Recursive algorithms for estimating the parameters in a one dimensional heat diffusion system: derivation and implementation

Bharath Bhikkaji*, Torsten Söderström* and Kaushik Mahata*

Abstract

In this paper we consider a one-dimensional heat diffusion system, which is modeled by a partial differential equation (PDE) involving some unknown parameters. Here we estimate these unknown parameters recursively using a frequency domain approach.

Keywords: Diffusion, LTI-System, Recursive algorithm, FFT.

*Division of Systems and Control, Department of Information Technology, Uppsala University, P O Box 337, SE 75105, Uppsala, Sweden, email bb@it.uu.se, ts@it.uu.se, km@it.uu.se

1 Introduction

In the recent past, see [7] and [8], we have considered the problem of estimating the parameters of a one dimensional heat diffusion system. In the above mentioned papers, we have estimated the parameters using the standard output error method, see [9], in the time domain. Here, we device a recursive algorithm in the frequency domain to estimate the parameters involved.

Recursive identification algorithms, as opposed to batch estimation algorithms, are procedures where one estimates the parameters sequentially in time (i.e, as and when the data arrives) and avoids the reuse of old data. Advantages of recursive algorithms over batch algorithms are that, they involve limited storage costs, smaller computational complexity, and comes in handy when the parameters are time-varying. For a more detailed discussion on recursive algorithms we refer to [3].

Most of the recursive identification algorithms are defined in the time domain, as there exists a natural ordering in that domain. In some cases, like the one considered here, due to simplicity in the expressions it may be technically easier to handle the estimation problem in the frequency domain than in the time domain. Even though there exists no natural ordering in the frequency domain, we device a recursive algorithm based on the ordering in the time domain.

Here we consider a one dimensional heat diffusion system, which is modeled by a PDE involving the unknown parameters, α_0, the diffusion co-efficient, and κ_0, the thermal conductivity. The given PDE model can be transformed into a standard Linear Time Invariant model (LTI-model) in the frequency domain. The frequency domain representation of the PDE model involves mathematically simple and elegant expressions, thereby making it technically easier to handle the estimation problem in the frequency domain than in the time domain. Moreover, here we assume that the data arrive sequentially in time, hence it would be more appropriate to use a recursive algorithm.

We organise this paper in the following way. In section 2, we introduce the one dimensional heat diffusion system along with the associated PDE model and discuss the parameter estimation problem involved. We transform the given PDE model into a standard LTI model in the frequency domain in section 3. We then proceed to device a recursive algorithm in the frequency domain in section 4. In section 5 we discuss the implementation aspects of the proposed algorithm and present some simulation results. We conclude this paper in section 6, by discussing some finer aspects involved in the algorithm.

2 Problem description

As in [7] and [8], we consider the simple case of a one dimensional heat diffusion across a homogeneous wall. Consider a homogeneous wall of thickness d units.

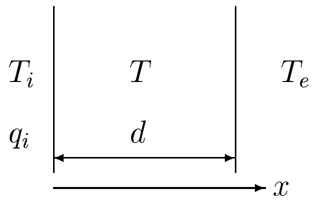

Figure 1: Heat diffusion through a homogeneous wall.

Let x be the co-ordinate across the wall, T_e be the exterior temperature, T_i be the interior temperature, and q_i be the heat supplied from the interior, see Figure 1.

The temperature dynamics across the wall is described by the PDE

$$\frac{\partial T(x,t)}{\partial t} = \alpha_0 \frac{\partial^2 T(x,t)}{\partial x^2}, \quad 0 < x < d \tag{1}$$

with boundary constraints

$$q_i(t) = -\kappa_0 \frac{\partial T(x,t)}{\partial x} |_{x=0}, \tag{2}$$

$$T_e(t) = T(d,t), \tag{3}$$

and initial condition

$$T(x,0) = 0, \quad 0 \leq x \leq d. \tag{4}$$

In (1)-(4) $T(x,t)$ denotes the temperature at the abscissa x and at time t. The diffusion coefficient α_0, (1), and the thermal conductivity κ_0, (2), which characterise the wall, are the unknown parameters.

The estimation problem considered here is the following:

Given: At any given time instant $t = Nh$, we are given the discrete time data $Y^N = \{q_i(0), T_i(0), T_e(0), q_i(h), T_i(h), T_e(h) \ldots\ldots q_i((N-1)h), T_i((N-1)h), T_e((N-1)h)\}$ for a known sampling interval of h units.

Estimate: The parameters α_0 and κ_0.

2.1 Assumptions and Notations

In this subsection, we fix the notations, which will be used throughout this paper, and illustrate the assumptions made on the data.

Let

$$u(t) \triangleq [q_i(t)\ , T_e(t)]^T, \tag{5}$$

and

$$y(t) \triangleq T_i(t). \tag{6}$$

Let

$$y^d(n) \triangleq y(nh), \quad n = 0, 1, 2, \ldots \tag{7}$$

and

$$y^c(t) \triangleq y(t)p(t), \tag{8}$$

where

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nh) \tag{9}$$

and $\delta(t)$ is the Dirac delta function. Normally $p(t)$ is referred as "train of impulses with period h". Note that $y^d(n)$ is a discrete time signal, and $y^c(t)$ is a continuous time signal, $y^c(t)$ is often referred as the ideally sampled version of $y(t)$.

Let

$$Y(i\Omega) \triangleq \int_{-\infty}^{\infty} y(t)e^{-i\Omega t}dt \tag{10}$$

be the Fourier transform of the $y(t)$. Note that by assumption

$$y(t) = 0 \;\; \forall\, t < 0, \tag{11}$$

hence

$$Y(i\Omega) = \int_{0}^{\infty} y(t)e^{-i\Omega t}dt. \tag{12}$$

Let

$$\begin{aligned} Y^D(e^{i\omega}) &\triangleq \sum_{n=0}^{\infty} y^d(n)e^{-i\omega n} \\ &= \sum_{n=0}^{\infty} y(nh)e^{-i\omega n} \end{aligned} \tag{13}$$

be the discrete time Fourier transform (DTFT) of the samples $\{y(nh)\}_{n=0}^{\infty}$. Similarly, let

$$Y^C(i\Omega) \triangleq \int_{-\infty}^{\infty} y^c(t)e^{-i\Omega t}dt \tag{14}$$

be the Fourier transform of the ideally sampled signal $y^c(t)$. Note that

$$\begin{aligned}
Y^C(i\Omega) &= \int_{-\infty}^{\infty} y^c(t)e^{-i\Omega t}dt \\
&= \int_{-\infty}^{\infty} y(t)\sum_{n=-\infty}^{\infty}\delta(t-nh)e^{-i\Omega t}dt \\
&= \sum_{n=-\infty}^{\infty}\int_{0}^{\infty} y(t)\delta(t-nh)e^{-i\Omega t}dt \\
&= \sum_{n=0}^{\infty} y(nh)e^{-i\Omega nh}, \qquad (15)
\end{aligned}$$

which implies that, if one scales the frequency axis Ω by h, i.e, let

$$\Omega h = \omega, \qquad (16)$$

then

$$Y^C(i\frac{\omega}{h}) = Y^D(e^{i\omega}). \qquad (17)$$

Therefore $Y^C(i\Omega)$, where $\Omega = \frac{\omega}{h}$, can be computed using the standard discrete time Fourier transform. One also can see that $Y^C(i\Omega)$, (15), is periodic with period $\frac{2\pi}{h}$.

Since $p(t)$, (9), is periodic with period h, an alternative form for $Y^C(i\Omega)$ can be derived by expanding $p(t)$ as a Fourier series. That is, note that

$$p(t) = \sum_{n=-\infty}^{\infty} c_n e^{i\frac{2\pi n}{h}t}, \qquad (18)$$

where

$$\begin{aligned}
c_n &= \frac{1}{h}\int_{-\frac{h}{2}}^{\frac{h}{2}} p(t)e^{i\frac{2\pi n}{h}t}dt \\
&= \frac{1}{h}. \qquad (19)
\end{aligned}$$

Hence, from (8),

$$y^c(t) = \left(\frac{1}{h}\sum_{n=-\infty}^{\infty} e^{i\frac{2\pi n}{h}t}\right)y(t). \qquad (20)$$

Since an exponential weighting in the time domain results in a shift in the frequency domain, see [4] or [6], from (20), we have

$$Y^C(i\Omega) = \frac{1}{h}\sum_{n=-\infty}^{\infty} Y(i\Omega + i\frac{2\pi n}{h}). \qquad (21)$$

From (21) it is clear that the effect of sampling, (8), on the Fourier transform $Y^C(i\Omega)$, is to add the versions of the spectrum $Y(i\Omega)$, of $y(t)$, shifted by multiples of the sampling frequency $\frac{2\pi}{h}$.

In a similar vein, one can also define $u^d(n)$, $u^c(n)$, $U(i\Omega)$, $U^D(e^{i\omega})$ and $U^C(i\Omega)$, as in (7), (8) (12), (13) and (14) respectively, and show that

$$U^C(i\frac{\omega}{h}) \quad = \quad U^D(e^{i\omega}), \tag{22}$$

and

$$U^C(i\Omega) = \frac{1}{h}\sum_{n=-\infty}^{\infty} U(i\Omega + i\frac{2\pi n}{h}). \tag{23}$$

Now we make the following crucial assumption, called the bandlimitedness assumption, normally made in most signal processing texts. Let $Y(i\Omega)$ and $U(i\Omega)$ be such that

$$Y(i\Omega) \quad = \quad 0 \;\; \forall \;\; |\,\Omega\,| > \frac{\pi}{h} \tag{24}$$

and

$$U(i\Omega) \quad = \quad 0 \;\; \forall \;\; |\,\Omega\,| > \frac{\pi}{h}. \tag{25}$$

In words $Y(i\Omega)$ and $U(i\Omega)$ are band-limited with bandwidth less than $\frac{\pi}{h}$. Note that, since $Y(i\Omega)$ is the Fourier transform of the system output $y(t)$ corresponding to the input $u(t)$, the bandwidth condition on $Y(i\Omega)$, (24), holds automatically if the the bandwidth condition on $U(i\Omega)$, (25), holds.

Using the assumptions (24) and (25), in (21) and (23) we have

$$Y(i\Omega) \quad = \quad hY^C(i\Omega) \;\; \forall \;\; |\,\Omega\,| \leq \frac{\pi}{h} \tag{26}$$

and

$$U(i\Omega) \quad = \quad hU^C(i\Omega) \;\; \forall \;\; |\,\Omega\,| \leq \frac{\pi}{h}. \tag{27}$$

Further, scaling the frequency axis by h, $\Omega h = \omega$, as in (16), and using (17) and (22), we get

$$Y(i\frac{\omega}{h}) \quad = \quad hY^D(e^{i\omega}) \;\; \forall \;\; |\,\omega\,| \leq \pi \tag{28}$$

and

$$U(i\frac{\omega}{h}) \quad = \quad hU^D(e^{i\omega}) \;\; \forall \;\; |\,\omega\,| \leq \pi. \tag{29}$$

Thus, under the bandlimited assumptions (24) and (25), one can compute the continuous time Fourier transforms $Y(i\frac{\omega}{h})$ and $U(i\frac{\omega}{h})$ using the discrete time data $\{y^d(n)\}_{n=0}^{\infty}$ and $\{u^d(n)\}_{n=0}^{\infty}$ respectively.

For a more detailed discussion on ideal sampling (normally referred as Shannon's sampling theorem), we refer to the pages $80 - 84$ in [6].

It must be mentioned that in most cases the bandlimited condition, (25), is never strictly adhered by the input signals, but for many commonly used signals, the bandlimited condition holds approximately. In such cases the relations (28) and (29) hold only approximately.

Here, we make another assumption on the existence of the DTFTs $Y^D(e^{i\omega})$ and $U^D(e^{i\omega})$. Normally for a DTFT, say $Y^D(e^{i\omega})$, to exist one would have to assume one of the following sufficiency conditions on the discrete sequence, $\{y^d(n)\}_{n=0}^{\infty}$:

1. The absolute summability of the sequence $\{y^d(n)\}_{n=0}^{\infty}$,

$$\sum_{n=0}^{\infty} \mid y^d(n) \mid < \infty, \tag{30}$$

   which implies that the finite sum

$$Y_N^D(e^{i\omega}) \triangleq \sum_{n=0}^{N} y^d(n) e^{i\omega n} \tag{31}$$

   would converge to $Y^D(e^{i\omega})$ uniformly as $N \to \infty$.

2. The square summability of the sequence $\{y^d(n)\}_{n=0}^{\infty}$,

$$\sum_{n=0}^{\infty} \mid y^d(n) \mid^2 < \infty, \tag{32}$$

   which implies that the finite sum (30), would converge in the mean square sense to $Y^D(e^{i\omega})$, i.e,

$$\lim_{N\to\infty} \int_0^{2\pi} \mid Y_N^D(e^{i\omega}) - Y^D(e^{i\omega}) \mid^2 d\omega = 0. \tag{33}$$

Note that, the first condition is stronger than the second, and both the conditions hold for finite duration signals. For a more thorough discussion on the existence of DTFT's we refer to pages $47 - 49$ in [4].

Similarly we assume that, the sequence $\{u^d(n)\}_{n=0}^{\infty}$ is absolutely square summable, (32). Therefore, defining

$$U_N^D(e^{i\omega}) \triangleq \sum_{n=0}^{N} u^d(n) e^{i\omega n}, \tag{34}$$

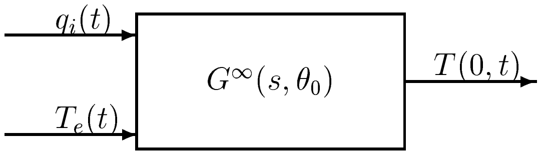

Figure 2: PDE viewed as a dynamic system

we have

$$\lim_{N\to\infty}\int_0^{2\pi} \mid U_N^D(e^{i\omega}) - U^D(e^{i\omega}) \mid^2 d\omega = 0. \tag{35}$$

3 System dynamics

In this section, we briefly present a characterisation of the temperature dynamics of the infinite order PDE model in the frequency domain. For a more detailed discussion on the results presented here refer to either [7] or [8].

The given diffusion system can be interpreted as an LTI system with inputs

$$u(t) \;=\; [q_i(t)\;, T_e(t)], \tag{36}$$

and the output

$$\begin{aligned} y(t) &= T(0,t), \\ &= T_i(t) \end{aligned} \tag{37}$$

see also the Figure 2. In [7] and [8], we have shown that the given diffusion system can be re-written in the Laplace domain as

$$Y(s) \;=\; G^\infty(s,\theta_0)U(s), \tag{38}$$

where $Y(s)$ and $U(s)$ are the Laplace transforms of the output $y(t)$ and the input $u(t)$ respectively, and the transfer-function $G^\infty(s,\theta_0)$ is

$$\begin{aligned} G^\infty(s,\theta_0) &= \left[\frac{\tanh(d\sqrt{\frac{s}{\alpha_0}})}{\kappa_0\sqrt{\frac{s}{\alpha_0}}}\;,\;\frac{1}{\cosh(d\sqrt{\frac{s}{\alpha_0}})}\right] \\ &\triangleq [G_1^\infty(s,\theta_0), G_2^\infty(s,\theta_0)] \end{aligned} \tag{39}$$

where $\theta_0 = (\alpha_0, \kappa_0)$.

Note that both the system transfer-functions, $G_1^\infty(s,\theta_0)$ and $G_2^\infty(s,\theta_0)$ are transcendental functions having the same set of poles, which are real and negative. Hence the given system is stable. In [7] and [8] we have plotted both the transfer functions $G_1^\infty(i\omega,\theta_0)$ and $G_2^\infty(i\omega,\theta_0)$ for a wide range of frequencies. The plots therein suggest that both $G_1^\infty(s,\theta_0)$ and $G_2^\infty(s,\theta_0)$ have lowpass characteristics.

4 Recursive algorithm in the frequency domain (The FD-R algorithm)

4.1 Simple Estimator

From the previous section it is fairly clear that, an estimate of the parameter vector θ_0 in the least square sense is

$$\hat{\theta} \;=\; \arg\min_{\theta} \int_0^{\infty} \mid Y(i\Omega) - \hat{Y}(i\Omega) \mid^2 d\Omega \tag{40}$$

where $Y(i\Omega)$, (12), is the Fourier transform of the system output,

$$\hat{Y}(i\Omega) = G^{\infty}(i\Omega, \theta)U(i\Omega), \tag{41}$$

is the Fourier transform of the model output, and $U(i\Omega)$ is the Fourier transform of the system input. Note that, the limits of the integral in (40) are in the positive frequency axis. Here, we have neglected the negative frequency axis since the given signals are real valued. It can be checked that the transfer-function $G^{\infty}(i\Omega, \theta)$ is a bijection with respect to the parameter θ, i.e,

$$G^{\infty}(i\Omega, \theta_1) \;=\; G^{\infty}(i\Omega, \theta_2) \; \forall \; \Omega \tag{42}$$

iff $\theta_1 = \theta_2$, hence, if $\hat{\theta}$ can be computed, we would have $\hat{\theta} = \theta_0$.

Note that one cannot compute $\hat{\theta}$, (40), without discretising the integral, hence a more reasonable estimator of θ_0 at a given instant $t = Nh$ is

$$\hat{\theta}(N) \;=\; \arg\min_{\theta} \frac{1}{m} \sum_{k=0}^{m-1} \mid Y(i\Omega_k) - \hat{Y}(i\Omega_k) \mid^2, \tag{43}$$

where $\{\Omega_k\}_{k=0}^{m-1}$ are a set of chosen frequency points, where $Y(i\Omega_k)$ and $\hat{Y}(i\Omega_k)$ are either known or can be computed at the given instant $t = Nh$.

We now go on device a scheme to compute (43).

Note that at any given instant $t = Nh$, the data at our disposal $\{y^d(n)\}_{n=0}^{N-1}$ and $\{u^d(n)\}_{n=0}^{N-1}$ are in the time domain. Therefore, to compute (43) one would have to compute the Fourier transforms $Y(i\Omega)$ and $\hat{Y}(i\Omega)$ through their DTFTs $Y^D(e^{i\omega})$ and $\hat{Y}^D(e^{i\omega})$, see (28) and (29). It is also well known that DFTs can be smartly computed using the Fast Fourier Transform (FFT), see [4].

Let

$$\Omega_k = \frac{\omega_k}{h} \quad k = 0, 1, \ldots, N-1, \tag{44}$$

where

$$\omega_k = \begin{cases} \frac{2\pi k}{N} & k = 0, 1, \ldots, \frac{N}{2} \\ \frac{2\pi(k-N)}{N} & k = \frac{N}{2}+1, \ldots, N-1 \end{cases} \tag{45}$$

Note that

$$| \Omega_k | \leq \frac{\pi}{h} \quad k = 0, 1, \ldots, N-1. \tag{46}$$

Assuming that $y(t)$ and $u(t)$ satisfy the conditions (24) and (25) respectively, and using (26) and (27), we have

$$Y(i\Omega_k) \;=\; hY^C(i\Omega_k) \quad k = 0, 1, \ldots, N-1 \tag{47}$$

and

$$U(i\Omega_k) \;=\; hU^C(i\Omega_k) \quad k = 0, 1, \ldots, N-1. \tag{48}$$

Further, using (28) and (29), we get

$$Y(i\Omega_k) \;=\; hY^D(e^{i\omega_k}) \quad k = 0, 1, \ldots, N-1 \tag{49}$$

and

$$U(i\Omega_k) \;=\; hU^D(e^{i\omega_k}) \quad k = 0, 1, \ldots, N-1. \tag{50}$$

Thus the Fourier transforms $Y(i\Omega_k)$ and $U(i\Omega_k)$, can be computed using their respective DFTs $Y^D(e^{i\omega_k})$ and $U^D(e^{i\omega_k})$, where $k = 0, 1, \ldots, N-1$.

Since $y(t)$ and $u(t)$ are real valued, it is enough if one considers the positive frequencies

$$\omega_k = \frac{2\pi}{N}k \qquad k = 0, 1, \ldots, N/2. \tag{51}$$

As the given data at any instant $t = Nh$ is of length N, one cannot compute the DFTs $Y^D(e^{i\omega_k})$ and $U^D(e^{i\omega_k})$, $k = 0, 1, \ldots, N-1$, but only their approximations

$$\begin{aligned} Y_N^D(e^{i\omega_k}) &= \sum_{n=0}^{N-1} y^d(n) e^{-i\omega_k n} \\ &\triangleq \mathrm{DFT}[\{y^d(n)\}_{n=0}^{N-1}] \mid_{\omega_k} \end{aligned} \tag{52}$$

and

$$\begin{aligned} U_N^D(e^{i\omega_k}) &= \sum_{n=0}^{N-1} u^d(n) e^{-i\omega_k n} \\ &\triangleq \mathrm{DFT}[\{u^d(n)\}_{n=0}^{N-1}] \mid_{\omega_k}, \end{aligned} \tag{53}$$

where ω_k's are as defined in (51). Therefore, the parameter estimate (43) will be replaced by the computable estimate

$$\hat{\theta}(N) \;=\; \arg\min_{\theta} \frac{1}{N} \sum_{k=0}^{N-1} | Y_N^D(e^{i\omega_k}) - \hat{Y}_N^D(e^{i\omega_k}) |^2, \tag{54}$$

where

$$\hat{Y}_N^D(e^{i\omega_k}) = G^\infty(i\frac{\omega_k}{h}, \theta) U_N^D(e^{i\omega_k}). \tag{55}$$

It must be mentioned here that, we do not claim $Y_N^D(e^{i\omega}) \to Y^D(e^{i\omega})$ and $U_N^D(e^{i\omega}) \to U^D(e^{i\omega})$, as we have not assumed the absolute summability, (30), of the sequences $\{y^d(n)\}_{n=0}^{\infty}$ and $\{u^d(n)\}_{n=0}^{\infty}$. But, since $\{y^d(n)\}_{n=0}^{\infty}$ and $\{u^d(n)\}_{n=0}^{\infty}$ are square summable, (32), one can show that $\hat{\theta}(N)$, (54), converges to θ_0, as $N \to \infty$.

Note that to compute the least squares estimate $\hat{\theta}(N)$, (54), one has to use a non-linear search, which is computationally expensive. Further, to compute $Y_N^D(e^{i\omega_k})$ and $U_N^D(e^{i\omega_k})$, for each $k = 0, 1, 2, \ldots, N-1$, one has to use the whole data set $\{y^d(n)\}_{n=0}^{N-1}$ and $\{u^d(n)\}_{n=0}^{N-1}$ at each instant $t = Nh$. Therefore a direct computation of the conceptually simple estimator (54), could incur large storage costs. Further, in some cases the system parameter θ_0 could be time varying, even though we do not consider such a case here. In such cases direct usage of the least squares estimate, $\hat{\theta}(N)$, may not be appropriate. Hence it would be better to device a recursive algorithm.

4.2 Frequency Domain Recursive Algorithm (FD-R)

To circumvent the above mentioned problems of computational complexity and storage cost, we device the following recursive algorithm.

Let L be a fixed positive integer integer. Consider the following windowed data $\{y^d(n)\}_{n=N-L}^{N-1}$ and $\{u^d(n)\}_{n=N-L}^{N-1}$ of length L, at the time instant $t = Nh$, where $N \geq L$. Let

$$E_L^N(i\omega_k, \theta) \triangleq Y_L^N(e^{i\omega_k}) - \hat{Y}_L^N(e^{i\omega_k}, \theta), \tag{56}$$

where

$$\begin{aligned}
Y_L^N(e^{i\omega_k}) &= \sum_{m=0}^{L-1} y^d(m+N-L) e^{-i\omega_k m} \\
&= \mathrm{DFT}[\{y^d(n)\}_{n=N-L}^{N-1}] \mid_{\omega_k}, &(57)\\
\hat{Y}_L^N(e^{i\omega_k}, \theta) &= G^\infty(i\frac{\omega_k}{h}, \theta) U_L^N(e^{i\omega_k}), &(58)\\
U_L^N(e^{i\omega_k}) &= \sum_{m=0}^{L-1} u^d(m+N-L) e^{-i\omega_k m} \\
&= \mathrm{DFT}[\{u^d(n)\}_{n=N-L}^{N-1}] \mid_{\omega_k} &(59)\\
\omega_k &= \frac{2\pi k}{L} \quad k = 0, 1, 2, \ldots, \frac{L}{2}, &(60)
\end{aligned}$$

and

$$V_L^N(\theta) \triangleq \sum_{k=0}^{\frac{L}{2}} | E_L^N(i\omega_k, \theta) |^2 . \tag{61}$$

We define the cost function at the instant $t = Nh$ to be

$$W^N(\theta) \triangleq \frac{1}{N-L+1} \sum_{m=L}^{N} V_L^m(\theta). \tag{62}$$

Note that, we have defined the cost function to be $W^N(\theta)$, (62), and not $V_L^N(\theta)$, (61). From (61) and (62) it is apparent that, by using $V_L^N(\theta)$ as the cost function, at any given instant $t = Nh$ one would only use the windowed data $\{y^d(n)\}_{n=N-L}^{N-1}$ and $\{u^d(n)\}_{n=N-L}^{N-1}$, but in the case of $W^N(\theta)$ one would use the whole data set $\{y^d(n)\}_{n=0}^{N-1}$ and $\{u^d(n)\}_{n=0}^{N-1}$. For the sake of clarity in the next subsection, we device a recursive algorithm corresponding to the cost function $V_L^N(\theta)$ and show that by using $W^N(\theta)$ as the cost function one neither increases the computational complexity nor incurs any additional storage costs. Further, in section 5, using a simulation example we demonstrate the superiority of the FD-R algorithm obtained by using $W^N(\theta)$ as the cost function over the recursive algorithm minimising $V_L^N(\theta)$.

We now derive a recursive algorithm for estimating θ_0 with $W^N(\theta)$, (62), as the cost function.

Here, we wish to obtain an estimate $\hat{\theta}(N)$ of θ_0 at each instant $t = Nh$, such that

$$\hat{\theta}(N) = \arg\min_{\theta} W^N(\theta). \tag{63}$$

Note that this estimate is different from the estimate (54).

Most of the derivation here is patterned after the derivation of RPEM presented in page 26 of [3].

Let $\hat{\theta}(N-1)$ be the estimate of θ_0 at the time instant $t = (N-1)h$. As mentioned above, we wish to obtain $\hat{\theta}(N)$ at the instant $t = Nh$, which minimises $W^N(\theta)$, at least approximately.

Note that, by a Taylor expansion of $W^N(\theta)$ around $\hat{\theta}(N-1)$, we have

$$\begin{aligned} W^N(\theta) &= W^N(\hat{\theta}(N-1)) + W^{N\prime}(\hat{\theta}(N-1))(\theta - \hat{\theta}(N-1)) \\ &\quad + \frac{1}{2}(\theta - \hat{\theta}(N-1))^T W^{N\prime\prime}(\hat{\theta}(N-1))(\theta - \hat{\theta}(N-1)) \\ &\quad + o(| \theta - \hat{\theta}(N-1) |^2), \end{aligned} \tag{64}$$

where the prime denotes the differentiation with respect to θ and $o(x)$ denotes a function such that $\frac{o(x)}{x} \to 0$ as $| x | \to 0$. Minimisation of (64) with respect to θ

gives

$$
\begin{aligned}
\hat{\theta}(N) &= \hat{\theta}(N-1) - [W^{N\,''}(\hat{\theta}(N-1))]^{-1}[W^{N\,'}(\hat{\theta}(N-1))] \\
&\quad + o(\mid \hat{\theta}(N) - \hat{\theta}(N-1) \mid). \qquad (65)
\end{aligned}
$$

Note that

$$
W^N(\theta) = \frac{N-L}{N-L+1} W^{N-1}(\theta) + \frac{1}{N-L+1} V_L^N(\theta), \qquad (66)
$$

and hence

$$
W^{N\,'}(\theta) = \frac{N-L}{N-L+1} W^{N-1\,'}(\theta) + \frac{1}{N-L+1} V_L^{N\,'}(\theta). \qquad (67)
$$

To evaluate (65), we introduce the following approximations

1. First we assume that $\hat{\theta}(N-1)$ minimises $W^{N-1}(\theta)$, which implies that

$$
W^{N-1\,'}(\hat{\theta}(N-1)) = 0. \qquad (68)
$$

2. Next we assume that $\hat{\theta}(N)$ lies in a small neighborhood of $\hat{\theta}(N-1)$. This assumption is reasonable for a large N. This assumption implies that

$$
o(\mid \hat{\theta}(N) - \hat{\theta}(N-1) \mid) \approx 0. \qquad (69)
$$

3. Finally we assume L is large enough, so that

$$
\sum_{m=L}^{N} V_L^{m\,''}(\hat{\theta}(N-1)) \approx (N-L+1) V_L^{N\,''}(\hat{\theta}(N-1)). \qquad (70)
$$

Thus using (67)-(70) in (65), we have the following recursive algorithm

$$
\hat{\theta}(N) = \hat{\theta}(N-1) - \frac{1}{N-L+1} [V_L^{N\,''}(\hat{\theta}(N-1))]^{-1} V_L^{N\,'}(\hat{\theta}(N-1)) \qquad (71)
$$

to estimate θ_0 recursively.

4.3 Windowed Frequency Domain Recursive Algorithm (WFD-R)

Here, we describe a recursive algorithm in the frequency domain using $V_L^N(\theta)$, rather than $W^N(\theta)$, as the cost function. We call the resulting algorithm as the "windowed frequency domain recursive algorithm (WFD-R)".

We wish to obtain an estimate $\tilde{\theta}(N)$ of θ_0 at each instant $t = Nh$, such that

$$
\tilde{\theta}(N) = \arg\min_{\theta} V_L^N(\theta). \qquad (72)
$$

The derivation of the WFD-R algorithm is indeed similar to the derivation of the FD-R algorithm presented in the previous subsection.

Let $\tilde{\theta}(N-1)$ be the estimate of θ_0 at the time instant $t=(N-1)h$. Substituting $W^N(\theta)$ and its derivatives by $V_L^N(\theta)$ and its derivatives in (71) leads directly to

$$\begin{aligned}\tilde{\theta}(N) &= \tilde{\theta}(N-1)-[V_L^{N''}(\tilde{\theta}(N-1))]^{-1}[V_L^{N'}(\tilde{\theta}(N-1))] \\ &\quad +o(\mid \tilde{\theta}(N)-\tilde{\theta}(N-1)\mid). \end{aligned} \tag{73}$$

As in the previous subsection, we assume that $\tilde{\theta}(N)$ lies in a small neighborhood of $\tilde{\theta}(N-1)$. This assumption is reasonable for a large N. This assumption implies that

$$o(\mid \tilde{\theta}(N)-\tilde{\theta}(N-1)\mid) \approx 0. \tag{74}$$

Hence using (74) in (73), we get the following recursive algorithm

$$\tilde{\theta}(N) = \tilde{\theta}(N-1)-[V_L^{N''}(\tilde{\theta}(N-1))]^{-1}[V_L^{N'}(\tilde{\theta}(N-1))]. \tag{75}$$

Note that the FD-R algorithm, (71), and the WFD-R algorithm (75), differ only in their respective step sizes. In the case of the FD-R algorithm the step size is $\frac{1}{N-L+1}$ and in the case of the WFD-R algorithm the step size is 1. Later on, it can be seen that this small but crucial difference makes the FD-R algorithm exhibit better convergence properties than the WFD-R algorithm.

4.4 Recursive computation of the DFT's

Note that at any given instant one has to store the last L values of the data to compute (57) and (59), thereby incurring a storage cost. The computational complexity involved in computing the DFT's can be considerably reduced by exploiting the recursion inherently present in them.

Note that by definition

$$Y_L^N(e^{i\omega_k}) = \sum_{n=0}^{L-1} y^d(N-L+n)e^{-i\omega_k n}, \tag{76}$$

which implies that

$$Y_L^N(e^{i\omega_k})-y^d(N-L)=\sum_{n=1}^{L-1} y^d(N-L+n)e^{-i\omega_k n}. \tag{77}$$

Multiplying both the sides of (77) by $e^{i\omega_k}$, we have

$$[Y_L^N(e^{i\omega_k})-y^d(N-L)]e^{i\omega_k}=\sum_{n=1}^{L-1} y^d(N-L+n)e^{-i(n-1)\omega_k}. \tag{78}$$

Finally adding $y^d(N)e^{-i(L-1)\omega_k}$ to both sides of (78), we have

$$\begin{aligned}
[Y_L^N(e^{i\omega_k}) - y^d(N-L)]e^{i\omega_k} + y^d(N)e^{-i(L-1)\omega_k} &= \sum_{n=1}^{L-1} y^d(N-L+n)e^{-i(n-1)\omega_k} \\
&+ y^d(N)e^{-i(L-1)\omega_k} \\
&= \sum_{n=0}^{L-1} y^d(N+1-L+n)e^{-i\omega_k n} \\
&= Y_L^{N+1}(e^{i\omega_k}). \qquad (79)
\end{aligned}$$

Note that, from(79), one can compute $Y_L^{N+1}(e^{i\omega_k})$ recursively, for each ω_k, without explicitly using the data at the instants $t = (N-L+1)h, \ldots, (N-1)h$, and moreover, computing $Y_L^{N+1}(e^{i\omega_k})$, for each ω_k in this fashion involves only two multiplications and two additions.

Similarly, one can show that

$$U_L^{N+1}(e^{i\omega_k}) = [U_L^N(e^{i\omega_k}) - u^d(N-L)]e^{i\omega_k} + u^d(N)e^{-i(L-1)\omega_k}. \qquad (80)$$

5 Implementation and Simulations

In this section, we discuss some implementation aspects involved in simulating the FD-R and the WFD-R algorithms. We also a consider a simulation example and present the results.

5.1 Implementation

Since the FD-R and the WFD-R algorithms differ only in their step sizes, it is sufficient to discuss the implementation issues involved in only one of them. Therefore, here, we discuss the implementation of the FD-R algorithm alone.

Note that to use the FD-R algorithm, (71), one has to compute the first and second derivatives of $V_L^N(\theta)$, with respect to θ, at each instant $t = Nh$.

From, (61), we know that

$$V_L^N(\theta) = \sum_{k=0}^{\frac{L}{2}} E_L^N(i\omega_k, \theta)E_L^{N\,*}(i\omega_k, \theta), \qquad (81)$$

where * denotes the conjugate. Hence,

$$V_L^{N\,\prime}(\theta) = \sum_{k=0}^{\frac{L}{2}} \{E_L^{N\,\prime}(i\omega_k, \theta)E_L^{N\,*}(i\omega_k, \theta) + E_L^N(i\omega_k, \theta)E_L^{N\,\prime *}(i\omega_k, \theta)\}. \qquad (82)$$

Moving on to the second derivative ${V_L^N}''(\theta)$, from (82) we can see that

$$\begin{aligned} {V_L^N}''(\theta) &= \sum_{k=0}^{\frac{L}{2}} \{ {E_L^N}''(i\omega_k,\theta) {E_L^N}^*(i\omega_k,\theta) + {E_L^N}'(i\omega_k,\theta) {E_L^N}'^*(i\omega_k,\theta) \} \\ &+ \sum_{k=0}^{\frac{L}{2}} \{ E_L^N(i\omega_k,\theta) {E_L^N}''^*(i\omega_k,\theta) + {E_L^N}'(i\omega_k,\theta) {E_L^N}'^*(i\omega_k,\theta) \}. \quad (83) \end{aligned}$$

Note that, the matrix ${V_L^N}''(\theta)$ has to be positive definite for (71) to converge. Therefore, it would be better to force the positive definiteness on ${V_L^N}''(\theta)$, then to assume it. Hence we approximate

$$\begin{aligned} {V_L^N}''(\theta) &\approx \sum_{k=0}^{\frac{L}{2}} \{ {E_L^N}'(i\omega_k,\theta) {E_L^N}'^*(i\omega_k,\theta) + {E_L^N}'(i\omega_k,\theta) {E_L^N}'^*(i\omega_k,\theta) \}. \\ &= \sum_{k=0}^{\frac{L}{2}} {E_L^N}'(i\omega_k,\theta) {E_L^N}'^*(i\omega_k,\theta). \quad (84) \end{aligned}$$

From (56) and (58), we have

$$ {E_L^N}'(i\omega_k,\theta) = -\frac{dG^\infty}{d\theta}(i\frac{\omega_k}{h},\theta) U_L^N(e^{i\omega_k}) \quad (85)$$

and

$$ {E_L^N}'^*(i\omega_k,\theta) = [-\frac{dG^\infty}{d\theta}(i\frac{\omega_k}{h},\theta) U_L^N(e^{i\omega_k})]^*. \quad (86)$$

Thus form (82), (84), (85) and (86), we can see that to compute ${V_L^N}'(\theta)$ and ${V_L^N}''(\theta)$, one has to analytically determine $\frac{dG^\infty}{d\theta}(i\frac{\omega}{h},\theta)$, and evaluate it each ω_k. It can be checked that, by differentiating transfer-function (39),

$$\begin{aligned} \frac{dG_1^\infty}{d\alpha}(s,\theta) &= \frac{1}{2\kappa}\left(\frac{\tanh(d\sqrt{\frac{s}{\alpha}})}{\sqrt{s}\alpha} - \frac{d}{\alpha}\frac{1}{\cosh^2(d\sqrt{\frac{s}{\alpha}})}\right) && (87) \\ \frac{dG_1^\infty}{d\kappa}(s,\theta) &= -\frac{G_1^\infty(s,\theta)}{\kappa} && (88) \\ \frac{dG_2^\infty}{d\alpha}(s,\theta) &= \frac{d}{2}\frac{\sinh(d\sqrt{\frac{s}{\alpha}})}{\cosh^2(d\sqrt{\frac{s}{\alpha}})}\sqrt{\frac{s}{\alpha^3}} && (89) \\ \frac{dG_2^\infty}{d\kappa}(s,\theta) &= 0. && (90) \end{aligned}$$

Summarising the FD-R algorithm, we have

$$ \hat{\theta}(N) = \hat{\theta}(N-1) - \frac{1}{N-L+1}[{V_L^N}''(\hat{\theta}(N-1))]^{-1} {V_L^N}'(\hat{\theta}(N-1)), \quad (91)$$

where

$$\left\{\begin{array}{rcl}
V_L^{N\,\prime}(\theta) & = & \sum_{k=0}^{\frac{L}{2}}\{E_L^{N\,\prime}(i\omega_k,\theta)E_L^{N\,*}(i\omega_k,\theta)+E_L^N(i\omega_k,\theta)E_L^{N\,\prime *}(i\omega_k,\theta)\} \\
V_L^{N\,\prime\prime}(\theta) & = & \sum_{k=0}^{\frac{L}{2}} E_L^{N\,\prime}(i\omega_k,\theta)E_L^{N\,\prime *}(i\omega_k,\theta) \\
\omega_k & = & \frac{2\pi k}{L} \quad k=0,1,2,\ldots,\frac{L}{2} \\
E_L^N(i\omega_k,\theta) & = & Y_L^N(e^{i\omega_k})-\hat{Y}_L^N(e^{i\omega_k},\theta) \\
\hat{Y}_L^N(e^{i\omega_k},\theta) & = & G^\infty(i\frac{\omega_k}{h},\theta)U_L^N(e^{i\omega_k}) \\
Y_L^N(e^{i\omega_k}) & = & [Y_L^{N-1}(e^{i\omega_k})-y^d(N-L-1)]e^{i\omega_k}+y^d(N-1)e^{-i(L-1)\omega_k} \\
U_L^N(e^{i\omega_k}) & = & [U_L^{N-1}(e^{i\omega_k})-u^d(N-L-1)]e^{i\omega_k}+u^d(N-1)e^{-i(L-1)\omega_k} \\
E_L^{N\,\prime}(i\omega_k,\theta) & = & -\frac{dG^\infty}{d\theta}(i\frac{\omega_k}{h},\theta)U_L^N(e^{i\omega_k}) \\
\frac{dG_1^\infty}{d\alpha}(i\frac{\omega_k}{h},\theta) & = & \frac{1}{2\kappa}\left(\frac{\tanh(d\sqrt{\frac{i\omega_k}{\alpha h}})}{\sqrt{i\frac{\omega_k\alpha}{h}}}-\frac{d}{\alpha}\frac{1}{\cosh^2(d\sqrt{\frac{i\omega_k}{\alpha h}})}\right) \\
\frac{dG_1^\infty}{d\kappa}(i\frac{\omega_k}{h},\theta) & = & -\frac{G_1^\infty(i\frac{\omega_k}{h},\theta)}{\kappa} \\
\frac{dG_2^\infty}{d\alpha}(i\frac{\omega_k}{h},\theta) & = & \frac{d}{2}\frac{\sinh(d\sqrt{\frac{i\omega_k}{\alpha h}})}{\cosh^2(d\sqrt{\frac{i\omega_k}{\alpha h}})}\sqrt{\frac{i\omega_k}{\alpha^3 h}} \\
\frac{dG_2^\infty}{d\kappa}(i\frac{\omega_k}{h},\theta) & = & 0.
\end{array}\right. \tag{92}$$

5.2 Simulation example

In this subsection we use the FD-R algorithm, (91) and (92), and the WFD-R algorithm, (75) and (92), to estimate the parameter θ_0 from a simulated data set.

Before we go to the simulation example, it may be worth mentioning that, we do not allow the components of the parameter estimates $\hat{\theta}(N)$, (71), and $\tilde{\theta}(N)$, (75), obtained by using the FD-R the WFD-R algorithms respectively, to get negative, as they would render the system unstable. If at any particular instant, the components of parameter estimates $\hat{\theta}(N)$ or $\tilde{\theta}(N)$ do get negative, we then project them back into positive quadrant by forcing them to equal a value $\delta = 0.1$.

We generate the system inputs, $\{q_i(nh)\}_{n=0}^{N-1}$ and $\{T_e(nh)\}_{n=0}^{N-1}$, and the system output $\{T_i(nh)\}_{n=0}^{N-1}$, for the parameter estimation in the following fashion.

First, we generate two independent white noise sequences $\{v(n)\}_{n=0}^{N-1}$ and $\{w(n)\}_{n=0}^{N-1}$ of variance one, and pass them through a lowpass filter

$$H(q^{-1}) \;=\; \frac{1}{1-1.88q^{-1}+0.9732q^{-2}}, \tag{93}$$

which is of finite bandwidth and has a resonance peak. Let $\{v_1(n)\}_{n=0}^{N-1}$ and $\{w_1(n)\}_{n=0}^{N-1}$ be the filter outputs corresponding to the white noise sequences $\{v(n)\}_{n=0}^{N-1}$ and $\{w(n)\}_{n=0}^{N-1}$ respectively, i.e.

$$v_1(n) - 1.88v_1(n-1) + 0.9732v_1(n-2) \;=\; v(n) \tag{94}$$

and

$$w_1(n) - 1.88w_1(n-1) + 0.9732w_1(n-2) \;=\; w(n), \tag{95}$$

where $v_1(0) = v(0)$, $v_1(1) = v(1)$, $w_1(0) = w(0)$, $w_1(1) = w(1)$ and $n = 2, 3, \ldots, N-1$.

To the filter output $\{v_1(n)\}_{n=0}^{N-1}$, we further add a white noise sequence $\{v_2(n)\}_{n=0}^{N-1}$ of variance 0.8 and scale the net signal by a value 0.2. The scaled total signal

$$u_1(n) = 0.2(v_1(n)+v_2(n)), \quad n = 0,1,2,\ldots,N-1, \tag{96}$$

is then chosen as the system input $\{q_i(nh)\}_{n=0}^{N-1}$.

To the filter output $\{w_1(n)\}_{n=0}^{N-1}$, we add a white noise sequence $\{w_2(n)\}_{n=0}^{N-1}$ of variance 0.8, and the net signal

$$u_2(n) = w_1(n) + w_2(n), \quad n = 0,1,2,\ldots,N-1, \tag{97}$$

is chosen as system input $\{T_e(nh)\}_{n=0}^{N-1}$.

Note that both input signals $\{q_i(nh)\}_{n=0}^{N-1}$ and $\{T_e(nh)\}_{n=0}^{N-1}$ are reasonably rich in the frequencies within the band-width of the filter (93) and also have some high frequency components.

Let the system parameters be $\alpha_0 = 2$ m/s^2 and $\kappa_0 = 1$ J/msK. Assume that the sampling interval h is expressed in units of seconds.

The output $\{y^d(n) = T_i(nh)\}_{n=0}^{N-1}$ is generated by first multiplying the DFT of the input,

$$U^N(e^{i\omega_k}) = \mathrm{DFT}[\{u(n), \}_{n=0}^{N-1},]\,|_{\omega_k} \tag{98}$$

with

$$G_1^{\infty}(i\frac{\omega_k}{h}, \theta_0), \tag{99}$$

where

$$\omega_k = \frac{2\pi k}{N} \;\text{ rads} \quad k = 0,1,2,\ldots,\frac{N}{2}, \tag{100}$$

at each ω_k, and taking the IDFT of the product, i.e.,

$$y^d(n) = \mathrm{IDFT}[\{G_1^{\infty}(i\frac{\omega_k}{h}, \theta_0)U^N(e^{i\omega_k})\}_{k=-\frac{N}{2}+1}^{\frac{N}{2}},] \,|_n \, . \tag{101}$$

From the generated input and output data pairs, $\{y^d(n)\}_{n=0}^{N-1}$ and $\{u^d(n)\}_{n=0}^{N-1}$, we then estimate θ_0 using the FD-R and the WFD-R algorithms.

Since the chosen system inputs are random in nature, we estimate the mean

$$\hat{\theta}_m^L(N) \triangleq \frac{1}{50}\sum_{r=1}^{50} \hat{\theta}_r^L(N), \tag{102}$$

and the variance

$$\sigma_L^2(N) \triangleq \frac{1}{50}\sum_{r=1}^{50}(\hat{\theta}_r^L(N) - \hat{\theta}_m^L(N))(\hat{\theta}_r^L(N) - \hat{\theta}_m^L(N))^T, \tag{103}$$

of the parameter estimates over 50 different input realisations. Here $\hat{\theta}_r^L(N)$ denotes the parameter estimate at the instant $t = N$, obtained by using either the FD-R or the WFD-R algorithm with a window of length L, for the r^{th} input realisation.

In Figures 3, 4 and 5, we have plotted the mean $\hat{\theta}_m^L(N)$, (102), and the variance $\sigma_L^2(N)$, (103), obtained by using the the FD-R algorithm with the window lengths $L = 100, 200$ and 300. Note that, from Figure 3, as L increases the averaged estimates, $\hat{\theta}_m^L(N)$, get closer to θ_0 asymptotically. And similarly, from Figures 4 and 5 we can see that the variance $\sigma_L^2(N)$ gets closer to zero, asymptotically, for an increasing L.

In Figures 6, 7 and 8, we have plotted the averaged parameter estimates, (102), obtained by using the WFD-R algorithm, for window lengths $L = 500, 600$ and 700. Note that, the averaged parameter estimates oscillate violently around θ_0, and as L increases the oscillations are less frequent. Here, we have deliberately avoided plots with window lengths $L = 100 - 400$, as the oscillations in the parameter estimates are then too violent. And also note that here we have avoided the plots of the variance $\sigma_L^2(N)$, as they are too large to be meaningful.

A common strategy to remove the large variations in a given data set, is to filter the data using a lowpass filter. Here, in order to remove the large spikes in averaged WFD-R estimates $\hat{\theta}_m^L(N)$, see Figures 6, 7 and 8, we pass $\hat{\theta}_m^L(N)$ through the lowpass filter

$$H(q^{-1}; \beta) = \frac{\beta}{1 - (1 - \beta)q^{-1}}, \tag{104}$$

where q^{-1} denotes the unit time delay operator and β is a real number such that $0 < \beta < 1$, and obtain the estimate

$$\hat{\hat{\theta}}^L(N) = H(q^{-1}; \beta)\hat{\theta}_m^L(N). \tag{105}$$

Note that, from (105),

$$\hat{\hat{\theta}}^L(N) = (1-\beta)\hat{\hat{\theta}}^L(N-1) + \beta\hat{\theta}_m^L(N). \tag{106}$$

Hence note that for $\beta = 1$

$$\hat{\hat{\theta}}^L(N) = \hat{\theta}_m^L(N), \tag{107}$$

for $\beta \approx 1$

$$\hat{\hat{\theta}}^L(N) \approx \beta\hat{\theta}_m^L(N), \tag{108}$$

and for $\beta << 1$

$$\hat{\hat{\theta}}^L(N) \approx \hat{\hat{\theta}}^L(N-1) + \beta\hat{\theta}_m^L(N). \tag{109}$$

Thus for a β close 1, $H(q^{-1};\beta)$ is essentially an allpass memoryless filter, and as $\beta \to 0$, $H(q^{-1};\beta)$ gets more and more lowpass, having a large memory.

In Figures 9, 10, 11, 12, 13 and 14 we have plotted the filtered parameter estimates $\hat{\hat{\theta}}^L(N) \triangleq (\hat{\hat{\alpha}}^L(N), \hat{\hat{\kappa}}^L(N))$, along with the averaged WFD-R estimates $\hat{\theta}_m^L(N)$, for $L = 500, 600$ and 700 and $\beta = 0.1, 0.01$ and 0.001. Note that in all the above mentioned figures the case $\beta = 1$ refers to the averaged parameter estimate $\hat{\theta}_m^L(N)$.

From the Figures 9, 10, 11, 12, 13 and 14 one can see that, $\hat{\theta}_m^L(N)$ and $\hat{\hat{\theta}}^L(N)$ do not differ much for a large β, and as $\beta \to 0$ the oscillations of the averaged WFD-R estimate $\hat{\theta}_L^m(N)$ are filtered out and one gets a smoother convergence. This is not surprising, as we have observed earlier that $H(q^{-1};\beta)$ is close to an all pass filter for a large β. And as $\beta \to 0$, $H(q^{-1};\beta)$ gets more and more lowpass.

Note that, not surprisingly, a reduction in the variance of the parameter estimates, by filtering, leads to a reduction in the rate of convergence of the parameter estimates, even though the filtered parameter estimates $\hat{\hat{\theta}}^L(N)$ do get closer to θ_0, asymptotically, for an increasing L.

5.3 Time domain recursive algorithms

For the sake of completion, here, we briefly discuss about a recursive algorithm in the time domain, proposed in [2], for estimating the parameters α and κ. We also present some numerical results obtained by using it.

In [2], we first approximate the infinite order dynamics of PDE model, (1)-(4), by a finite order linear system of the form

$$\begin{aligned} \dot{x} &= \alpha Ax + \alpha Bu \\ y &= Cx + Du \end{aligned} \tag{110}$$

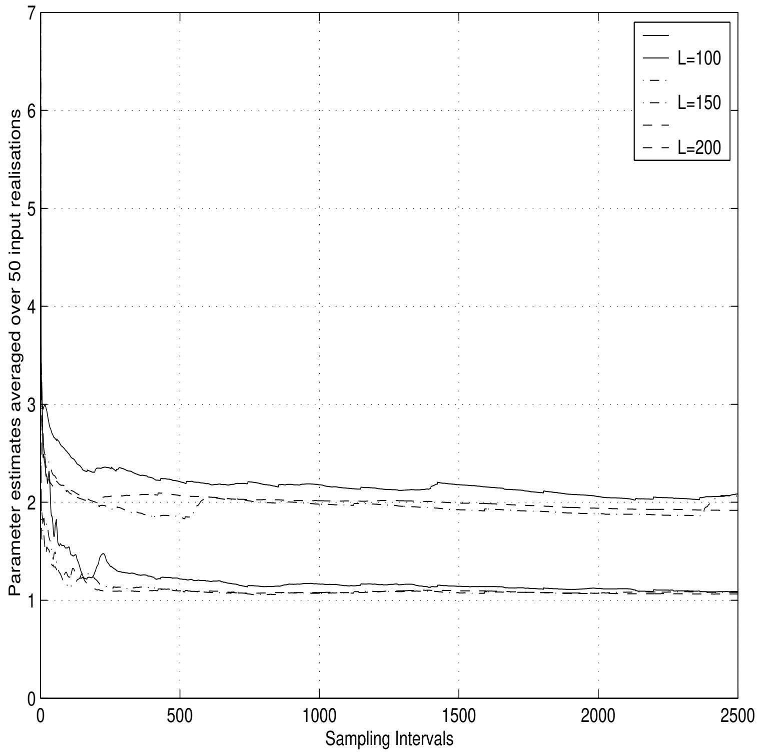

Figure 3: Convergence of the the averaged parameter estimates $\hat{\alpha}_m^L(N)$ and $\hat{\kappa}_m^L(N))$ to $\alpha_0 = 2$ m/s^2 and $\kappa_0 = 1$ J/msK, respectively, for different values of the window length L, $L = 100, 150, 200$.

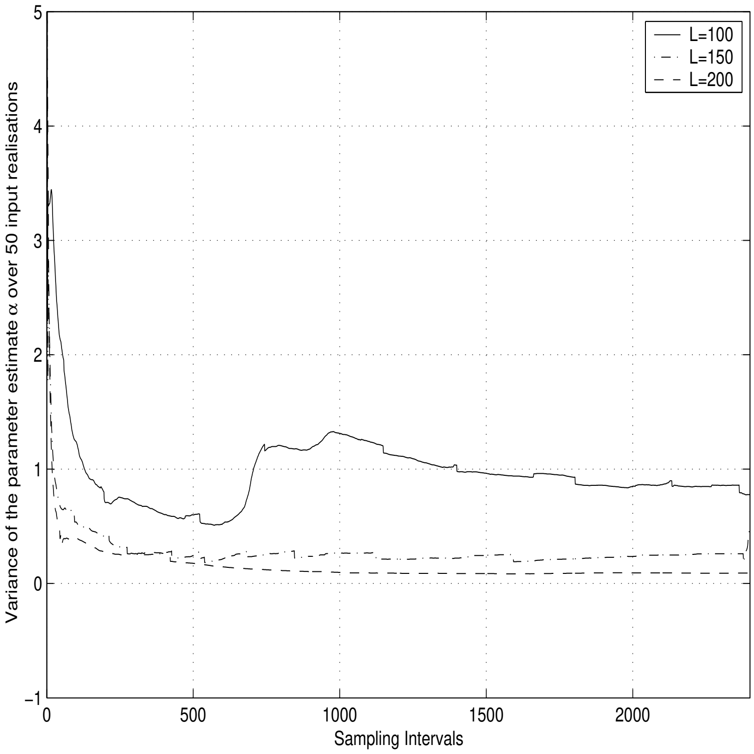

Figure 4: Variance of the averaged parameter estimate $\hat{\alpha}_m^L(N)$, taken over 50 different input realisations for window lengths $L = 100, 150, 200$.

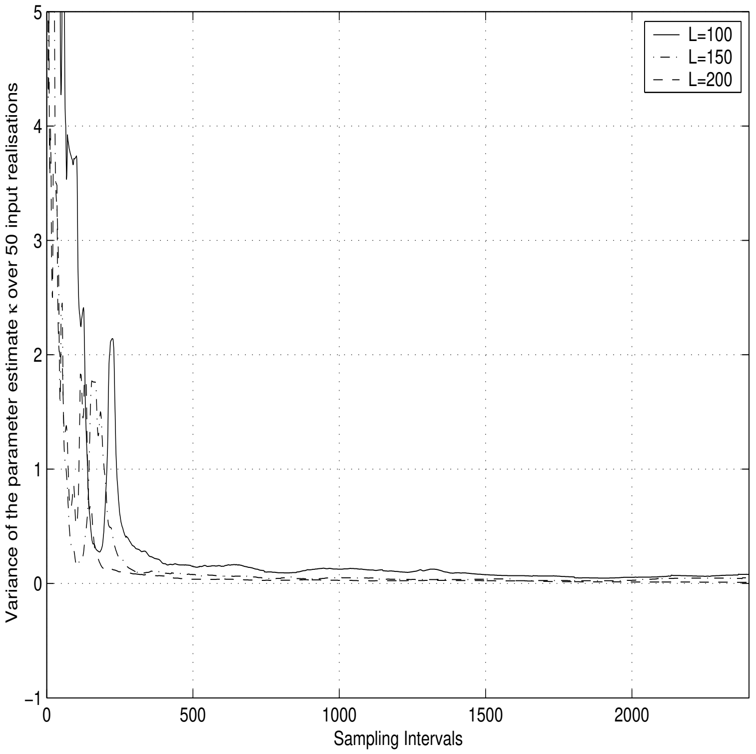

Figure 5: Variance of the averaged parameter estimate $\hat{\kappa}_m^L(N)$, taken over 50 different input realisations for the window lengths $L = 100, 150, 200$.

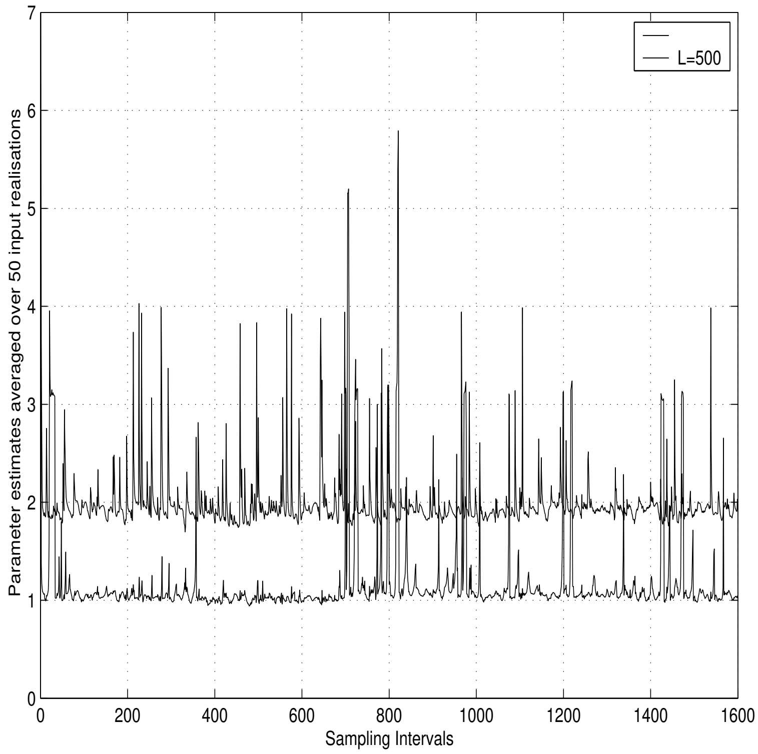

Figure 6: Convergence of the averaged parameter estimates $\hat{\alpha}_m^L(N)$ and $\hat{\kappa}_m^L(N)$, obtained by using WFD-R algorithm, to $\alpha_0 = 2$ m/s^2 and $\kappa_0 = 1$ J/msK respectively, for a window of length $L = 500$.

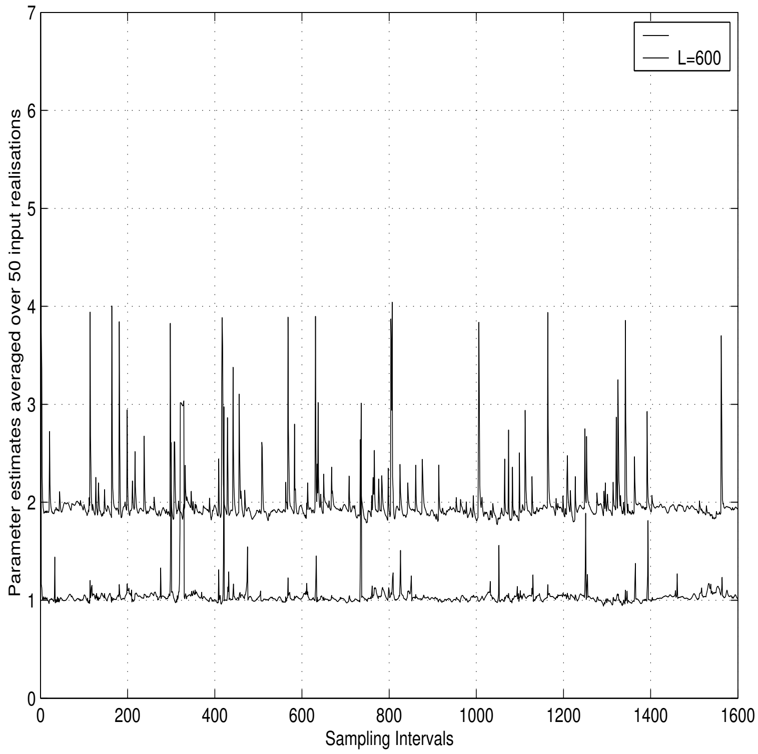

Figure 7: Convergence of the averaged parameter estimates $\hat{\alpha}_m^L(N)$ and $\hat{\kappa}_m^L(N)$, obtained by using WFD-R algorithm, to $\alpha_0 = 2$ m/s^2 and $\kappa_0 = 1$ J/msK respectively, for a window of length $L = 600$.

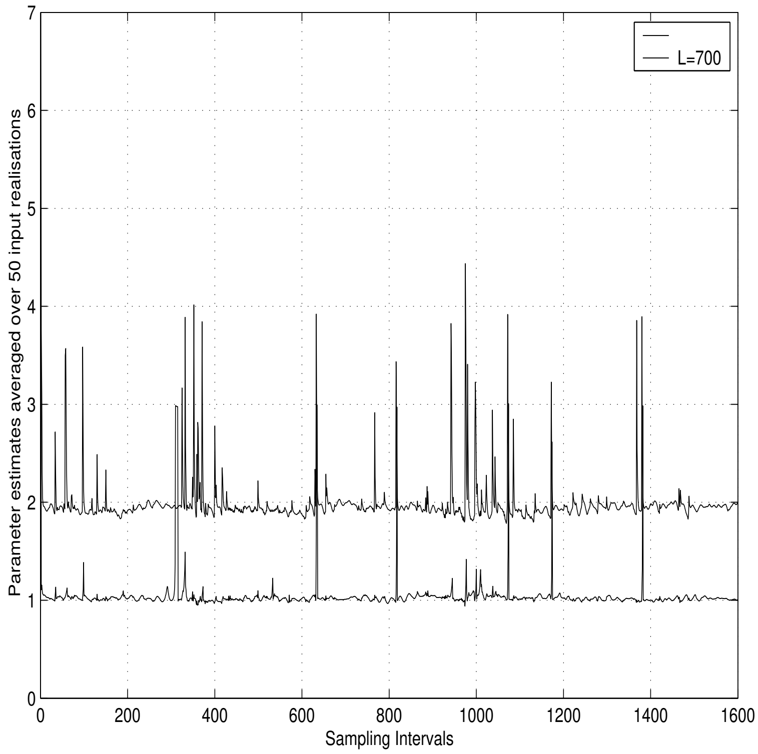

Figure 8: Convergence of the parameter estimates $\hat{\alpha}_m^L(N)$ and $\hat{\kappa}_m^L(N)$, obtained by using WFD-R algorithm, to $\alpha_0 = 2$ m/s^2 and $\kappa_0 = 1$ J/msK, for a window of length $L = 700$.

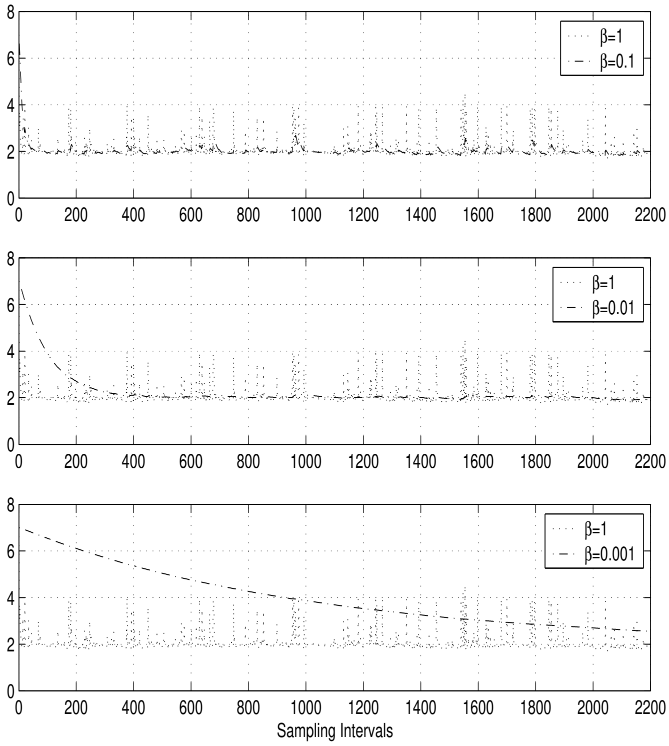

Figure 9: Filtered parameter estimates $\hat{\hat{\alpha}}(N)$ of the true of the parameter $\alpha_0 = 2$ obtained by passing $\hat{\theta}_m^L(N)$, where $L = 500$, thorough the filter $H(q^{-1};\beta)$, see (104) and (105), for different values of the filter parameter β.

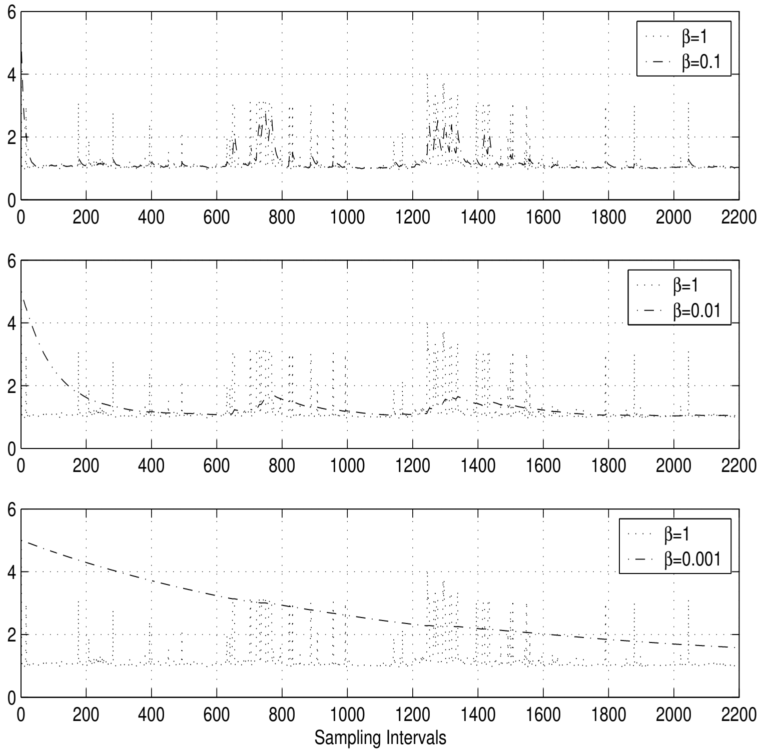

Figure 10: Filtered parameter estimates $\hat{\hat{\kappa}}(N)$ of the true of the parameter $\alpha_0 = 2$ obtained by passing $\hat{\theta}_m^L(N)$, where $L = 500$, thorough the filter $H(q^{-1};\beta)$, see (104) and (105), for different values of the filter parameter β.

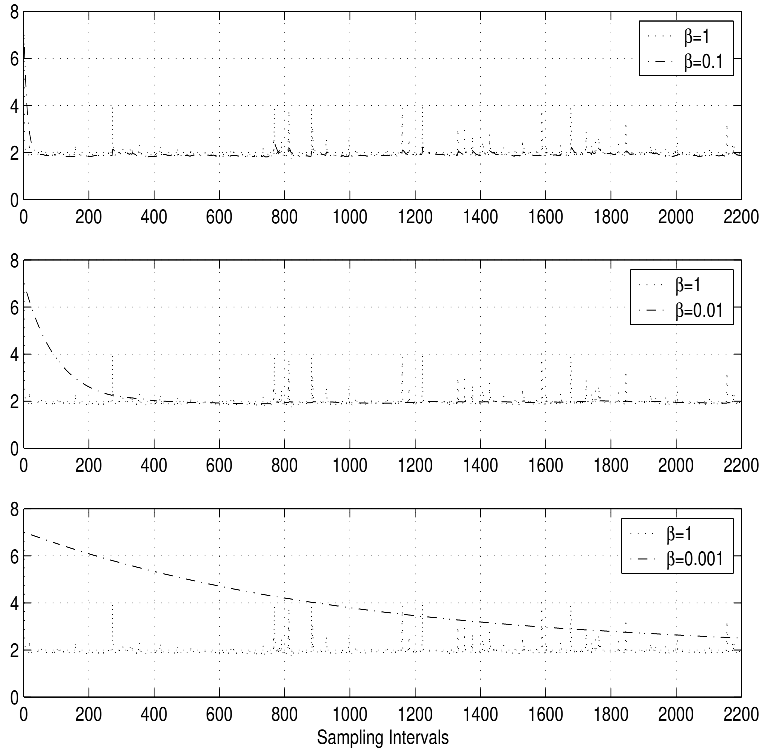

Figure 11: Filtered parameter estimates $\hat{\hat{\alpha}}(N)$ of the true of the parameter $\alpha_0 = 2$ obtained by passing $\hat{\theta}_m^L(N)$, where $L = 600$, thorough the filter $H(q^{-1};\beta)$, see (104) and (105), for different values of the filter parameter β.

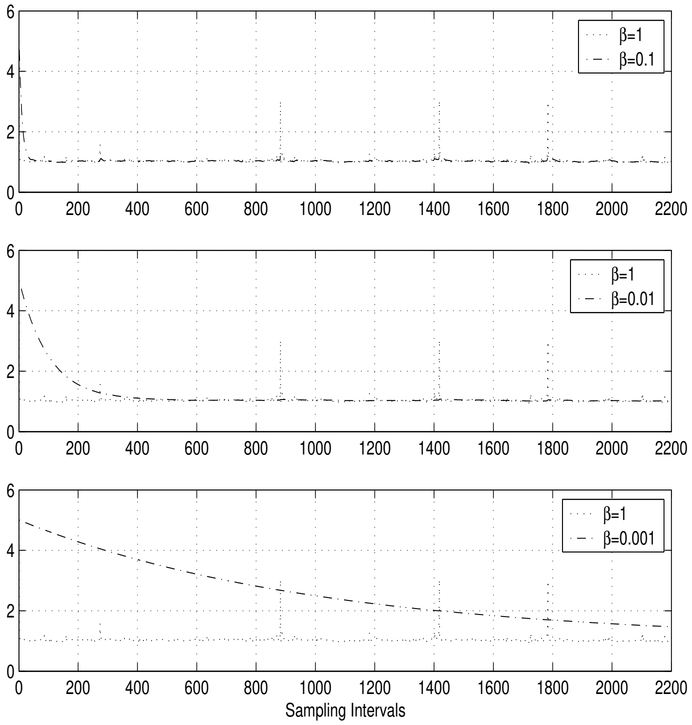

Figure 12: Filtered parameter estimates $\hat{\hat{\kappa}}(N)$ of the true of the parameter $\alpha_0 = 2$ obtained by passing $\hat{\theta}_m^L(N)$, where $L = 600$, thorough the filter $H(q^{-1};\beta)$, see (104) and (105), for different values of the filter parameter β.

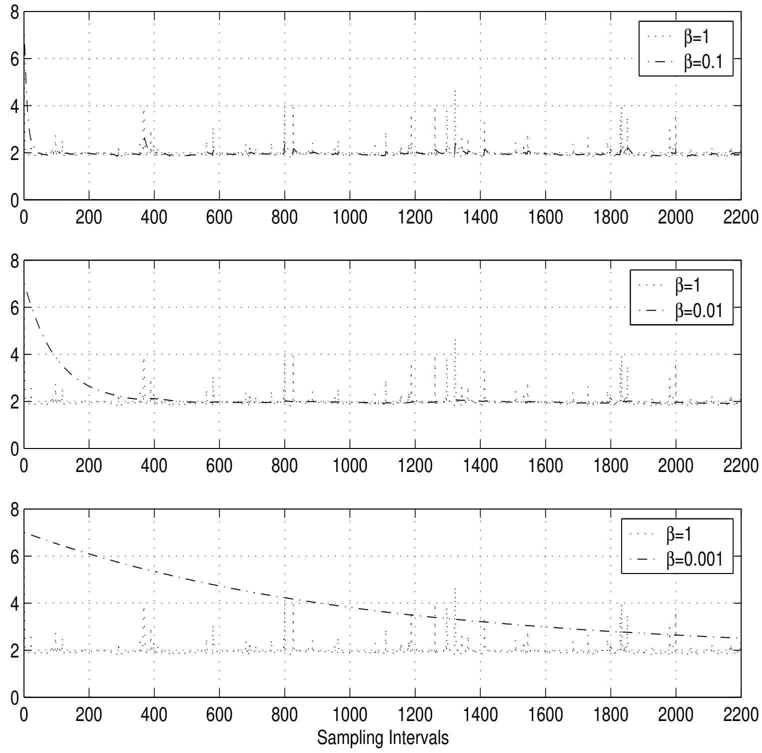

Figure 13: Filtered parameter estimates $\hat{\hat{\alpha}}(N)$ of the true of the parameter $\alpha_0 = 2$ obtained by passing $\hat{\theta}_m^L(N)$, where $L = 700$, thorough the filter $H(q^{-1};\beta)$, see (104) and (105), for different values of the filter parameter β.

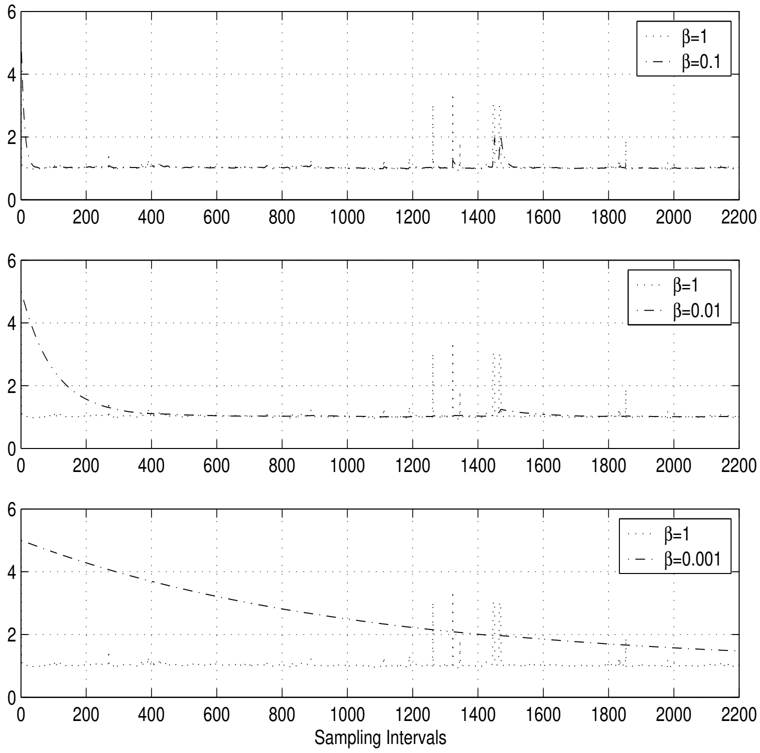

Figure 14: Filtered parameter estimates $\hat{\hat{\kappa}}(N)$ of the true of the parameter $\alpha_0 = 2$ obtained by passing $\hat{\theta}^L_m(N)$, where $L = 700$, thorough the filter $H(q^{-1}; \beta)$, see (104) and (105), for different values of the filter parameter β.

where A, B, C and D are $n \times n$, $n \times 2$, $1 \times n$ and 1×2 matrices respectively, with real entries, and with the matrices B and D having the following form,

$$B = [\frac{1}{\kappa}b_{j1} \ , \ b_{j2}] \ \ j = 1, 2, \ldots, n \tag{111}$$

$$D = [\frac{1}{\kappa}d_{11} \ , \ b_{12}]. \tag{112}$$

We then recursively estimate the parameters α and κ in the time domain by using the approximate model, (110), and the standard recursive prediction error method (RPEM), see [3]. It must be mentioned that, approximations of the type (110) can be constructed in several ways, in particular using standard PDE solvers. Here, as in [2], we performed the simulations with an approximate model constructed by using the Chebyshev collocation method, see [7], where it was observed in [7] that Chebyshev collocation gives a better fit than the Finite difference approximation and Chebyshev Galerkin methods.

We now estimate the parameter θ_0 from the data set $\{y^d(n)\}_{n=0}^{N-1}$ and $\{u^d(n)\}_{n=0}^{N-1}$, which is generated as mentioned in the previous subsection, using the time domain recursive algorithm (abbreviated as TR algorithm, from now on). For a detailed exposition on the implementation of the TR algorithm, we refer to [2].

In the simulation example considered here, as before, we do not allow the parameter estimates to get negative. If they do get negative at any instant we project them back into the positive quadrant by equating them to 0.1.

Here, as in the previous subsection, we estimate the parameters using the TR algorithm for 50 different input realisations, and present the average parameter estimates. In Figure 15, we have plotted the averaged parameter estimates α_t and κ_t, obtained by using the TR algorithm with an approximate model, constructed using Chebyshev collocation method, of order $n = 50$. Note that the parameter estimates get close to the true parameter without much variations, but at a slower rate as compared to the FD-R algorithm.

It can be noted form Figure 15 that there is a small but noticeable bias in the parameter estimates. It has been observed that increasing the model order neither improves the convergence rate nor appreciably reduces the noticeable bias, see Figure 16. Further simulations have also shown that increasing the model order beyond 70 renders the Chebyshev collocation approximating numerically unstable. Therefore we conclude that the noticeable bias is perhaps due to the slow convergence of the algorithm.

It must be mentioned that implementing the FD-R, and the WFD-R, algorithm was found to be much easier than implementing the TR algorithm. To implement the TR algorithm, one has to first construct a good approximation of the form (110) for the given PDE. Constructing a good approximation involves experimenting with a set a PDE solvers and choosing the best among them, see [1] for details. Further, to use the RPEM algorithm one has to not only set up the state space equations (110) but also the state equations of its derivatives with

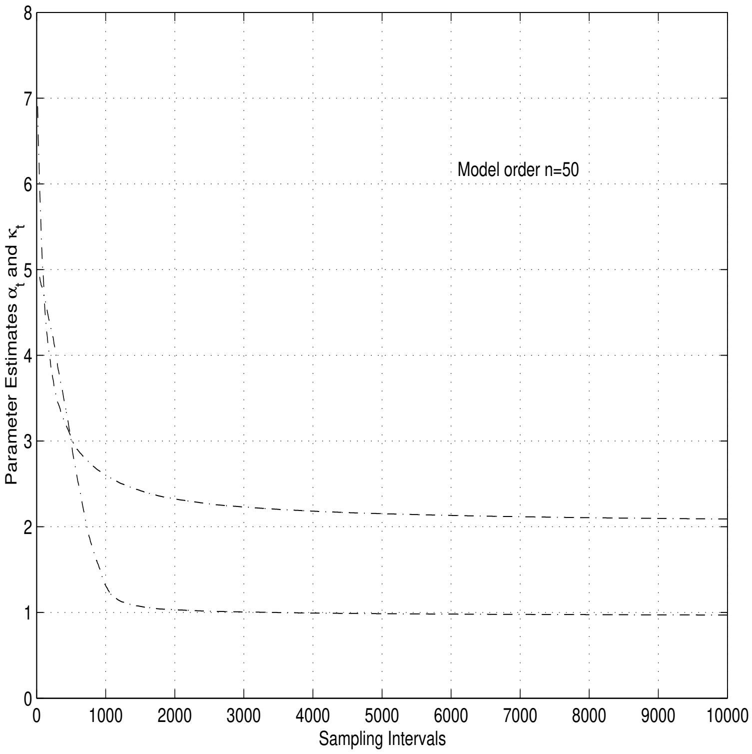

Figure 15: Averaged parameter estimates α_t and κ_t, for the true parameters $\alpha_0 = 2$ and $\kappa_0 = 1$ respectively, obtained by using the TR algorithm with an approximate model of order $n = 50$.

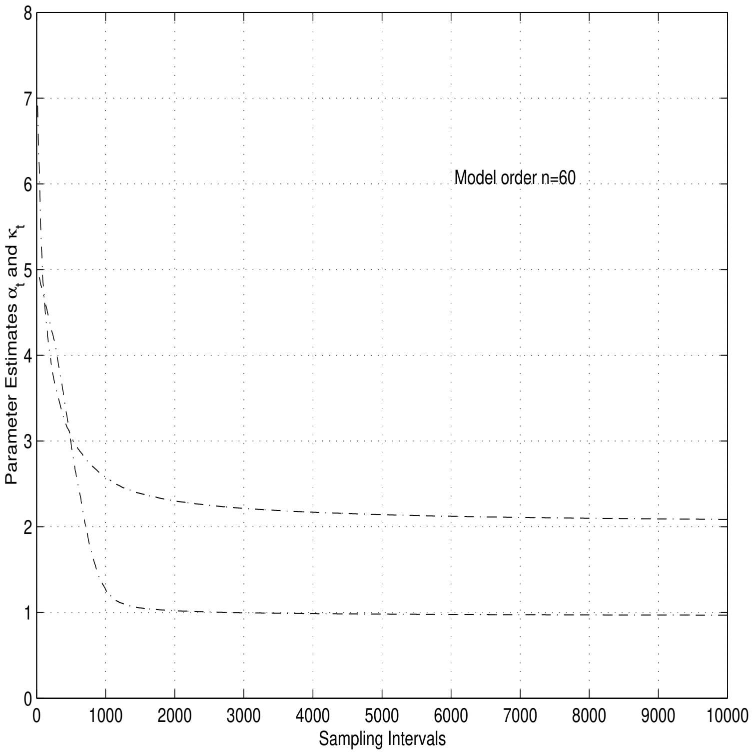

Figure 16: Averaged parameter estimates α_t and κ_t, for the true parameters $\alpha_0 = 2$ and $\kappa_0 = 1$ respectively, obtained by using the TR algorithm with an approximate model of order $n = 60$.

respect to θ, see [2] for details.

6 Conclusions

In this paper, we have proposed two recursive algorithms in the frequency domain, the FD-R and the WFD-R, and also briefly mentioned about a recursive algorithm in the time domain, the TR algorithm, to estimate the parameter $\theta_0 = (\alpha_0\ ,\ \kappa_0)^T$ of the given system.

Here in the FD-R and the WFD-R algorithms, we have exploited the facts that the given system has a very simple and a easily computable expression for the transfer function, (39), and over a sliding window one can use the DFT's of the system input and the output recursively, to device them. The recursive computation of the DFT's not only reduces the computational burden of the FD-R and the WFD-R algorithms, but also makes them elegant and easy to implement. It must be noted that, even though the recursions (79) and (80) and the algorithms have been derived in the frequency domain, the ordering of the data is in the time domain.

From the simulation example presented in the subsection 5.2, we can see that the FD-R algorithm converges quite rapidly to the true parameter θ_0 for all $L \geq 200$. In fact a window of length $L = 100 - 200$ seems to suffice for the chosen input signals (96) and (97). Note that in the simulation example, the band-limited assumptions (24) and (25), were not strictly followed.Therefore the FD-R alorithm has a wider scope, in practice, than the conservative assumptions made. The apparent reason for this wider scope is due to the low-pass filter characteristics of the system transfer-functions $G^{\infty}(s, \theta_0)$, which forces the system output $y(t)$ close to band-limitedness, with high frequency components having negligible magnitude, even when the input $u(t)$ contains certain high frequency components of moderate magnitude.

In the case of the WFD-R algorithm, the parameter estimates oscillate wildly around the true parameter θ_0, even for an L as large as $L = 700$. These oscillations were curtailed by passing the parameter estimates thorough a lowpass filter of the form (104). If one chooses the filter parameter β, see (104), to be very small, say of the order of 10^{-2}, then the oscillations of the parameter estimates obtained by using the WFD-R are curtailed to a large extent, but at the expense of decreasing the rate of convergence of the parameter estimates.

In the case of the TR algorithm, one can see that the parameter estimates approach the true parameters more smoothly than the FD-R and the WFD-R algorithms, but at a slower rate. Further, if one uses the TR algorithm one has to go through a laborious implementation procedure.

It can be checked that each iteration of the FD-R and the WFD-R involves $11L$ multiplications and each iteration of the TR involves $25n$ multiplications. From the simulations done in the subsection 5.2, if one accepts that using an FD-R with $L = 200$ is as good using a TR with $n = 60$, then each iteration of $FD - R$ would involve 2200 multiplications and each iteration of the TR involves 1250 multiplications. This would imply that TR is computationally less complex than the FD-R and the WFD-R. Nevertheless, this difference in computations adds a negligible burden to a modern computer. Moreover, here we have used the Chebyshev collocation method to construct the approximate model (110) in the case of the TR algorithm. The choice of using the Chebyshev collocation method was arrived at after doing considerable research and proving numerically that Chebyshev collocation is better suited for the chosen diffusion system than many popular PDE solvers, see [1]. If one would use a PDE solver like Finite difference scheme to construct the approximate model (110), see [5], the computational complexities, would have been more or less the same, or perhaps more in the case of the TR.

References

[1] B. Bhikkaji and T. Söderström. Reduced Order Models for Diffusion Systems. *International Journal of Control*, 74(15):1543–1557, 2001.

[2] B. Bhikkaji and T. Söderström. Recursive algorithm for estimating the parameters in one dimensional heat diffusion system. *Reglermöte*, 2002. Linköping, Sweden, 20-30 May.

[3] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification.* The MIT Press, Cambridge, Massachusetts, 1983.

[4] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing.* Printice Hall-Inc, New Jersey, 1975.

[5] S. Remle. Modeling and Parameter Estimation of the Diffusion Equation. *Licentiate thesis, Department of Information Technology, Uppsala University, Uppsala*, 2000.

[6] S. K. Mitra and J. F. Kaiser. *Handbook for Digital Signal Processing.* John Wiley and Sons, New York, 1993.

[7] T. Söderström and B. Bhikkaji. Reduced Order Models for Diffusion Systems via Collocation Methods. *Proc of 12th IFAC Symposium on System Identification*, 2000. Santa Barbara, CA, 21-23 June.

[8] T. Söderström and S. Remle. Parameter Estimation for Diffusion Models. *Proc of 12th IFAC Symposium on System Identification*, 2000. Santa Barbara, CA, 21-23 June.

[9] T. Söderström and P. Stoica. *System Identification.* Prentice Hall International, Hemel Hempstead, UK, 1989.