k) = \omega_q(k)$$

Furthermore, if we define the δ operator covariances as:

$$Q_\delta = \frac{Q_q}{\Delta} \qquad R_\delta = \Delta R_q \qquad S_\delta = S_q \tag{39}$$

then the δ formulation for the optimal filter may be obtained from (33),(34) and (35) as:

$$\delta \hat{x}(k) = A_\delta \hat{x}(k) + H_\delta(k)[y(k) - C_\delta \hat{x}(k)] \tag{40}$$

where

$$H_\delta(k) = \frac{H_q(k)}{\Delta} = [P_\delta(k) C_\delta^T + \Delta A_\delta P_\delta(k) C_\delta^T + S_\delta][R_\delta + \Delta C_\delta P_\delta(k) C_\delta^T]^{-1} \tag{41}$$

and $P_\delta(k)$ satisfies:

$$\delta P_\delta(k) = Q_\delta + A_\delta P_\delta(k) + P_\delta A_\delta^T - H_\delta(k)[R_\delta + \Delta C_\delta P_\delta(k) C_\delta^T] H_\delta^T(k) + \Delta A_\delta P A_\delta^T \tag{42}$$

There are several points to note about this formulation.

1. This δ formulation for the optimal filter can be derived directly without going through the shift operator formulation first [15].

2. The formulation of the δ covariances in (39),(39) and (39) imply that they converge as $\Delta \to 0$ to the spectral densities of the continuous time processes. That is:

$$\lim_{\Delta \to 0} Q_\delta = Q \quad \lim_{\Delta \to 0} R_\delta = R \quad \lim_{\Delta \to 0} S_\delta = 0$$

3. The specification of the discrete time approximation to the continuous time process converges to the continuous time process as $\Delta \to 0$. That is

$$\lim_{\Delta \to 0} A_\delta = A \quad \lim_{\Delta = 0} C_\delta = C$$

4. Because of the above, the discrete Riccati difference equation (DRDE) in (42) converges to the CRDE in (30) and hence the discrete gain vector $H_\delta(k)$ tends to the continuous gain vector $H(t)$.

5. Because the δ operator formulation converges to the continuous time expressions as $\Delta \to 0$ we can use our generalised notation to express the Kalman filter for both continuous and discrete systems in a unified manner as:

$$\rho \hat{x} = A\hat{x} + H(y - C\hat{x}) \tag{43}$$

   where

$$H = [(\Delta A + I)PC^T + S][\Delta CPC^T + R]^{-1}$$

   By substituting (41) into (42) we get that P satisfies

$$\rho P = Q + PA^T + AP + PC^T(\Delta CPC^T + R)^{-1}CP + \mathcal{O}(\Delta)$$

   Here, A,P,C,S, and R are the δ subscripted versions defined earlier and we note that setting $\Delta = 0$ gives the continuous time solution.

Therefore, the δ operator formulation achieves the aim of preserving the innate link between continuous and discrete time in the context of state estimation. Furthermore, because of this smooth transition from discrete time results to continuous time results, at high sampling rates the solution of the DRDE and DRAE are better numerically conditioned when formulated using the δ operator rather the than the shift operator q [15].

For example, consider a system having transfer function:

$$G(s) = \frac{-s + 2}{(s + 2)(s + 10)}$$

with continuous state space model in observer form. The DRDE for this system was solved with A_δ and A_q rounded to 4 decimal places and $R = 1$, $Q = I$, $\Delta = 0.02$ and $P_0 = I$. Figure (6) shows the propagation of the relative error defined as:

$$\zeta_k = \frac{\|P_{FP}(k) - P(k)\|_F}{\|P(k)\|_F}$$

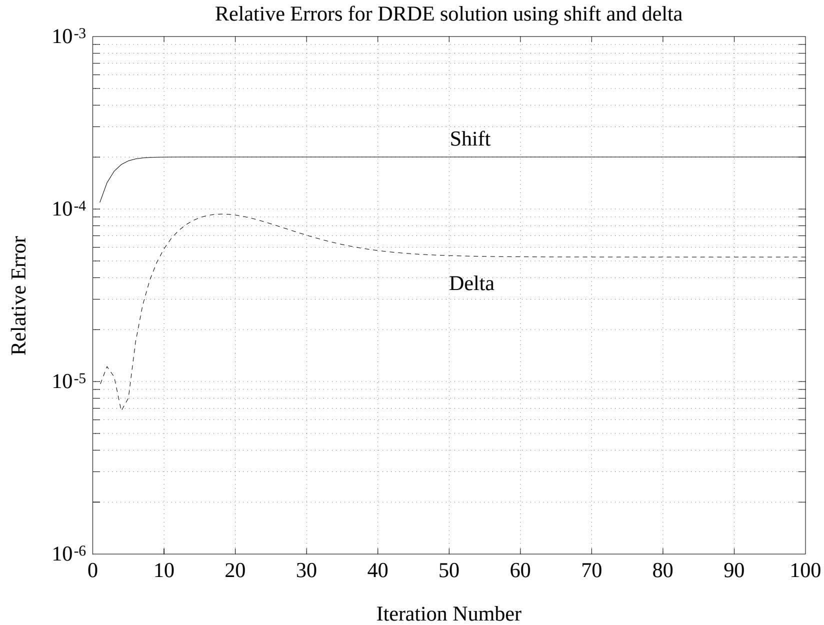

Figure 6: *Comparison of errors for solution of DRDE using δ and q operators*

where $P_{FP}(k)$ and $P(k)$ denote the floating point and 'infinite' precision solution to the Riccati equation and $\|.\|_F$ denotes the Fröbenius norm. Obviously the δ formulation leads to a significant improvement in the relative error in the computation of the DRDE. Furthermore, the levelling off of the error suggests that the solution of the DARE for this system will also be more precise using δ operator implementation. A different 2nd order example is considered in [15] where a more thorough discussion of these numerical considerations is given.

3.2 ARMAX Modelling

We now go on to motivate ARMAX modelling by showing it to be a convenient way of expressing the innovations form of the Kalman Filter given in (43). Defining the innovations process $\{\xi_t\}$ by:

$$\xi_t = y_t - C\hat{x}_t$$

then the steady state Kalman Filter of (43) can be written as:

$$\begin{aligned} \rho\hat{x}_t &= A\hat{x}_t + Bu_t + K\xi_t && (44) \\ y_t &= C\hat{x}_t + \xi_t && (45) \end{aligned}$$

Where K is the steady state value of H_t. Note that the innovations process $\{\xi_t\}$ is a white 'noise' process with spectral density D given by:

$$D = \Delta CPC^T + R$$

To convert (44) and (45) to a more convenient form for black box modelling we may assume, without loss of generality, that the model is in observer form. That is A,B and K are of the form:

$$A = \begin{bmatrix} -a_{n-1} & 1 & 0 & \cdots & 0 \\ -a_{n-2} & 0 & 1 & & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ \vdots & \vdots & & & 1 \\ -a_0 & 0 & \cdots & \cdots & 0 \end{bmatrix} \tag{46}$$

$$B^T = [b_{n-1}, b_{n-2}, \cdots, b_0] \tag{47}$$

$$K^T = [k_{n-1}, k_{n-2}, \cdots, k_0] \tag{48}$$

With this model structure, we can successively eliminate the state vector $\hat{x}_t$ to yield a model expressed purely in terms of input and output quantities. This yields the following left matrix fraction representation [6],[8]:

$$A(\rho)y_t = B(\rho)u_t + C(\rho)\xi_t \tag{49}$$

where

$$A(\rho) = a_n\rho^n + a_{n-1}\rho^{n-1} + \cdots + 1 \tag{50}$$

$$B(\rho) = b_{n-1}\rho^{n-1} + \cdots + b_0 \tag{51}$$

$$C(\rho) = c_n\rho^n + c_{n-1}\rho^{n-1} + \cdots + c_0 \tag{52}$$

with

$$c_i = k_i + a_i \qquad i \in [0, n-1]$$

This is the familiar ARMAX model for a plant [10] except for the oddity that we have chosen to normalise $A(\rho)$ from the right so that it is not monic as is usually the case. The reasons for this are discussed in the last section of this chapter.

3.3 Fractional Representations

It is common to write models for systems using fractional representations [3]

$$M(\rho)y_t = N(\rho)u_t + \nu_t \tag{53}$$

Here $M(\rho)$ and $N(\rho)$ belong to some desired class of transfer functions. A commonly desired class are those analytic and bounded in the right half plane so that $M(s)$ and $N(s)$ are in H_∞. We note that the ARMAX representation (49) representation just derived will not have $A(\rho)$ and $B(\rho)$ in this class. However, if we choose a Hurwitz polynomial $E(\rho)$ of the same order as $A(\rho)$ and also normalised from the right:

$$E(\rho) = e_n\rho^n + e_{n-1}\rho^{n-1} + \cdots + e_1\rho + 1 \tag{54}$$

then the simple choice

$$M(\rho) = \frac{A(\rho)}{E(\rho)} \qquad N(\rho) = \frac{B(\rho)}{E(\rho)}$$

will give a fractional representation for the ARMAX model (49) of the form (53) with $M(\rho)$ and $N(\rho)$ in H_∞. Other restrictions on $E(\rho)$ allow fractional representations for other classes of transfer functions to be found.

The advantage of this fractional representation is that it allows the innovations process $\{\xi_t\}$ to be expressed as a function of the plant input and output sequences $\{y_t\}$ and $\{u_t\}$ [3]. This allows the characterisation of all linear, unbiased estimates of the state of a plant to be expressed as an affine function of a free design variable [3],[12]. Consequently, the design of observers subject to min-max or H_∞ type constraints (as opposed to the quadratic constraint just used) is possible [12]. Because the topic of the existence of many linear unbiased estimates for the state has been brought up, we should note that the solution given in (43) is the best (in the quadratic sense of (27)) linear estimate of the state, and if the noise distributions are Gaussian is well known to be the best estimate of the state (linear or non-linear).

3.4 Parameter Estimation

Now that we have considered the problem of state estimation we will go on to consider parameter estimation. We will begin by deriving a parameter estimator from the Kalman filter state estimate just presented and then go on to consider least squares estimation.

3.4.1 Derivation from Kalman Filtering

For the sake of generality, we will include $E(\rho)$ in our ARMAX description of the Kalman filter (53) to give:

$$\frac{A(\rho)}{E(\rho)} y_t = \frac{B(\rho)}{E(\rho)} u_t + \frac{C(\rho)}{E(\rho)} \xi_t \tag{55}$$

Because of its connection with the Kalman filter $E(\rho)$ is commonly referred to as the observer polynomial. Notice that in discrete time theory, it is common to use $E = q^n$ to give $M(q)$ and $N(q)$ only involving backward time shifts. However, we will argue below that the choice $E = q^n$ is, in general, a poor one. It is easy to rearrange (55) into linear regression form by:

$$y_t = \left(\frac{E(\rho) - A(\rho)}{E(\rho)} \right) y_t + \frac{B(\rho)}{E(\rho)} u_t + \frac{C(\rho)}{E(\rho)} \xi_t \tag{56}$$

$$\Rightarrow \qquad y_t = \phi_t^T \theta_0 + \eta_t \tag{57}$$

where

$$\phi_t^T = \left[\frac{\rho^n y_t}{E(\rho)}, \cdots, \frac{\rho y_t}{E(\rho)}, \frac{\rho^m u_t}{E(\rho)}, \cdots, \frac{u_t}{E(\rho)} \right] \tag{58}$$

$$\theta_0^T = [e_n - a_n, \cdots, e_1 - a_1, b_m, \cdots, b_0] \tag{59}$$

Note due to the introduction of the observer polynomial $E(\rho)$ the elements of the regression vector ϕ_t are filtered derivatives. The condition that the parameter vector is time invariant may be expressed as;

$$\delta\theta_0 = 0 \tag{60}$$

Equations (60) and (57) are then precisely in the joint Markov model form of (37) and (38) with:

$$A_\delta = 0 \quad C_\delta = \phi_t^T \quad \omega_\delta(t) = \eta_t \quad \nu_\delta(t) = 0 \quad x(t) = \theta_0$$

For the moment, we assume that $C(\rho) = E(\rho)$ so that η_t represents a white noise sequence. The more general case of coloured noise will be taken up in section 3.7.3. The definitions on the noise imply:

$$Q_\delta = S_\delta = 0$$

and

$$R_\delta = \Delta\mathcal{E}\{\nu_t^2\} \triangleq \sigma_t^2 \tag{61}$$

Therefore, we can use the δ operator formulation of the optimal filter in (40), (41) and (42) to recursively calculate the minimum variance linear unbiased estimate of θ by;

$$\delta\hat{\theta}_k = \frac{P_k\phi_k(y_k - \phi_k^T\hat{\theta}_k)}{\sigma_k^2 + \Delta\phi_k^T P_k\phi_k} \tag{62}$$

$$\delta P_k = \frac{-P_k\phi_k\phi_k^T P_k}{\sigma_k^2 + \Delta\phi_k^T P_k\phi_k} \tag{63}$$

We note that since $\hat{\theta}$ is an unbiased estimate of θ_0, then the variance of the prediction error

$$\varepsilon_t = y_t - \hat{y}_t = y_t - \phi_t^T\hat{\theta}$$

will be given by:

$$\mathcal{E}\{\varepsilon_t^2\} = \phi_t^T P_t\phi_t - y_t^2$$

Therefore, since the Kalman filter gives the minimum variance estimate of $\hat{\theta}$, then for $\hat{\theta}'$ any other linear unbiased estimate with covariance P_t' we have

$$P_t \leq P_t'$$

in a matrix sense and hence the Kalman filter estimate $\hat{\theta}$ gives the predictor of minimum error variance.

3.4.2 Recursive Least Squares

We now approach the problem of parameter estimation from a different viewpoint. We formulate this by using our generalised notation and by defining a new cost function to be minimised:

$$J(\hat{\theta}) = \frac{1}{2}\left\{ \mathrm{S}_0^t \frac{1}{\sigma_\tau^2}(y_\tau - \phi_\tau^T\hat{\theta})^2 d\tau + (\hat{\theta} - \hat{\theta}_0)^T P_0^{-1}(\hat{\theta} - \hat{\theta}_0) \right\} \tag{64}$$

where

$$\begin{aligned}\hat{\theta}_0 &= \text{Some } \textit{a-priori} \text{ estimate for } \theta_0 \\ \{\sigma_t^2\} &= \text{Sequence of positive scalars} \\ P_0^{-1} &= \text{Positive definite symmetric matrix}\end{aligned}$$

The last term in (64) is a term reflecting *a-priori* information about what θ_0 might be and σ_t^2 in the first term in (64) weights the importance of the data in the cost function. If we wish to find the estimate $\hat{\theta}$ which minimises this cost function then it is appropriate to find the partial derivative of $J(\hat{\theta})$ with respect to $\hat{\theta}$;

$$\frac{\partial J(\hat{\theta})}{\partial \hat{\theta}} = \mathrm{S}_0^t \frac{\phi_\tau y_\tau}{\sigma_\tau^2} d\tau + P_0^{-1}\hat{\theta}_0 - \left(P_0^{-1} + \mathrm{S}_0^t \frac{\phi_\tau \phi_\tau^T}{\sigma_\tau^2} d\tau\right)\hat{\theta}$$

Setting this to zero then gives the modified least squares estimate in generalised notation as:

$$\hat{\theta}_t = P_t\left(P_0^{-1}\hat{\theta}_0 + \mathrm{S}_0^t \frac{\phi_\tau y_\tau}{\sigma_\tau^2} d\tau\right) \tag{65}$$

$$P_t^{-1} = P_0^{-1} + \mathrm{S}_0^t \frac{\phi_\tau \phi_\tau^T}{\sigma_\tau^2} d\tau \tag{66}$$

A recursive formulation may be found by applying the generalised derivative to both sides of (65):

$$\rho\hat{\theta}_t = (\rho P_t)\left(P_0^{-1}\hat{\theta}_0 + \mathrm{S}_0^t \frac{\phi_\tau y_\tau}{\sigma_\tau^2} d\tau\right) + P_t\frac{\phi_t y_t}{\sigma_t^2} + \frac{\Delta}{\sigma_t^2}(\rho P_t)\phi_t y_t \tag{67}$$

However, noting that

$$\rho P_t^{-1} = \frac{\phi_t \phi_t^T}{\sigma_t^2}$$

and applying Lemma 3 gives:

$$\rho P_t = \frac{-P_t\phi_t\phi_t^T P_t}{\sigma_t^2 + \Delta\phi_t^T P_t \phi_t} \tag{68}$$

Substituting this into (67) and rearranging then gives;

$$\rho\hat{\theta}_t = \frac{P_t\phi_t(y_t - \phi_t^T\hat{\theta}_t)}{\sigma_t^2 + \Delta\phi_t^T P_t\phi_t} \tag{69}$$

Some points to note about this solution are:

1. (68) and (69) are exactly the same as (62) and (63) that were derived from the Kalman filter if the sequence $\{\sigma_t^2\}$ is chosen as per (61).

2. If we have no *a-priori* information about θ_0, then we should set $\hat{\theta}_0 = 0$ and $P_0^{-1} = 0$. If we have no *a-priori* information about the noise process $\{\nu_t\}$ then we can arbitrarily set $\sigma_t^2 = 1 \quad \forall t \in R^+$. With these choices (65) and (66) become:

$$\hat{\theta}_t = P_t \int_0^t \phi_\tau y_\tau d\tau \tag{70}$$

$$P_t = \int_0^t \phi_\tau \phi_\tau^T d\tau \tag{71}$$

which in discrete time notation becomes:

$$\hat{\theta}_n = \left(\sum_{k=0}^{n-1} \phi_k \phi_k^T \right)^{-1} \sum_{k=0}^{n-1} \phi_k y_k \tag{72}$$

This is easily recognised as the standard least squares estimate of $\hat{\theta}$.

This completes our formulation of parameter estimation using our generalised notation. we now go on to consider the numerical properties of discrete time least squares estimation using δ operator formulation.

3.5 Conditioning of Least Squares Estimation

The expression (72) shows that least squares estimation implicitly involves calculating the solution to a linear system of equations which will be ill conditioned if the condition number of the covariance matrix $P_n = \sum_{k=0}^{n-1} \phi_k \phi_k^T$ is large. It is shown in [12] that in many cases, if the regression vector is derived from a δ operator ARMAX model rather than a shift operator one, then the condition number of the associated covariance matrix is lower. This effect is exacerbated for high model orders and for high sampling rates relative to the bandwidths of the plant input and output signals y_t and u_t.

As an example, consider the continuous time system used in the discussion on Kalman Filtering:

$$G(s) = \frac{-s+2}{(s+2)(s+10)}$$

Least squares as per (72) was then used to fit a fixed denominator model to this system using both q and δ parameterisation. In this case the regression vector was:

$$\phi_t^T = \left[\frac{\hat{b}_0 u_t}{\hat{A}(\xi)}, \frac{\hat{b}_1 \xi u_t}{\hat{A}(\xi)}, \cdots, \frac{\hat{b}_m \xi^m u_t}{\hat{A}(\xi)} \right]$$

Where ξ is either the δ of q operator. The sampling rate was ranged from 0.5 Hz to 50 Hz, the numerator model order m was chosen as 3, the input signal u_t was a 0.1 Hz fundamental square wave, and the observation record was 20 seconds long. Finally the fixed denominator for the shift and δ operator forms was chosen to be:

$$\hat{A}(\delta) = (\delta+4)^2(\delta+8)^2 \qquad \hat{A}(q) = (q-(1-4\Delta))^2(q-(1-8\Delta))^2$$

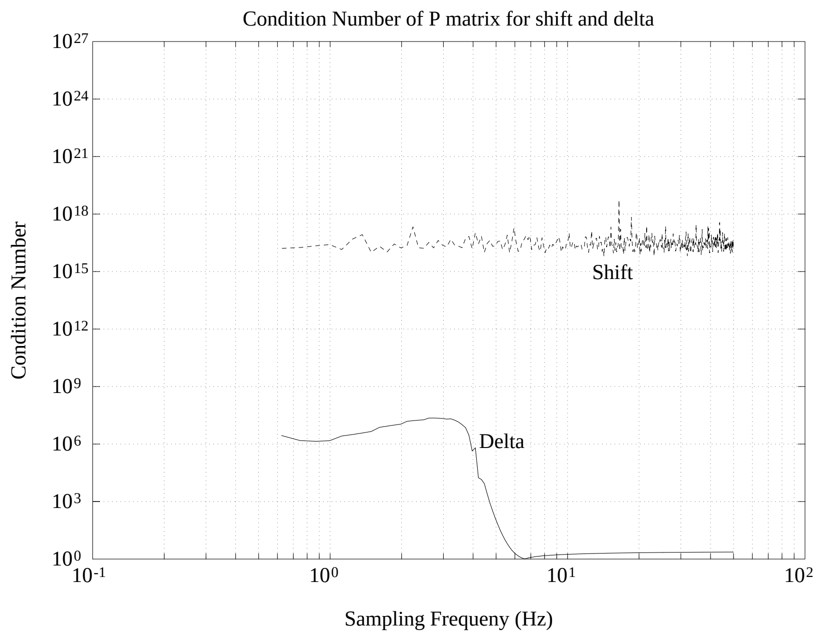

Figure 7: *Comparison of condition numbers for shift operator and δ operator covariance matrices.*

For each sampling frequency the condition numbers of the q and δ operator covariance matrices were calculated and are shown vs. sampling frequency in figure 7. Note that the P matrices were scaled to give 1's along the diagonal so that the condition numbers where more meaningful comparisons of the fixed point difficulty of inversion. As can be seen from figure 7, the normal equations are much better conditioned using δ operator implementation, especially at higher sampling rates.

3.6 Calculation of Regressors

In order to implement the recursive solution to parameter estimation it is necessary to have a method for calculating the elements of the regression vector ϕ, which was defined in (58). This is easily achieved by noting that the definition of $E(\rho)$ given in (54) implies:

$$E(\rho)y = e_n\rho^n y + e_{n-1}\rho^{n-1}y + \ldots + e_1\rho y + y$$

to give (dropping the explicit dependence on ρ):

$$\frac{\rho^n y}{E} = \left[-\left(\frac{e_{n-1}}{e_n}\right)\left(\frac{\rho^{n-1}y}{E}\right) - \ldots - \left(\frac{e_1}{e_n}\right)\left(\frac{\rho y}{E}\right) - \left(\frac{1}{e_n}\right)\left(\frac{y}{E}\right)\right] + \frac{y}{e_n}$$

Defining ψ and ϕ_y by:

$$\psi^T \triangleq \left[\frac{-e_{n-1}}{e_n}, \ldots, \frac{-e_1}{e_n}, \frac{-1}{e_n}\right]$$

$$\phi_y^T \triangleq \left[\frac{\rho^{n-1}y}{E}, \ldots, \frac{\rho y}{E}, \frac{y}{E}\right]$$

then gives:

$$\frac{\rho^n y}{E} = \psi^T \phi_y + \frac{y}{e_n}$$

Performing this similarly for the input $u(t)$ gives:

$$\frac{\rho^n u}{E} = \psi^T \phi_u + \frac{u}{e_n}$$

Where ϕ_u is defined similarly to ϕ_y. These equations can be written in state space form as:

$$\begin{aligned} \rho\phi_y &= \Upsilon\phi_y + \Sigma y \\ \rho\phi_u &= \Upsilon\phi_u + \Sigma u \end{aligned}$$

With the following definitions for Υ and Σ:

$$\Upsilon = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \\ -\frac{1}{e_n} & -\frac{e_1}{e_n} & \cdots & \cdots & -\frac{e_{n-2}}{e_n} & -\frac{e_{n-1}}{e_n} \end{bmatrix}$$

$$\Sigma^T = \left[0, \ldots, 0, \frac{1}{e_n}\right]$$

In discrete time, we interpret ρ as representing the discrete time operator δ to get the discrete time recursive update equations that are necessary to calculate the filtered derivatives of $y(t)$ and $u(t)$ that are required in the regression vector ϕ:

$$\begin{aligned} \delta\phi_y &= \Upsilon\phi_y + \Sigma y \\ \delta\phi_u &= \Upsilon\phi_u + \Sigma u \\ \frac{\delta^n y}{E} &= \psi^T \phi_y + \frac{y}{e_n} \end{aligned}$$

Which gives the difference equations:

$$\phi_y^{k+1} = \phi_y^k + \Delta\Upsilon\phi_y^k + \Delta\Sigma y_k \quad (73)$$

$$\phi_u^{k+1} = \phi_u^k + \Delta\Upsilon\phi_u^k + \Delta\Sigma u_k \quad (74)$$

$$\left(\frac{\delta^n y}{E}\right)_{k+1} = \psi^T \phi_y^{k+1} + \frac{y_{k+1}}{e_n}$$

All the filtered derivatives of the regression vector ϕ are thus known from these equations and so ϕ may be formed from the calculated elements by:

$$\phi_k^T = \left[\left(\frac{\delta^n y}{E} \right)_k, \phi_y^k(1), \ldots, \phi_y^k(n-1), \phi_u^k(n-m-1), \ldots, \phi_u^k(n) \right]$$

Note that not all the terms in ϕ_y and ϕ_u are used in forming the regression vector ϕ. Specifically, the term $\frac{y}{E}$ in ϕ_y is not used in ϕ and the terms of higher order than $\frac{\rho^m u}{E}$ in ϕ_u are not used in ϕ.

3.7 Altering the Dynamic Behaviour of RLS

Consideration of (68) shows that $P_k = 0$ is a solution of the difference equation for the update of P_k. Consequently, for RLS of P_k given by (68) we get $P_k \to 0$. Consideration of (69) shows that this means that the algorithms will not be able to track time varying parameter changes. In order to overcome this, we need to modify the dynamic behaviour of RLS. Here we discuss two algorithms for doing this namely the gradient and constant trace schemes.

3.7.1 Gradient Algorithm

The most obvious solution to the problem is to fix the covariance matrix so that it cannot tend to $\vec{0}$. The simplest constant to use for the covariance matrix is some multiple of the identity matrix. This then leads to the following recursive update scheme (we have dropped the explicit dependence on time):

$$\begin{aligned} \rho\hat{\theta} &= \frac{\alpha\phi e}{1 + \Delta\alpha\phi^T\phi} \\ P &= \alpha I \end{aligned}$$

This solution is only a crude one to the problem of tracking time varying plants. Specifically, it slows down the rate of parameter convergence. To see this consider first the RLS update scheme (ie. not the gradient scheme). In this case the covariance matrix P is updated. Considering the inverse of this matrix and the parameter estimation error $\tilde{\theta}$:

$$\tilde{\theta} \triangleq \hat{\theta} - \theta_0$$

Lemma 2 can be used to find the generalised derivative of their product:

$$\rho(P^{-1}\tilde{\theta}) = (\rho P^{-1})\tilde{\theta} + P^{-1}\rho\tilde{\theta} + \Delta(\rho P^{-1})(\rho\tilde{\theta}) \tag{75}$$

Noting that

$$\rho\tilde{\theta} = \frac{-P\phi\phi^T P}{1 + \Delta\phi^T P\phi} \tag{76}$$

$$\rho P^{-1} = \phi\phi^T \tag{77}$$

and substituting into (75) gives:

$$\rho(P^{-1}\tilde{\theta}) = \phi\phi^T\tilde{\theta} + P^{-1}\left(\frac{-P\phi\phi^T P}{1+\Delta\phi^T P\phi}\right) + \Delta\phi\phi^T\left(\frac{-P\phi\phi^T P}{1+\Delta\phi^T P\phi}\right) \tag{78}$$

consider also the quadratic form:

$$V(t) = \tilde{\theta}^T(t)P^{-1}\tilde{\theta}(t)$$

Using Lemma 2 gives the generalised derivative of $V(t)$ as:

$$\rho V = (\rho\tilde{\theta}^T)P^{-1}\tilde{\theta} + \tilde{\theta}^T\rho(P^{-1}\tilde{\theta}) + \Delta(\rho\tilde{\theta}^T)\rho(P^{-1}\tilde{\theta})$$

Using (77),(78) then results in:

$$\rho V = -\tilde{e}_t^2 \tag{79}$$

Where $\tilde{e}_t$ is the normalised prediction error defined as:

$$\tilde{e}_t \triangleq \frac{y - \phi^T\hat{\theta}}{\sqrt{1+\Delta\phi^T P\phi}}$$

Defining an associated normalised cost function:

$$\tilde{C}(\hat{\theta},t) \triangleq \mathop{\mathrm{S}}_0^t \tilde{e}_t^2$$

and applying the generalised integral to both sides of (79) gives:

$$V(t) = -\mathop{\mathrm{S}}_0^t \tilde{e}_t^2 = -\tilde{C}(\hat{\theta},t)$$

Note that as $\Delta \to 0$ the normalised cost function tends towards the least squares cost function $C(\hat{\theta},t)$. Therefore, the equation:

$$V(t) = (\hat{\theta}-\theta_0)^T P^{-1}(\hat{\theta}-\theta_0) = -\tilde{C}(\hat{\theta},t) \tag{80}$$

describes a hyper-ellipsoid centred around the true parameter values. Furthermore, the 'size' of the hyper-ellipsoid is proportional to the value of the normalised cost function. That is, for a certain normalised least squares cost function value, all the possible estimates of the plant that could achieve that cost function value must lie on the surface of the hyper-ellipsoid. The two dimensional case is shown diagrammatically in figure 8. Note that the vector:

$$\tilde{\theta} = \left(\sqrt{\frac{\tilde{C}(\hat{\theta},t)}{\phi^T P\phi}}\right) P\phi$$

is a solution of the ellipsoid equation (80). That is, given a particular parameter estimate $\hat{\theta}$, projection should be made in the direction $P\phi$ in order to arrive at the true parameter value θ_0. Projection such as this would require a parameter update of the form:

$$\rho\hat{\theta} = \alpha P\phi \qquad \alpha \text{ a scalar}$$

Note that this is precisely what the unmodified least squares algorithm does. Because P is not updated with the gradient scheme the projection of $\hat{\theta}$ is not in the correct straight line direction towards θ_0 and hence convergence is slower than for RLS.

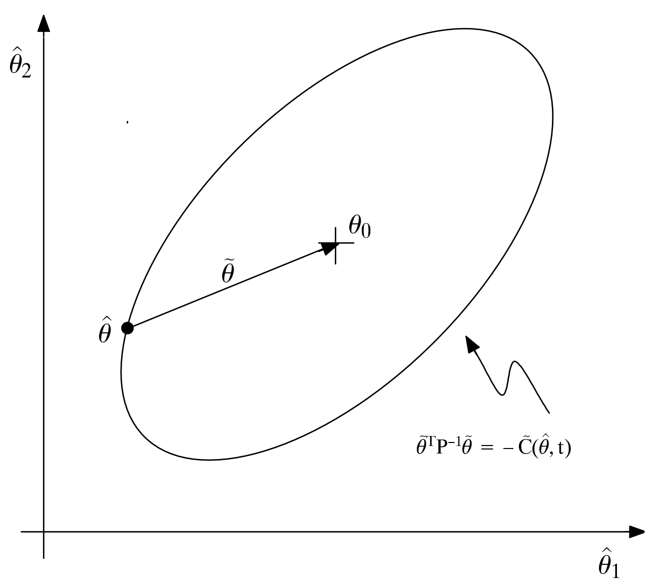

Figure 8: *Ellipse of possible estimates giving the same cost function value*

3.7.2 Constant Trace Algorithm

A solution to the problem of the covariance matrix going to zero, whilst not fixing it and thus losing valuable search direction information, is to simply multiply it by an appropriate constant. Specifically, a possible solution is to update the covariance matrix using the basic RLS, and then multiply the matrix by an appropriate scalar determined so as to fix the trace of the matrix at a specified value. This scheme will give the constant trace algorithm which may be written

$$\begin{aligned}
\hat{\theta}_{k+1} &= \hat{\theta}_k + \frac{\Delta P_k \phi_k e_k}{1 + \Delta \phi_k^T P_k \phi_k} \\
P_{k+1} &= \xi \left(P_k - \frac{\Delta P_k \phi_k \phi_k^T P_k}{1 + \Delta \phi_k^T P_k \phi_k} \right) \\
\xi &= \frac{\text{Trace}(P_0)}{\text{Trace}(P_{k+1})} \\
e_k &= y_k - \phi_k^T \hat{\theta}_k
\end{aligned}$$

Note that scaling P by a scalar does not distort the search direction information P contains, but it does prevent P tending to zero. Other related algorithms are described in [16].

3.7.3 Dealing with Coloured Noise

The linear regression model (57) that was derived from the Kalman filter motivated ARMAX model (49) involves a measurement noise process $\{\eta_t\}$. This is derived from the white 'noise' innovations process $\{\xi_t\}$ via:

$$\eta_t = \frac{C(\rho)}{E(\rho)} \xi_t$$

Therefore, $\{\eta_t\}$ will in general not be a white noise process. Consequently, since both ϕ_t and ξ_t will then depend on past data, they will be correlated, Therefore, the normal least squares estimate $\hat{\theta}_t$ will not be consistent [5]. There are a number of ways of overcoming this bias in $\hat{\theta}$.

1. If $C(\rho)$ is known then putting $E(\rho) = C(\rho)$ gives the linear regression form (57) as:

$$y_t = \left(\frac{C - A}{C} \right) y_t + \frac{B}{C} u_t + \xi_t \tag{81}$$

   This gives the white measurement noise process we require

2. If $C(\rho)$ is not known, then $C(\rho)$ can be estimated from the regression model (81). However, this will be a non-linear optimisation problem. This implies non-unique minima to the least squares cost function (64). Therefore $\hat{\theta}$ may converge to a local minima and give worse bias than if no steps were taken. Furthermore, it is usually necessary to project $\hat{C}(\rho)$ to ensure that it is stable [6].

3. If appropriate instruments are known, then the instrumental variable method may be used to obtain a consistent estimate [17].

4. A Pseudo-Linear regression may be used. That is, the ARMAX model (49) may be written in the regression form:

$$y_t = \left(\frac{E-A}{E}\right) y_t + \frac{B}{E} u_t + \left(\frac{C-E}{E}\right) \xi_t + \xi_t$$

This is a linear regression model of the form (57) with white noise where

$$\begin{aligned} \phi^T &= \left[\frac{\rho^n y_t}{E}, \cdots, \frac{\rho y_t}{E}, \frac{\rho^m u_t}{E}, \cdots, \frac{u_t}{E}, \frac{\rho^n \xi_t}{E}, \cdots, \frac{\xi_t}{E}\right] \\ \theta^T &= [e_n - a_n, \cdots, e_1 - a_1, b_m, \cdots, b_0, c_n - e_n, \cdots, c_0 - e_0] \end{aligned}$$

This is not in a form suitable for parameter estimation since ϕ depends on the unmeasured innovations process $\{\xi_t\}$. The pseudo-linear regression method circumvents this by replacing ξ_t by an on-line estimate $\hat{\xi}_t$:

$$\hat{\xi}_t = y_t - \phi_t^T \hat{\theta}_t$$

This method provides a consistent estimate for θ so long as $E(\rho)$ and $C(\rho)$ satisfy a positive real condition [6],[17],[4].

5. We can try to linearize the estimation of $C(\rho)$ proposed in 2. This can be done by writing $C^{-1}(\rho)$ as a power series and then approximating $C^{-1}(\rho)$ by truncation of the series at r terms:

$$C^{-1}(\rho) \approx \sum_{k=1}^{r} c_k \rho^{-k} \tag{82}$$

Operating on both sides of (49) when $\rho = q$ then shows this to be the well known idea of modelling MA processes with AR approximations:

$$A'(q^{-1}) y_k \approx B'(q^{-1}) u_k + \xi_k$$

Where

$$A'(q^{-1}) = A(q^{-1}) q^n \sum_{k=1}^{r} c_k q^{-k} \qquad B'(q^{-1}) = B(q^{-1}) q^n \sum_{k=1}^{r} c_k q^{-k}$$

We can now perform estimation with approximately white measurement errors. Note that the resultant model estimate is non-minimal.

We intend to pursue this last method here, but suggest the use of *a priori* knowledge of $C(\rho)$ in order to improve the performance of the method. Specifically, we note that the convergence of the power series (82) depends on the location of the zeroes of $C(\rho)$. If these are close to the stability boundary, then the convergence will be slow and large orders of $A'(\rho)$ and $B'(\rho)$ will be required to provide approximately white errors.

Therefore, we propose that $C(\rho)$ be expanded not in terms of ρ^{-1}, but in terms of $(\rho + e)^{-1}$, where $-e$ is chosen by prior knowledge to be close to the zeroes of $C(\rho)$ so that the expansion will converge quickly. This idea is summarised in the following Lemma.

Lemma 4. *Consider the stochastic operator model (55). Provided we know an $e \in (0, \frac{1}{\Delta}]$ such that the zeroes γ_i of $C(\rho)$ satisfy:*

$$|\gamma_i + e| < e \qquad \forall i \in [1, n] \tag{83}$$

then $\forall \varepsilon > 0$ there exists a stable operator $E'(\rho)$ such that (55) can be expressed as

$$\frac{A'(\rho)}{E'(\rho)} y_t = \frac{B'(\rho)}{E'(\rho)} u_t + \xi_t + \xi'_t$$

where $\{\xi_t\}$ is a white noise process and ξ'_t has variance less than ε.

Proof. For the sake of simplicity, assume that $C(\rho)$ has no repeated zeroes so that by partial fraction expansion we may write:

$$\begin{aligned} \frac{E(\rho)}{C(\rho)} &= \frac{E(\rho)}{\prod_{i=1}^{n}(\rho + \gamma_i)} \\ &= 1 + \sum_{i=1}^{n} \left(\frac{\alpha_i}{\rho + \gamma_i} \right) \\ &= 1 + \sum_{i=1}^{n} \left(\frac{\alpha_i}{\rho + e} \right) \left[\frac{1}{1 + \left(\frac{\gamma_i - e}{\rho + e} \right)} \right] \end{aligned} \tag{84}$$

The term in square brackets may be expanded via a power series as [9]:

$$\frac{1}{1 + \left(\frac{\gamma_i - e}{\rho + e} \right)} = \sum_{j=0}^{\infty} \left(\frac{e - \gamma_i}{\rho + e} \right)^j$$

By the Ratio Test [9], this power series is convergent for:

$$\left| \frac{e - \gamma_i}{\rho + e} \right| < 1 \Rightarrow |\gamma_i - e| < e \qquad \forall i \in [1, n] \quad \forall \rho \in [0, \frac{1}{\Delta}]$$

That is, e must be 'closer to' the γ_i's than to the origin. This may be satisfied if e is bigger than all the γ_i's are. The closer e is to the γ_i's, the faster the power series will converge. If we truncate this power series at m terms, then we may write:

$$\begin{aligned} \frac{1}{1 + \left(\frac{\gamma_i - e}{\rho + e} \right)} &= \sum_{j=0}^{m} \left(\frac{e - \gamma_i}{\rho + e} \right)^j + R(\rho) \\ &\triangleq \frac{F(\rho)}{(\rho + e)^m} + R(\rho) \end{aligned}$$

Where $R(\rho)$ is a remainder term. Substituting this into (84) gives:

$$\frac{1}{C(\rho)} = \frac{1}{E(\rho)} \left(1 + \frac{F(\rho)}{(\rho + e)^m} + R(\rho) \right)$$

Motivated by this, define:

$$\frac{F'(\rho)}{E'(\rho)} \triangleq \frac{1}{E(\rho)}\left(1+\frac{F(\rho)}{(\rho+e)^m}\right) \approx \frac{1}{C(\rho)}$$

In this case, $E'(\rho) \triangleq E(\rho)(\rho+e)^m$ and $F'(\rho)$ is of order m. Operating on both sides of (57) by this stable operator gives:

$$A'\left(\frac{y}{E'}\right) = B'\left(\frac{u}{E'}\right) + \nu - \left(\frac{HR}{E}\right)\nu$$

Where $A'(\rho) \triangleq A(\rho)F'(\rho)$ and $B'(\rho) \triangleq B(\rho)F'(\rho)$. $\nabla\nabla\nabla$

This Lemma suggests a paradigm of adding extra zeroes to the observer polynomial $E(\rho)$, appropriately chosen by *a-priori* knowledge and (83), and then by fitting an appropriately higher order model to the process. Some points to note about this method are;

1. The model used is non-minimal. However, because of the normalisation of $A(\rho)$ and $E(\rho)$ from the right, the extraction of an appropriate minimal order estimate is easy since the high order terms go to zero as the power series converges. Such is not the case if normalisation from the left is used to force the high order term to be 1.

2. The convergence condition in (83) is precisely the positive real condition necessary for the pseudo-linear regression method 4 to converge [6],[17]. This highlights the fact that this expansion method essentially involves the estimation of $C(\rho)$.

To illustrate this discussion of noise an example is now presented. The following discrete time system was simulated:

$$\begin{aligned}(\delta+2)y_k &= 2u_k + (\delta+2)\nu_k \\ \Rightarrow y_k &= \phi^T\theta + \left(\frac{\delta+2}{E(\delta)}\right)\nu_k\end{aligned}$$

The sampling rate used was 30 Hz and ν_k was a white Gaussian distributed process with variance $\sigma^2 = 0.01$. A Constant Trace estimator was run for 5 seconds with the trace set to 100. Initially the observer polynomial was set to $E(\delta) = (\delta+5)$ so that the noise error was coloured. As expected, this produced a bias in the estimation of the parameters with the identified model being:

$$\hat{G}(\delta) = \frac{0.55}{0.14\delta+1}$$

This is not very close to the true model of $\frac{1}{0.5\delta+1}$. Figure 9 shows the evolution of the estimates together with a comparison between the true and estimated frequency responses. As can be seen, there is a large bias error. Note also that the parameter estimates vary with input changes. This is typical of undermodelled behaviour.

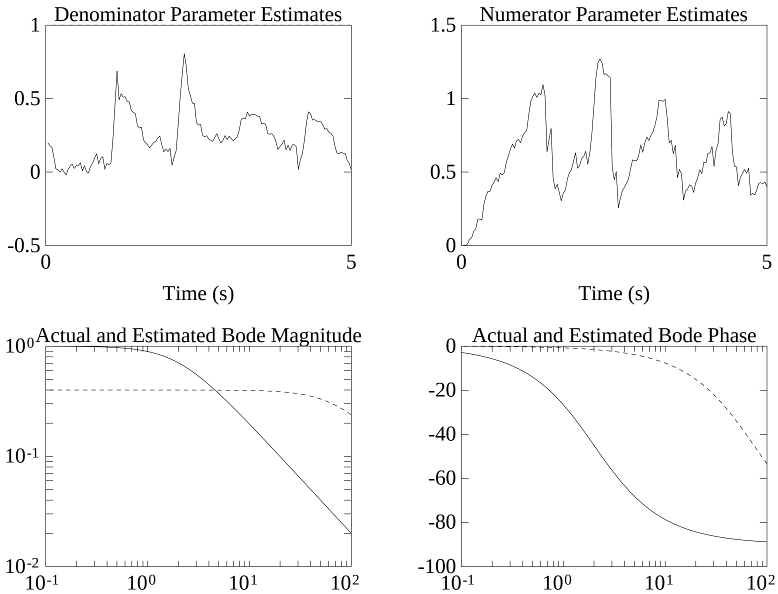

Figure 9: *Evolution of Estimates, and comparison of frequency responses of true and estimated plants for* $E(\delta) = (\delta + 5)$

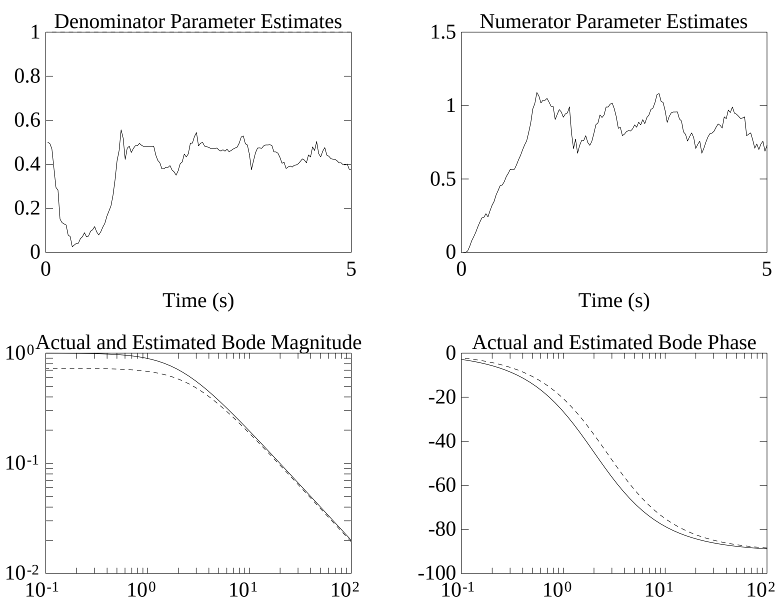

Figure 10: *Evolution of Estimates, and comparison of frequency responses of true and estimated plants for* $E(\delta) = (\delta + 2)$

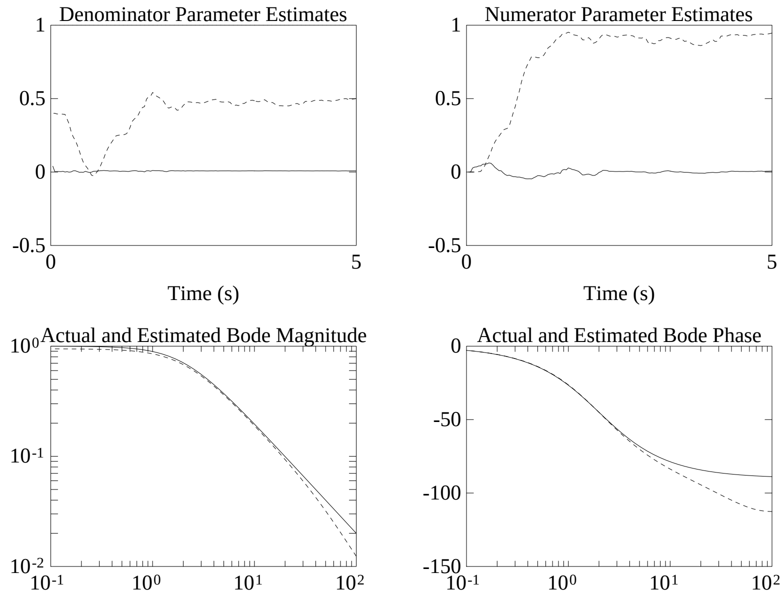

Figure 11: *Evolution of Estimates, and comparison of frequency responses of true and estimated plants for* $E(\delta) = (\delta + 5)^2$, $m = 1$

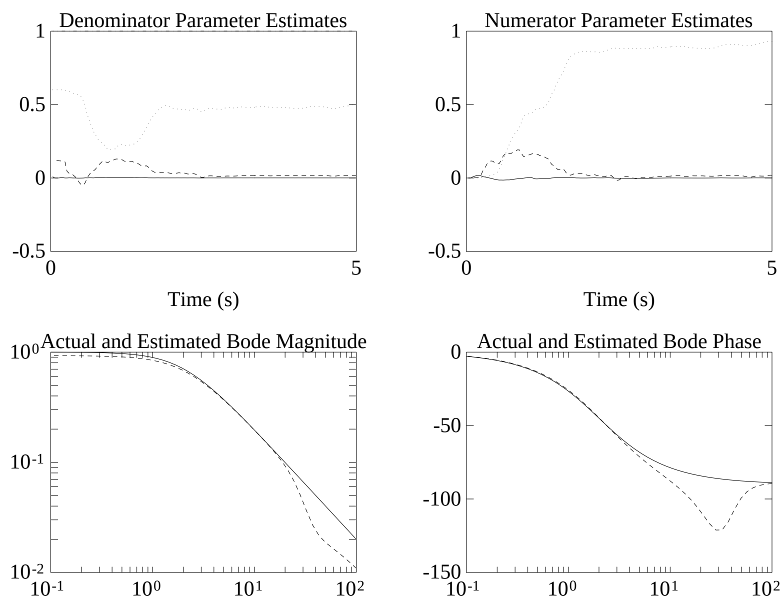

Figure 12: *Evolution of Estimates, and comparison of frequency responses of true and estimated plants for* $E(\delta) = (\delta + 5)^3$, $m = 2$

The same estimator was then run with $E(\delta) = (\delta + 2)$ so that the discrete time simulation became:

$$\begin{aligned} y_k &= \phi^T \theta + \left(\frac{\delta + 2}{\delta + 2} \right) \nu_k \\ &= \phi^T \theta + \nu_k \end{aligned}$$

In this case the noise error is white, and so by the Kalman optimal filter properties, the estimated plant should be unbiased. This was found to be the case with the estimates being:

$$\hat{G}(\delta) = \frac{0.83}{0.41\delta + 1}$$

This is quite close to the true plant as the frequency response curves in figure 10 show. This illustrates the use of method 2. If the spectral properties are not known, then the expansion method of Lemma 4 may be if the order of $E(\delta)$ and consequently $\hat{G}(\delta)$ are extended by m. The case $m = 1$ was simulated with $E(\delta) = (\delta + 5)^2$ and the estimated plant arrived at was:

$$\hat{G}(\delta) = \frac{-0.012\delta + 0.87}{0.009\delta^2 + 0.43\delta + 1} \approx \frac{0.87}{0.43\delta + 1}$$

This estimate is quite close to the true plant as the frequency response comparison in figure 11 shows. The case $m = 2$ was also simulated with $E(\delta) = (\delta + 5)^3$ to give an estimate of:

$$\hat{G}(\delta) = \frac{0.0007\delta^2 + 0.0086\delta + 0.87}{0.0007\delta^3 + 0.019\delta^2 + 0.46\delta + 1} \approx \frac{0.87}{0.46\delta + 1}$$

The frequency response comparison in figure 12 shows a negligible improvement over the case $m = 1$ for this example. Note how, with normalisation from the right, it is easy to extract the minimal order subsystem from the non-minimal estimate.

3.7.4 Effect of Normalising the Plant Model from the Right

In the analysis presented so far, it has been assumed that, apart form measurement noise, the system response can be exactly described by an nth order system parameterised by θ_0. In practice, however, all systems are infinite dimensional, and all we can hope to do is find and order n which allows approximate modelling of the system. The practice of normalising the plant model from the right is particularly amenable to this problem since it allows a paradigm of fitting a high order model to the system. If this order turns out to be too high then the high order parameters are estimated as zero. Such is not the case it the nominal model is normalised from the left and the highest order parameter is fixed at 1.

To illustrate this paradigm the following model:

$$G(\delta) = \frac{\hat{b}_1\delta + \hat{b}_0}{\hat{a}_3\delta^3 + \hat{a}_2\delta^2 + \hat{a}_1\delta + 1}$$

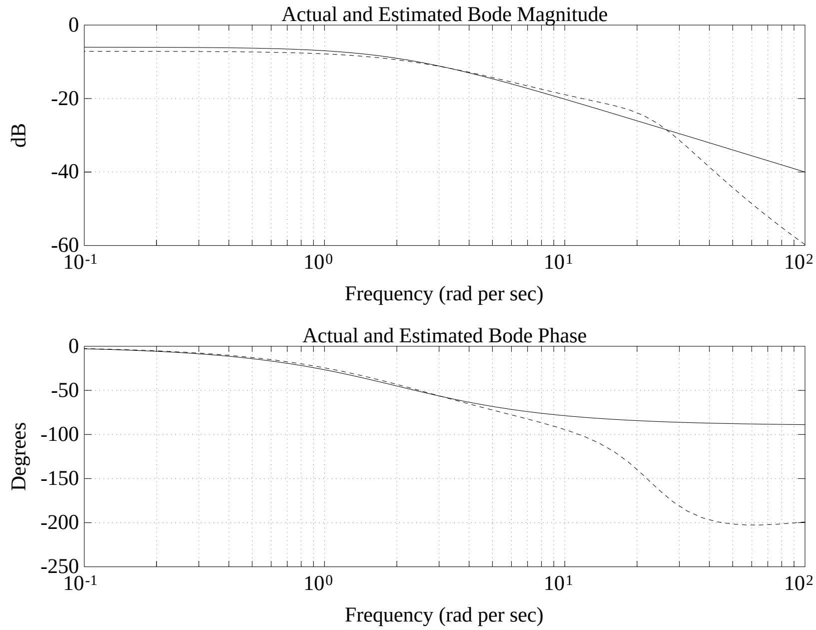

Figure 13: *Results of estimation for* $G_1(s)$.

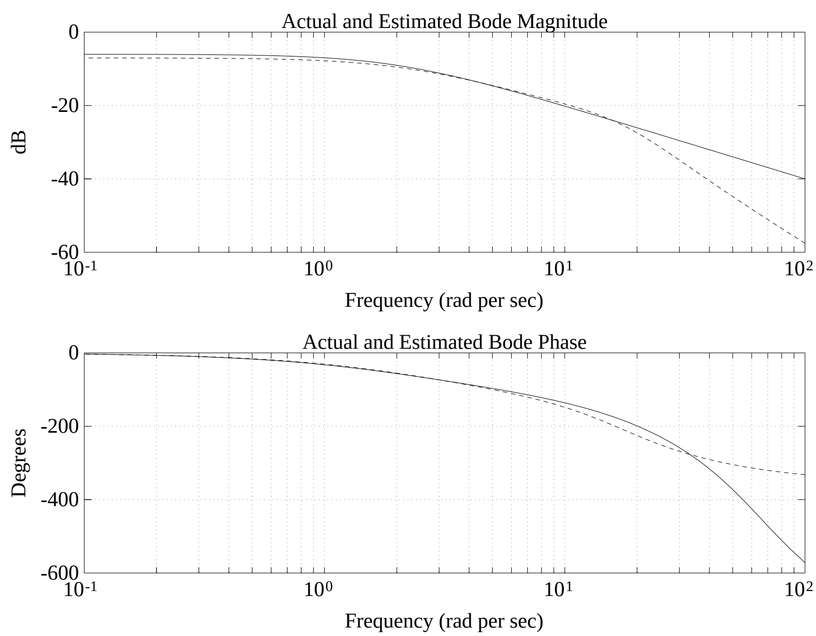

Figure 14: *Results of estimation for* $G_2(s)$.

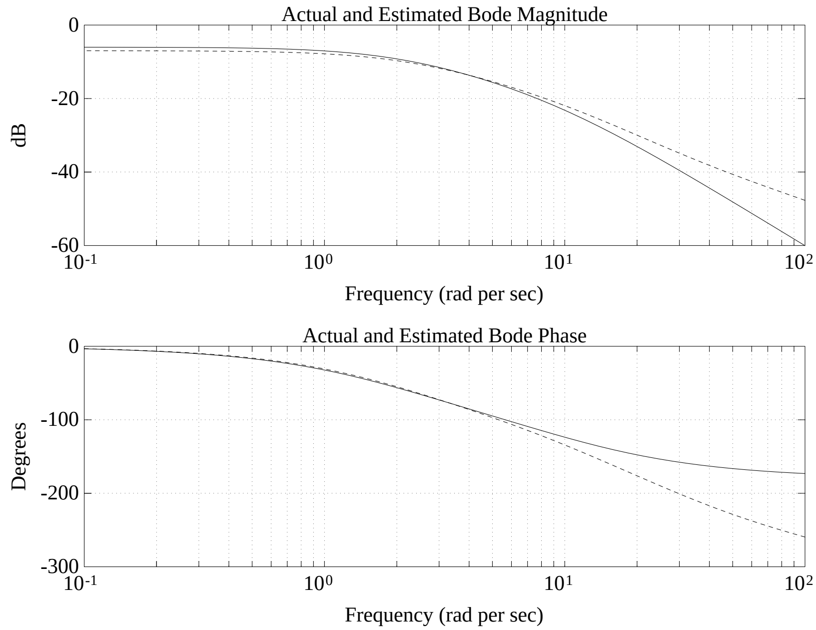

Figure 15: Results of estimation for $G_3(s)$.

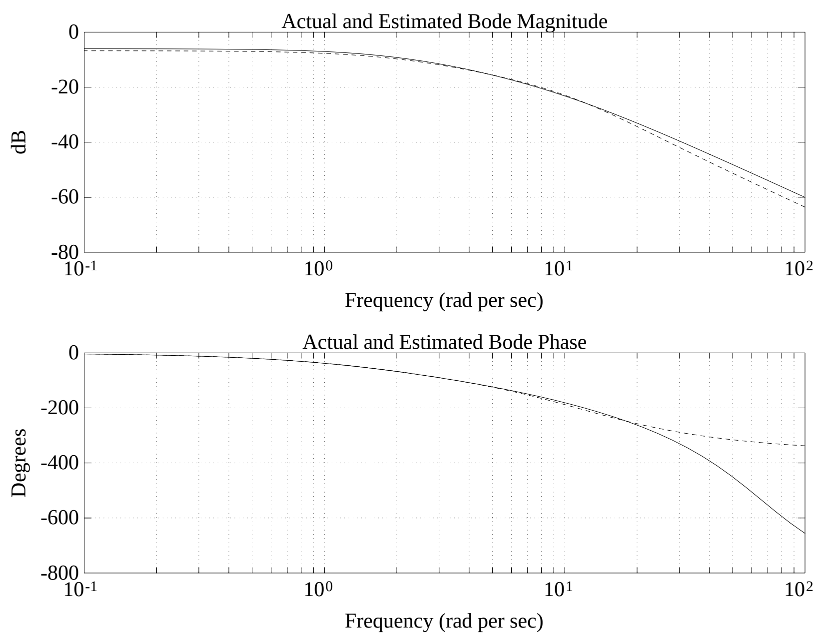

Figure 16: *Results of estimation for* $G_4(s)$.

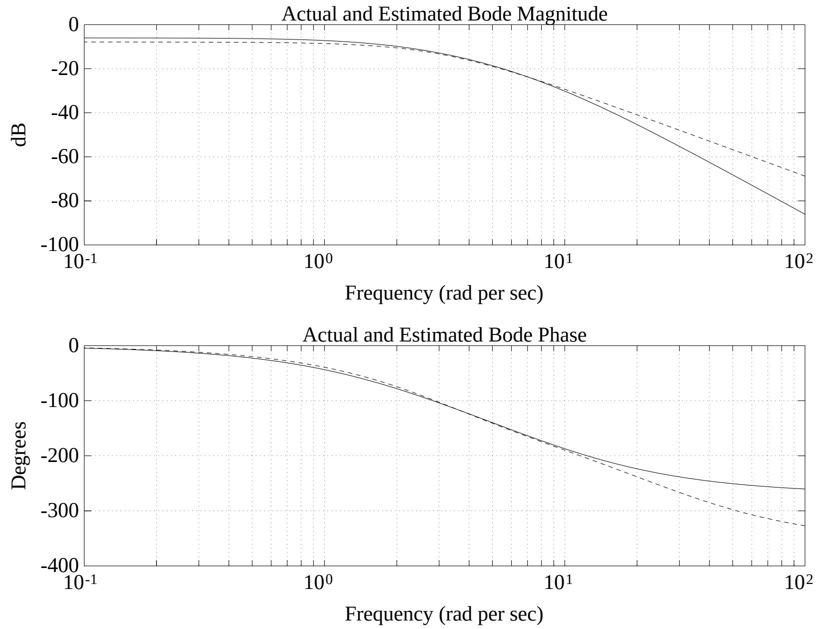

Figure 17: *Results of estimation for $G_5(s)$.*

was fitted to the following five plants:

$$G_1(s) = \frac{2}{s+2} \quad G_2(s) = \frac{2e^{0.1s}}{s+2} \quad G_3(s) = \frac{2}{(s+2)(0.1s+1)}$$

$$G_4(s) = \frac{2e^{0.1s}}{(s+2)(0.1s+1)} \quad G_5(s) = \frac{2}{(s+2)(0.1s+1)(0.2s+1)}$$

A sampling frequency of 30 Hz was used in the simulations. The plant excitation signal used was a 1.5 radian per second square wave. A Constant Trace Least Squares identification algorithm was used with the trace set to 100 and an observer polynomial of $E(\delta) = (\delta + 10)^3$ was used. The results are shown in figures 13 to 17. In each of the figures the true and estimated Bode magnitude and phase plots are shown. As can be seen the fixed third order model is very successfully fitted to all five different order plants. The comparison of frequency responses show that in all cases a good fit is found over a 10 radian per second range. This suggests that a method of fitting a high order model to a plant response would be useful in finding a smooth estimate of the frequency response of an unknown order plant. This could replace the common method of dividing the sample cross correlation between the plant output signal and the plant input signal by the sample autocorrelation function of the plant input signal since this method suffers from providing non-smooth estimates of frequency response and also of being sensitive to noise [10].

4 Conclusion

We have provided a new perspective on linear estimation by formulating our algorithms in a new generalised notation that provides both the discrete time, and the appropriate continuous time result simultaneously. Central to this has been the introduction of a new discrete time operator, the δ operator that is derived from a linear transformation of the familiar shift operator q. The major conceptual advantage of this new operator is that, unlike the q operator case, discrete time formulations do not ignore the fact that they stem from an underlying continuous time process. Consequently, the problem of having to juggle two different domains, the continuous and discrete, disappears.

We have shown that as well as the conceptual benefit of linking the continuous and discrete domains more closely, the δ operator allows much more numerically robust formulations of some key digital signal processing algorithms. In particular, we highlighted the the improved numerical conditioning of Kalman filter state estimation calculations, and the improved conditioning of the normal equations involved in least squares parameter estimation.

We concluded with a discussion of some practical aspects of parameter estimation, not limited to δ operator formulations. We showed how the dynamic properties of RLS could be improved to track time varying plants. We showed how the problem of measurement noise correlated with the regression vector could be overcome to avoid biased estimates, and we showed how the problem of parameter estimation on systems of unknown order could be approached with a simple paradigm.

REFERENCES

[1] K.J. Åstrom. *Introduction to Stochastic Control Theory.* Academic Press, New York, 1970.

[2] William D.Stanley, Gary R.Dougherty, and Ray Dougherty. *Digital Signal Processing.* Reston Publishing Company, Inc., Reston, Virginia, second edition, 1984.

[3] B.A. Francis. *A Course in H_∞ Control Theory.* Springer-Verlag, Berlin, 1987.

[4] G.C. Goodwin. Some observations on robust stochastic estimations. *Plenary Address, IFAC Symposium on System Identification and Parameter Estimation, Beijing*, 1988.

[5] G.C. Goodwin and R.L. Payne. *Dynamic System Identification.* Academic Press, 1977.

[6] G.C. Goodwin and K.W. Sin. *Adaptive Filtering Prediction and Control.* Prentice-Hall, Inc., New Jersey, 1984.

[7] F.B. Hilderbrand. *Introduction to Numerical Analysis.* McGraw Hill, New York, 1956.

[8] T. Kailath. *Linear Systems.* Prentice Hall, New Jersey, 1980.

[9] E. Kreyszig. *Advanced Engineering Mathematics.* John Wiley and Sons, London, 1983.

[10] Lennart Ljung. *System Identification: Theory for the User.* Prentice-Hall, Inc., New Jersey, 1987.

[11] R.H. Middleton and G.C. Goodwin. Improved finite word length characteristics in digital control using delta operators. *IEEE.Transactions on Automatic Control*, AC-31(11):1015–1021, 1986.

[12] R.H. Middleton and G.C. Goodwin. *Digital Estimation and Control: A Unified Approach.* Prentice-Hall, Inc., New Jersey, 1990.

[13] L.R. Rabiner and B. Gold. *Theory and Application of Digital Signal Processing.* Prentice Hall, New Jersey, 1975.

[14] M.E. Salgado. *Issues in Robust Identification.* PhD thesis, University of Newcastle, 1989.

[15] M.E. Salgado, C.E. de Souza, and G.C. Goodwin. Qualitative aspects of the distribution of errors in least squares estimation. *AUTOMATICA*, Special Issue on System Identification(7), January 1990.

[16] M.E. Salgado, G.C. Goodwin, and R.H. Middleton. A modified least squares algorithm incorporating exponential setting and forgetting. *International Journal of Control*, 47(2):477–491, 1988.

[17] T.Söderström and P.Stoica. *System Identification.* Prentice Hall, New York, 1989.

[18] J.H. Wilkinson. *Rounding Errors in Algebraic Processes.* Prentice Hall, New Jersey, 1963.