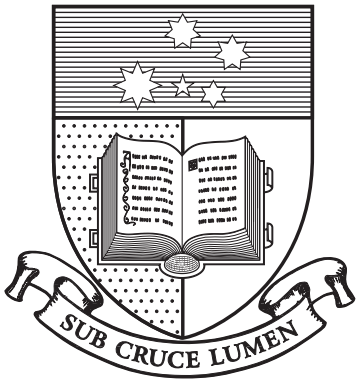

Laplace-Domain Analysis of Fluid Line Networks with Applications to Time-Domain Simulation and System Parameter Identification

by
Aaron C. Zecchin
B.E. (Civil) (Hons), B.Sc.

Thesis submitted to The University of Adelaide
School of Civil, Environmental & Mining
Engineering in fulfilment of the requirements for
the degree of
Doctor of Philosophy

Chapter 5

Time-Domain Simulation via the Inverse Laplace Transform

5.1 Introduction

From an applications perspective, the use of fluid line network models for the time-domain simulation of such networks is of primary importance. The main focus of the preceding chapters has been the characterisation of the frequency-domain behaviour of transient fluid line networks, by way of the Laplace-domain representation of the network dynamics. This chapter focuses on the use of the network models from Chapters 3 and 4 as the basis of a time-domain simulation model by way of the numerical inverse Laplace transform (NILT).

As discussed in Chapter 2, the complexity of the 1-D fluid line equations has necessitated the use of numerical partial differential equation (PDE) solvers to model the time-domain behaviour of such systems. Methods that have attracted particular interest are the method of characteristics (MOC) [*Chaudhry*, 1987; *Wylie and Streeter*, 1993], finite difference methods (FDMs) [*Chaudhry and Husssini*, 1985; *Kim*, 2008], and finite volume methods (FVMs) [*Zhao and Ghidaoui*, 2004]. For network modelling applications, these discrete PDE solvers serve as the basis for network models, which are constructed as a series of independent PDEs with coupled boundary conditions [*i.e.* discrete versions of the network equations (3.3)-(3.9)]. As discussed in Chapters 3 and 4, the vast majority of the research literature has focused on this approach to the time-domain simulation of pipe-networks (*e.g.* [*Karney*, 1984; *Chaudhry*, 1987; *Wylie and Streeter*, 1993; *Axworthy*, 1997; *Izquierdo and Iglesias*, 2004]).

Despite the utility of these discrete methods, their application to modelling pipeline networks suffers from the common issues associated with discrete meth-

ods, namely (i) difficulty in computational grid design, (ii) stability, (iii) accuracy, and (iv) computational efficiency [*Ghidaoui*, 1993; *Ghidaoui et al.*, 1998]. Point (i) is a major issue specific to multi-pipe systems, where the difficulty is associated with capturing the wave propagation effects as described by the hyperbolic PDEs at uniform time intervals [*Ghidaoui and Karney*, 1994]. To avoid the high computational cost associated with designing a grid that satisfies the ideal Courant condition ($\mathbb{C}_r = 1$ for all pipes), numerical strategies such as time and space interpolation methods (*e.g. Wiggert and Sundquist* [1977]; *Goldberg and Wylie* [1983]; *Yang and Hsu* [1990]) and wavespeed adjustment (*e.g. Wylie and Streeter* [1993]; *Ghidaoui et al.* [1998]) have been employed. These methods, however, all have drawbacks that affect stability and accuracy [*Wiggert and Sundquist*, 1977; *Goldberg and Wylie*, 1983; *Chaudhry and Husssini*, 1985; *Ghidaoui and Karney*, 1994; *Ghidaoui et al.*, 1998]. In addition to the issues with the grid design (and the implied issues with points (ii)-(iv) above) the incorporation of more sophisticated dynamic effects, such as unsteady friction and viscoelasticity, can pose difficulties for discrete methods in terms of high computational times [*Trikha*, 1975; *Schohl*, 1993; *Vítkovský et al.*, 2004].

Since the first use of the Laplace transform (LT) to solve the the fluid line PDEs in the 1950's [*Brown*, 1962], the use of inverse Laplace transform (ILT) methods to yield time-domain models from the Laplace-domain counterparts has been a source of active research. The advantage that these approaches possess is that the issues mentioned associated with discrete methods are avoided. However, due to the complexity of the Laplace-domain representations (particularly in the case of unsteady-friction or viscoelastic models) these methods are themselves, in many cases, only approximations. But they have seen broad and successful application (see the references in Section 5.2 below).

Many of these methods represent elegant and novel approaches to inverting the LT of the fluid line equations. A limitation, however, is that most are formulated for only single pipelines, with few being formulated for a limited class of compound lines [*Margolis and Yang*, 1985; *Yang and Tobler*, 1991], and only one approach able to deal with the case of general networks [*Suo and Wylie*, 1989]. Therefore the extension of these specialised methods to arbitrary pipeline networks, comprised not only of pipes, but other hydraulic elements is not straight forward, and in some cases not possible. The focus of this chapter is on such arbitrary systems, and as such, an entirely numerical inversion approach is investigated.

The chapter is structured as follows. Section 5.2 surveys the literature and discusses the role that the ILT has played in the development and application of time-domain pipeline models. Section 5.3 outlines the framework for the proposed NILT

model based on the admittance matrix for the $(\mathcal{G}(\mathcal{N}, \Lambda), \mathcal{P})$ network of Chapter 3. Section 5.4 gives a brief overview of the vast literature on NILT methods and formulates the adopted method. A sensitivity analysis for parameters of the adopted method for single line test functions is presented in Section 5.5, while a detailed study of the NILT applied to a series of networks is presented in Section 5.6. Conclusions are given in Section 5.7.

5.2 Background

This section surveys selected papers that outline the role that the ILT has had on the development and application of time-domain hydraulic models from their Laplace-domain counterparts, where the focus is primarily on pipelines. To facilitate discussion, the papers are categorised into the following four categories, (i) unsteady friction operator development, (ii) transfer function series expansion methods, (iii) modal based approximations of transfer functions, (iv) direct numerical inversion methods, and (v) discrete schemes based on the ILT. Category (i) investigates the use the ILT has had in the development of the $\mathcal{R}$ operator (Definition 2.1) to describe higher dimensional unsteady shear stress effects, categories (ii)-(iv) consider the different approaches to inverting the transfer function models (2.47)-(2.50), and (v) considers a difference scheme based on the ILT.

5.2.1 Unsteady friction operator development

The other categories, to be discussed in Sections 5.2.2-5.2.5, focus on the application of ILT methods to create hydraulic simulation models for pipelines, or simple pipeline networks. However, the ILT has been applied in a different, and more fundamental, way to pipeline models. The LT is a useful tool in working with differential equations as it facilitates analytic development by turning difficult PDEs into easier to solve ordinary differential equations (ODEs). The development of many one dimensional (1-D) pipeline models have relied on the use of the LT, and its inverse, to derive expressions for the shear stress, as an operator on the mean velocity, under conditions more general than that permitted by a 1-D model (*i.e.* conditions where there is a radial distribution of the axial velocity). These are discussed below.

Despite the earlier work of many authors [*Iberall*, 1950; *Brown*, 1962; *Holmboe and Rouleau*, 1967; *Rouleau and Young*, 1965a] on the topic of 1-D shear stress operators derived for slightly compressible, axisymetric, elastic closed conduit flow, *Zielke* [1968] was the first to derive a shear stress expression for the incorporation into the 1-D time-domain momentum equation. Starting from the basis of the two

dimensional (2-D) Navier-Stokes equations, *Zielke* [1968] derived an expression for the LT of the pipe wall shear stress as an operator on the mean axial velocity, implying that the time-domain representation consisted of a convolution operation on the mean velocity. To determine a time-domain expression for the shear stress, *Zielke* [1968] observed that no inverse existed for the expression of the operator on the mean velocity, as the Laplace-domain operator was unbounded for $|s| \rightarrow \infty$. Ingeniously, *Zielke* [1968] realised that one had to treat the shear stress as an operator on the mean axial acceleration, which included a $1/s$ term in the Laplace-domain operator and for which the ILT existed. *Zielke* [1968] used the residue theorem[1] to the determine the ILT which yielded the elegantly simple expression given in Example 2.6.

The LT and ILT featured in a fundamental way in the extension of a Zielke-type weighting function to describe the frequency dependent dynamics in transient turbulent flow for smooth [*Vardy and Brown*, 2003] and rough [*Vardy and Brown*, 2004] walled pipes. Under the assumption of a *frozen* eddy viscosity profile, *Vardy and Brown* [2003] used the LT to solve the 2-D Navier-Stokes equations for the relationship between the mean axial velocity and the cross-sectional shear stress[2]. The resulting Laplace-domain transfer function could not be analytically inverted (as it contained modified Bessel's functions of the first and second kind), and hence, *Vardy and Brown* [2003] proposed an approximation to the Laplace-domain transfer function of the form $W(s) = A/\sqrt{s+B}$ for which the ILT for the time-domain weighting function is $w(t) = Ae^{-Bt}/\sqrt{\pi t}$. This was extended to rough walled pipes in *Vardy and Brown* [2004], and in *Vardy and Brown* [2007] a uniform approach for computing A and B as a function of $\mathbb{R}_e$ and the pipe relative roughness was proposed.

5.2.2 Transfer function analytic series expansion methods

Due to the difficulty in determining closed form expressions for the ILT of hyperbolic functions involving $Z_c(s)$ and $\Gamma(s)$ in the transfer function models (2.47)-(2.50), an approach that has been adopted by many authors is to use the series expansion expressions for the hyperbolic functions, and perform the ILT term by term.

In his seminal paper, generalising the work of *Iberall* [1950], *Brown* [1962] determined expressions for $Z_c(s)$ and $\Gamma(s)$ for axisymetric, isothermal, rigid walled fluid

[1] See *Kreyszig* [1999] for a recent discussion of the residue theorem.

[2] This was achieved by dividing the cross section into an inner core region in which the axisymetric Navier-Stokes equations in cylindrical coordinates were used, and an outer wall region where planar coordinates were used. This seperation of regions was taken in order to simplify the equations for the outer wall region where the viscosity is radially varying.

lines. In this work, *Brown* [1962] used a low frequency approximation for the Bessel's function ratio term in $Z_c(s)$ and $\Gamma(s)$. Taking only the first few terms in this approximation, an expression for the transient pressure response, in an infinitely long line, to an impulse and step perturbation of velocity was analytically determined. This involved using the series approximations for $Z_c(s)$ and $\Gamma(s)$ to determine the analytic ILT to expressions of the form $Z_c^{-1}(s)\exp\{-\Gamma(s)\}$. This approximation has been termed Brown's approximation.

Using a similar expansion method, this approach was extended to a reservoir-pipe-valve (R-P-V) system, subjected to a sudden valve closure, by *Holmboe and Rouleau* [1967]. Based on similar approximations, [*Karam Jr. and Leonard*, 1973] used an operator block diagram framework to construct an impulse response function for a single finite length line. *Muto and Takahashi* [1985] extended the work of *Brown* [1962] to deal with the case of series pipelines. In this work, the full infinite series expressions were retained, to enable the approximation to be applicable over a longer time scale. Based on the work of *Muto et al.* [1981], *Muto and Kayukawa* [1986] used Brown's approximation to determine the transient response of a tapered pipe for pulse and step inputs.

A simple alternative to Brown's approximation, suitable for low frequency high damping systems was proposed in *Woods* [1983]. This approach used a first order square-root approximation to the Bessel's function ratio. Upon obtaining the Laplace-domain solution for a linear friction line with a pulse perturbation, *Ansari and Oldenburger* [1967] analytically determined the ILT for the case of a semi-infinite line, and used an infinite series expansion for the case of the R-P-V system.

An alternative series expansion method is that proposed in *Oldenburger and Goodson* [1964] and *Goodson* [1970]. Instead of dealing with expansions of $Z_c(s)$ and $\Gamma(s)$ directly, these authors proposed the use of infinite product expansions of the hyperbolic functions that appear in the transfer matrix expressions (2.47)-(2.50). *Oldenburger and Goodson* [1964] mainly focused on a single line terminated by a orifice or a tank, which involved series approximations to terms of the form

$$\cosh(as) + b\sinh(as), \quad a, b \in \mathbb{R}_+,$$

where each term was inverted analytically. *Goodson* [1970] presented this work in a more general context and discussed a range of different PDE applications. Numerical applications of the infinite product expansion method to a R-P-V system with linear friction showed it to be an amplitude preserving approximation. Due to the relatively complex expressions for the series terms, only simple expressions for $Z_c(s)$ and $\Gamma(s)$ were used (*i.e.* lossless, or linear friction).

A slightly different approach was proposed in *Rouleau and Young* [1965a] and *Rouleau and Young* [1965b] for inviscid and viscous liquids, respectively, in tapered pipes. From the original Laplace-domain ODEs, power series expressions were determined for $\Gamma(s)$ and $Z_c(s)$, which lead to analytically computable inverse transforms for the propagation expressions $\exp\{-\Gamma(s)x\}$. *Tarantine and Rouleau* [1967] generalised this work to deal with arbitrary tapers approximated by a series of step changes.

5.2.3 Modal approximations of transfer functions

The modal methods are termed as such as they involve approximating the transcendental functions in the matrix equations (2.47)-(2.50) by a finite number of modal terms. This modal structure is used to motivate the form of a rational approximation. The advantage of such a rational approximation is that it can be incorporated into standard state-space methods as it has a finite dimensional state space.

The origin of the modal approach is *Hullender and Healey* [1981]. In this work they combined the infinite product expansions of the transcendental functions with an m-th order product expansion of the Bessel's function ratio in $\Gamma(s)$ to yield expressions for the transcendental functions as an infinite product of ratios of polynomials of order m. In this approach, they recognised that, not only are there the modal poles on a line parallel to the imaginary axis, but that there are additional *dissipative* poles along the negative real line, which are associated with the unsteady friction dynamics. The modal terms are of the form

$$\frac{a_i s + b_i}{s^2 + c_i s + d_i} \tag{5.1}$$

for which the poles are complex conjugates (the coefficients a_i, b_i, c_i and d_i are functions of the dissipation number $\nu l/c_o r_o^2$, and the line impedance $\rho c_o/\pi r_o^2$), and the dissipation terms are of the form

$$\frac{e_i}{s + f_i}$$

for which the poles lie on the negative real line. In nondimensionalising the polynomial expressions, *Hullender and Healey* [1981] tabulated values of the polynomial coefficients as a function of the mode number, and a dimensionless parameter dependent on the dissipation properties of the system. These approximations were studied in *Hsue and Hullender* [1983] to determine the number of modes needed to satisfy given frequency-domain specifications. Using the modal approximation, *Hullender et al.* [1983] formulated a state-space model, which was successfully used

for time-domain simulation using a Runge-Kutta ODE solver.

Watton [1988] compared the modal method to three other time-domain approaches (FDM, FVM, and the MOC) in modelling a servo-valve transmission line system. These results showed that the modal method and the MOC where more accurate than the other two, with the modal method demonstrating greater numerical stability. The modal method from *Hullender and Healey* [1981] was extended to tapered pipelines by *Tahmeen et al.* [2001]. *Ayalew and Kulakowski* [2005] used this approach to derive the modal form for the admittance matrix organisation (2.48) of the transfer matrix equations.

The modal approximations from *Hullender and Healey* [1981] and *Hullender et al.* [1983] were used by *Margolis and Yang* [1985] and *Yang and Tobler* [1991] as the basis for a bond graph approach to modelling fluid line networks. Bond graphs provide a convenient and concise description of energy storage, transfer and dissipation of systems [*Karney*, 1990]. Approximating the fluid line dynamics by a finite number of modes lends itself conveniently to this framework. *Yang and Tobler* [1991] presented a simplification of the formulation in *Margolis and Yang* [1985] by considering only a linear friction model, the advantage of which being that the modal coefficients could be determined analytically. These methods were applied to simple systems consisting of pipes in series, branching pipes, and pipes terminated by lumped devices.

Based on the use of the approximation from *Woods* [1983], *Piche and Ellman* [1995] used the Puiseux expansion to develop an automatic analytic approach to computing the modal coefficients, thus avoiding the residue computations in *Hullender et al.* [1983]. Using an ODE solver, their approach compared favourably to that proposed in *Krus et al.* [1994]. *Chadwick* [1985] used a Fourier synthesis approach, combined with the Wentzel-Kramers-Brillouin (WKB) approximation to model the propagation of a pulse in an artery with leakage into side branches.

Considering the Laplace transform of the mass and momentum equations (2.38)-(2.39), *Makinen et al.* [2000] posed the problem of determining the distributions of pressure and flow in a variational framework. The Ritz method was used to determine an approximate solution to the variational problem. Using sinusoidal basis functions, the resulting transfer function expressions are of the same form as (5.1). An extension to the case of nonlinear shear stress $\tau \propto v|v|$ was given.

An alternative approach to generating a rational approximation to the fluid line transfer matrices was proposed by *Almondo and Sorli* [2006]. In their work, they modelled the transfer functions as a series of independent first order systems, where the coefficients were computed a using vector fitting approach. In the determination

of the model coefficients, special consideration was given to passivity requirements of the final approximate model. They applied their method to systems of pipes in series, branched pipes, and first-order looped pipes.

5.2.4 Direct numerical inversion of transfer functions

All of the methods discussed so far utilise some of the analytic properties of the fluid line transfer functions. The term direct numerical methods is used here to refer to methods that are based, primarily, on numerical approaches to the inversion of the Laplace transform.

Building on the work from *Brown* [1962], *Brown and Nelson* [1965] developed a series of pressure step response plots for semi-infinite fluid lines. The plots were developed from a combination of analytic and numerical approximations, where the low frequency components were covered by the approximation from *Brown* [1962] and a Bromwitch-type approximate numerical integration scheme was used for the high frequency components. An intensive book keeping method was proposed to deal with systems involving reflections from end conditions.

Obtaining the pipeline's impulse response function from a Fourier-type inversion to the pipeline transfer function, *Franke and Seyler* [1983] proposed the use of the impulse response convolution as a time-domain model for a viscoelastic pipeline. *Suo and Wylie* [1989] built on this idea by proposing the computation of the convolution in the frequency-domain, and the use of the inverse fast Fourier transform (IFFT) to determine the time-domain representation. This method, termed impulse response method (IPREM) was used by *Suo and Wylie* [1990] for modelling transients in viscoelastic rock-bored tunnels, and by *Kim* [2007, 2008] for application to modelling pipeline networks. In a similar manner, *Taylor et al.* [1997] used the inverse Fourier transform (IFT) applied to frequency dependent models [*Zielke*, 1968; *Vardy et al.*, 1993] as the base model for which a proposed FDM was assessed.

5.2.5 Discrete schemes based on Laplace inversion

The final category of ILTs method pertains to those methods for which the ILT serves as the basis for a temporal or spatial discrete method.

By way of the NILT, *Washio et al.* [1974a] computed the discrete-time step response for a series of two dimensional fluid line equations. This method was demonstrated to be more accurate than Brown's approximation and was extended to tapered lines [*Washio et al.*, 1974b], and lines involving nonlinear boundaries, such as air pockets, and orifices [*Washio et al.*, 1979]. The ability of this method to

deal with nonlinear boundary conditions relies on the sequential nature of the time point calculations.

In a gas dynamics application, *Kralik et al.* [1984a] developed a discrete spatial method by using a single mode approximation to the linear friction pipeline transfer functions to discrete segments of a pipeline. An interesting trade-off here is observed when comparing this method to the modal methods of Section 5.2.3. That is, rather than using transfer functions of a higher order (as is the case for the modal methods), the complexity of Kralik's approach is shifted to the use of many small reaches. This work was extended to networks in *Kralik et al.* [1984b], which was then used as the base simulation model for network state estimation applications in *Reddy et al.* [2006].

5.3 Framework for Network Time-Domain Simulations

A limitation of most of the existing time-domain simulation methods based on the ILT from Section 5.2 is the inability to deal with networks of an arbitrary structure. Most methods were formulated to deal with either single pipes, or compound lines with loops. Apart from the quasi-ILT methods of Section 5.2.5, the only purely ILT designed to deal with networks of an arbitrary structure is the IFFT based IPREM algorithm [*Suo and Wylie*, 1989]. An advantage and a limitation of the methods designed for networks of a simpler form is the emphasis placed on analytic representations of the time-domain behaviour, either closed form for simple cases, or as a series or modal expansion for more complex cases. The analytic forms possess benefits as they provide qualitative insight and are computationally efficient. However, as these methods are unsuitable for general network structures, hence purely numerical methods, such as IPREM, must be used.

As a completely general network simulation model is of interest here, a numerical inversion approach has been adopted within this research. The details of the numerical inversion approach are outlined in Section 5.4, but the structure of the Laplace-domain network equations are outlined here.

Consider a network $(\mathcal{G}(\mathcal{N}, \Lambda), \mathcal{P}, \mathcal{C})$ from Chapter 4 with node set $\mathcal{N}$, link set Λ, pipeline functions $\mathcal{P}$ and compound node functions $\mathcal{C}$. The time-domain simulation of such a network involves computing the time varying unknown states of a network for a given hydraulic scenario, where a hydraulic scenario is defined as well posed specification of the boundary conditions for the network[3] (Definitions 3.5 and 4.4).

[3]Note for this network problem to be well posed, it also requires the specification of the initial

Partitioning the node set $\mathcal{N}$ into compound nodes $\mathcal{N}_c$, junctions $\mathcal{N}_J$, demand nodes $\mathcal{N}_d$ and reservoirs $\mathcal{N}_r$, the required boundary conditions are the compound node controlled states $\boldsymbol{u} : \mathbb{R}_+ \mapsto \mathbb{R}^{n_u}$, the nodal demands $\boldsymbol{\theta}_d : \mathbb{R}_+ \mapsto \mathbb{R}^{n_u}$, and the reservoir pressures $\boldsymbol{\psi}_r : \mathbb{R}_+ \mapsto \mathbb{R}^{n_r}$. Chapters 3 and 4 demonstrated how the defined boundary conditions for the hydraulic scenario become the inputs for the Laplace-domain input/output (I/O) model, where the outputs are the unknown nodal states. Specifically, for the network $(\mathcal{G}(\mathcal{N}, \Lambda), \mathcal{P}, \mathcal{C})$ with the nodal partitioning as just defined, the nodal I/O model is given by

$$\begin{bmatrix} \boldsymbol{\Psi}_D(s) \\ \hdashline \boldsymbol{\Theta}_r(s) \end{bmatrix} = \boldsymbol{H}(s) \begin{bmatrix} \boldsymbol{U}(s) \\ \hdashline \boldsymbol{\Theta}_d(s) \\ \hdashline \boldsymbol{\Psi}_r(s) \end{bmatrix} \tag{5.2}$$

where the inputs are the controlled compound node states $\boldsymbol{U}$, the demand node controlled flows $\boldsymbol{\Theta}_d$, and the reservoir pressures $\boldsymbol{\Psi}_r$, the outputs are the nodal pressures $\boldsymbol{\Psi}_D$ for the nodes in $\mathcal{N}_D = \mathcal{N}_c \cup \mathcal{N}_J \cup \mathcal{N}_d$ and the reservoir nodal flows $\boldsymbol{\Theta}_r$, and where $\boldsymbol{H}(s)$ is the transfer matrix map derived from the network admittance matrix (see Section 4.6 for more details).

Given the above notation, of interest in this chapter is the construction of the functions $\boldsymbol{\psi}_D(t) = \mathcal{L}^{-1}\{\boldsymbol{\Psi}_D\}(t)$ and $\boldsymbol{\theta}_r(t) = \mathcal{L}^{-1}\{\boldsymbol{\Theta}_r\}(t)$ from the specification of the functions $\boldsymbol{u}(t)$, $\boldsymbol{\theta}_d(t)$, and $\boldsymbol{\psi}_r(t)$. By the convolution theorem of the ILT [*Franklin et al.*, 2001], the time-domain representation of (5.2) is given by

$$\begin{bmatrix} \boldsymbol{\psi}_D(t) \\ \hdashline \boldsymbol{\theta}_r(t) \end{bmatrix} = \int_0^t \boldsymbol{h}(t-\tau) \begin{bmatrix} \boldsymbol{u}(\tau) \\ \hdashline \boldsymbol{\theta}_d(\tau) \\ \hdashline \boldsymbol{\psi}_r(\tau) \end{bmatrix} d\tau \tag{5.3}$$

where the lower case symbols are the time-domain counterparts of their Laplace transforms. Since the impulse response matrix $\boldsymbol{h}$ is not analytically available, for any time point t, the computation of (5.3) first requires (i) the computation of $\boldsymbol{h}(\tau)$ on $\tau \in [0, t]$ via the ILT of $\boldsymbol{H}(s)$, and (ii) the convolution operation of $\boldsymbol{h}$ with the inputs. That is, the outputs are computed by

$$\int_0^t \mathcal{L}^{-1}\{\boldsymbol{H}(s)\}(t-\tau) \begin{bmatrix} \boldsymbol{u}(\tau) \\ \hdashline \boldsymbol{\theta}_d(\tau) \\ \hdashline \boldsymbol{\psi}_r(\tau) \end{bmatrix} d\tau. \tag{5.4}$$

In the interest of computational efficiency, (5.4) can be calculated more efficiently

conditions, but as outlined in Section 3.4, the initial state is typically the steady-state solution, implying homogeneous initial conditions for the temporal fluctuations, which are of interest here.

by computing the impulse response $\mathcal{L}^{-1}\{\boldsymbol{H}(s)\}(\tau), \tau \in [0,t]$ *a priori* so that only the convolution would require computation at each time point. This approach still requires the NILT of each elemental function of $\boldsymbol{H}$, which is still very computationally expensive. A more computationally efficient strategy of calculating (5.3) is

$$\begin{bmatrix} \boldsymbol{\psi}_D(t) \\ \hdashline \boldsymbol{\theta}_r(t) \end{bmatrix} = \mathcal{L}^{-1}\left\{ \boldsymbol{H}(s) \begin{bmatrix} \boldsymbol{U}(s) \\ \hdashline \boldsymbol{\Theta}_d(s) \\ \hdashline \boldsymbol{\Psi}_r(s) \end{bmatrix} \right\}(t), \tag{5.5}$$

that is, the convolution is performed in the Laplace-domain, and the resulting function is then inverted. Despite the fact that both approaches involve the calculation of $\boldsymbol{H}(s)$ at discrete points along some contour $s = a + \mathrm{i}\omega_i, i = 0\ldots, N$, the calculation of (5.5) is computationally simpler than (5.4) on two accounts:

1. Equation (5.5) requires a far reduced number of NILT calculations. As mentioned (5.4) involves the NILT of all the elemental functions of $\boldsymbol{H}(s)$, which comprises $(n_c+n_J+n_d+n_r)\times(n_u+n_d+n_r)$ functions (where n_X is the number of elements in $\mathcal{N}_X$, and the subscript $X = c, J, d,$ or r, and n_u is the number of controlled compound node states). In comparison, for the computation of the matrix multiplication for (5.4) in the Laplace-domain, only $(n_c+n_J+n_d+n_r)$ functions require numerical inversion, which is $1/(n_u+n_d+n_r)$ of the NILTs required by (5.4).

2. Equation (5.4) requires the calculation of a convolution at each time point, and (5.5) does not. Despite the fact that, for a hydraulic scenario, the nodal boundary conditions $\boldsymbol{u}(t)$, $\boldsymbol{\theta}_d(t)$, and $\boldsymbol{\psi}_r(t)$ are typically specified as functions of time, the Laplace transforms are typically analytically available. This fact reinforces the computational efficiency of the convolution calculation in the Laplace-domain.

Therefore, given these merits (5.5) represents the framework of the NILT model adopted in this research.

5.4 Numerical Inverse Laplace Transform

Numerical methods for inverting Laplace-domain functions have been around for over 50 years [*Abate and Whitt*, 1992], where one of the biggest areas of application is in the calculation of probability functions [*Abate and Whitt*, 1999]. There exists an extensive literature in the area, but only a sketch of relevant references are given. For more information, the interested reader is referred to the survey papers *Abate*

and Whitt [1992, 1999]. As noted in *Abate and Valko* [2004], there are essentially four groupings of of algorithms for the numerical inversion of Laplace transforms, these are approaches based on (i) Fourier series expansions, (ii) Laguerre function expansions, (iii) Gavier functionals, and (iv) Bromwich contour deformation approaches. A unified framework tying all these methods together is given in *Abate and Whitt* [2006].

The Fourier series expansion method yields an expression for the time-domain function as a summation of damped sines and cosines, where the coefficients are the real and imaginary parts of the Laplace function along a pre-specified contour in the complex plane $\mathbb{C}$ [*Crump*, 1976; *Abate and Whitt*, 1992, 1995, 2006]. This method is numerically robust and error bound estimates exist [*Crump*, 1976]. Similarly to the Fourier series method, the Laguerre function expansions yield an expression for $f(t)$ where Laguerre functions are the basis function set as opposed to sinusoids, and the coefficients are determined from the Laguerre generating function [*Abate et al.*, 1996]. In comparing the Fourier series method with the Laguerre function expansions, *Abate et al.* [1996] found that the Fourier series method served as a more stable and robust method than the Laguerre approach for a broader class of functions, as the Laguerre approach is not suited to functions containing discontinuities. Based on the Post-Widder formula for the inverse Laplace transform, the Gavier functional method utilises a difference approximation to the derivative terms in the Post-Widder formula to obtain an expression for $f(t)$ as the limit of a sequence of Gavier functionals [*Stehfest*, 1970; *Abate and Valko*, 2004]. Due to the slow convergence of this sequence it is typically coupled with sequence accelerators [*Valko and Abate*, 2004]. A limitation of this method is that it is very sensitive to roundoff errors and, as such, it is only suitable for high precision computing [*Abate and Valko*, 2004]. The contour deformation approaches (*e.g.* Talbot's method [*Murli and Rizzardi*, 1990]) aim to minimise the numerical error of the inversion by deforming the Bromwich contour to wraps around the poles of the Laplace-domain function. To enable the wrapping, this method requires that there exists some upper bound to the imaginary part of all poles of $F(s)$.

The methodology adopted within this research is the Fourier series expansion method. This is due to its numerical robustness, and ability to deal with a broad class of functions. In particular, the limitations of the other methods for this application are as follows. For many hydraulic scenarios of interest, the networks are excited by step inputs or other sharp changes causing discontinuities in $f(t)$, which yield the methods of type (ii) unsuitable. Multiprecision computing[4] is not avail-

[4]Multiprecision computing enables an arbitrary arithmetic precision as opposed to the standard 64-bit.

able in most software, therefore the methods in category (iii) are not suitable. For hydraulic networks, the Laplace-domain functions contains an infinite number of poles distributed just left of the imaginary axis, thus there is no upper bound to imaginary part of all poles, therefore the methods of (iv) are not suitable for this application.

5.4.1 Numerical inversion procedure

A different derivation of the Fourier series method is given here, and is based primarily on the contour integration as apposed to the periodic function approach of *Crump* [1976]; *Abate and Whitt* [1992, 1995]. The same form is arrived at, but the interpretation as a trapezoidal integration is important for applications to the network model as discussed later. Firstly, the theoretical basis for the adopted NILT method is presented in the following theorem.

Theorem 5.1. ***Inverse Laplace Transform:*** *Given the Laplace-domain function $F(s)$ for which there are no singularities in the right half plane defined by $\mathbb{R}\mathrm{e}\{s\} > \alpha$, the inverse Laplace transform of this function, $f(t)$, is given by the Bromwich contour integral*

$$f(t) = \frac{1}{2\pi \mathrm{i}} \int_{a-\mathrm{i}\infty}^{a+\mathrm{i}\infty} F(s) e^{st} \mathrm{d}s \tag{5.6}$$

for any real number $a \geq \alpha$.

If $F(s)$ is the transform of a real valued function, then $\overline{F(s)} = F(\overline{s})$ and (5.6) can be expressed as the real integral on a semi-infinite domain *Crump* [1976]

$$f(t) = \frac{e^{at}}{\pi} \int_0^{\infty} \mathbb{R}\mathrm{e}\left\{F(a + \mathrm{i}\omega) e^{\mathrm{i}\omega t}\right\} \, \mathrm{d}\omega \tag{5.7}$$

where the variable of integration has changed from s to $\mathrm{i}\omega$. Expanding out the real and imaginary components, (5.7) can be expressed as

$$f(t) = \frac{e^{at}}{\pi} \int_0^{\infty} \mathbb{R}\mathrm{e}\left\{F(a + \mathrm{i}\omega)\right\} \cos(\omega t) - \mathbb{I}\mathrm{m}\left\{F(a + \mathrm{i}\omega)\right\} \sin(\omega t) \, \mathrm{d}\omega. \tag{5.8}$$

An approximate numerical computation of (5.8) can be achieved by the use of F at discrete points $F_k = F(a + \mathrm{i}k\Delta\omega), k = 0, 1, \ldots, N$, where $\Delta\omega$ is the discretisation

interval on the line $\mathbb{R}\mathrm{e}\{s\} = a$, as[5]

$$\begin{aligned} f(t) &\approx \tilde{f}(t|a, \Delta\omega, N) \\ &= \frac{e^{at}\Delta\omega}{\pi}\left[\frac{F_0}{2} + \sum_{k=1}^{N} \mathbb{R}\mathrm{e}\{F_k\}\cos(k\Delta\omega t) - \mathbb{I}\mathrm{m}\{F_k\}\sin(k\Delta\omega t)\right]. \end{aligned} \tag{5.9}$$

Alternative methods exist that increase the accuracy of (5.9) by taking the contour step size as $\Delta\omega = 1/t$, which effectively converts (5.9) into an alternating sequence for which convergence accelerators can be used (see *Abate and Whitt* [1992] for a discussion of this). However, in this case, the step size as a function of t means that the sequence of Laplace function points $\{F_k\}_{k=1}^{N}$ requires a complete recalculation for each t. In the network case, the computation of $F(s)$ at a given s requires the computation of the I/O transfer function $\boldsymbol{H}(s)$, which, by (4.50), involves the inversion of a $n_D \times n_D$ complex matrix. As a time-domain simulation typically involves the computation of $f(t)$ for many values of t, performing many matrix inversions for each value of t is very computationally expensive. Therefore, given the expensive nature of computing F, such acceleration methods based on $\Delta\omega = 1/t$ are not appropriate for the application here.

5.4.2 Error analysis

For the numerical approximation (5.9), ignoring numerical roundoff errors, the sources of error are the discretisation error $E_D(t|a, \Delta\omega, n)$ for the integration in the interval $\omega \in [0, \omega_N]$, and the truncation error $E_T(t|a, \Delta\omega, n)$ for the neglected region of integration on $\omega \in (\omega_N, \infty)$. Given these sources of error, the original function $f(t)$ is related to the approximation $\tilde{f}(t|a, \Delta\omega, N)$ by

$$f(t) = \tilde{f}(t|a, \Delta\omega, N) + E_D(t|a, \Delta\omega, N) + E_T(t|a, \Delta\omega, N)$$

for which the discretisation error can be expressed as [*Crump*, 1976; *Abate and Whitt*, 1995]

$$E_D(t|a, \Delta\omega, N) = -\sum_{k=1}^{\infty} e^{-\frac{2k}{\Delta\omega}a} f\left(\frac{2k}{\Delta\omega} + t\right) \tag{5.10}$$

and, given the the expression of the Bromwich integral (5.6), the truncation error is

$$E_T(t|a, \Delta\omega, N) = \frac{e^{at}}{\pi}\int_{\omega_N}^{\infty} \mathbb{R}\mathrm{e}\left\{F(a + \mathrm{i}\omega)e^{\mathrm{i}\omega t}\right\}\mathrm{d}\omega. \tag{5.11}$$

[5] A true trapezoidal approximation possesses a factor of 1/2 for the last term. However, as (5.9) results from the truncation of an infinite series (trapezoidal integration on a semi-infinite domain) this factor is not included.

Given the form (5.10), and the property of a Laplace transformable function that $|f(t)| \leq Me^{\sigma t}$, $M \in \mathbb{R}$, *Crump* [1976]; *Abate and Whitt* [1992, 1995] show that the discretisation error is bounded above by

$$E_D(t|a, \Delta\omega, N) \leq Me^{\alpha t - \frac{2}{\Delta\omega}(a-\alpha)}$$

for $0 < t < 2\pi/\Delta\omega$ (the case of $t = 0$ is considered separately), which implies that the discretisation error can be made arbitrarily small by choosing a to be a large positive number. However, as seen in (5.11), $E_T(t|a, \Delta\omega, N)$ contains an e^{at} term, which implies that, for some functions[6] F, $E_T(t|a, \Delta\omega, N)$ could potentially grow with a. This was observed in *Abate and Whitt* [1992], by the statement that a large a may make the computation of (5.9) more difficult by making the series harder to sum, or by increasing the roundoff error due to the larger terms involved. Given these issues, the parameter recommendations in *Crump* [1976] have not been used, but appropriate values of a, $\Delta\omega$ and N were determined by a detailed sensitivity analysis, as presented in the following section.

5.5 Parametric Study for Single Lines

As was outlined in the previous section, the numerical inverse Laplace transform (NILT) approximation $\tilde{f}(t|a, \Delta\omega, N)$ is parameterised by three parameters: the location of the integration contour, a; the frequency spacing, $\Delta\omega$; and the number of terms used in the integration approximation, N. Recommendations for these settings exist in the literature [*Crump*, 1976; *Abate and Whitt*, 1992], however, the test functions are typically simpler than the type of functions encountered in fluid lines. In order to assess appropriate parameter settings for pipeline applications, a sensitivity analysis of the error with respect to the parameter variations has been studied for some dimensionless single pipeline test functions. The use of these simple test functions facilitates a detailed analysis of the NILT error with respect not only to the parameters a, $\Delta\omega$, and N, but also to the energy dissipation properties of the pipeline.

5.5.1 Pipeline test functions

For the sensitivity analysis, a simple R-P-V system subjected to a unit delay of a step flow perturbation was considered. The pressure response at the valve for such

[6]The function types referred to here are those for which $\int_{\omega_N}^{\infty} \mathbb{R}\mathrm{e}\left\{F(a + \mathrm{i}\omega)e^{\mathrm{i}\omega t}\right\} \mathrm{d}\omega$ has a slower than exponential decay for increasing a. Unfortunately all function types tested within this thesis were observed to posses this property.

a system can be derived from (2.42)-(2.43) as [*Goodson and Leonard*, 1972]

$$F(s) = Z_c(s) \tanh \Gamma(s) \left(-\frac{e^{-s}}{s} \right) \tag{5.12}$$

where, as outlined in Section 2.4, the dynamic properties for different pipe types are described by the propagation operator $\Gamma(s) = \sqrt{R_0[s + R(s)]C_0[s + C(s)]}$ and the series impedance $Z_c(s) = \sqrt{R_0[s + R(s)]/C_0[s + C(s)]}$, and the resistance $R(s)$ and compliance $C(s)$ functions are as defined in Corollary 2.2. The selected pipeline types were the dimensionless versions[7] of the laminar-steady-friction (LSF) model from Example 2.3 and the linearised turbulent-steady-friction (TSF) from Example 2.4, and Zielke's laminar-unsteady-friction (LUF) model from Example 2.6. For these cases, the functions $R(s)$ and $C(s)$ have the form[8]

$$R(s) = \begin{cases} \mathbb{M}\mathfrak{R} & \text{for the LSF and the linear TSF models} \\ \mathbb{M}\mathfrak{R} + \dfrac{\mathbb{M}\mathfrak{R}}{2} \displaystyle\sum_{j=1}^{\infty} \frac{s}{s + \eta_j^2 \frac{\mathbb{M}\mathfrak{R}}{8}} & \text{for the LUF model} \end{cases}$$

$$C(s) = 0 \tag{5.13}$$

where $\mathbb{M}$ is the Mach number, and $\mathfrak{R} \in \mathbb{R}_+$ is the linear resistance number given by

$$\mathfrak{R} = \frac{L}{D} \times \begin{cases} 32\, \mathbb{R}_{\mathrm{e}}^{-1} & \text{for the LSF model} \\ f q_0 & \text{for the TSF model linearised about the operating point } q_0 \neq 0 \end{cases}.$$

where the symbols are as defined in Section 2.2. The advantage of this nondimensionalisation is that all cases are parameterised by a single parameter $\mathbb{M}\mathfrak{R}$, which is indicative of the dissipation properties of the system[9]. Minimising the parameter set for the pipeline aids greatly in determining the relationship between the NILT parameters and the pipeline properties by reducing the dimensionality of the sensitivity analysis. As demonstrated in *Zecchin et al.* [2009], for most real world systems, $\mathbb{M}\mathfrak{R}$ lies in the range $[O\{10^{-6}\}, O\{10^{0}\}]$.

The use of the LSF/linearised TSF and the LUF models allows for a comparison

[7] These models are mathematically equivalent, but they use dimensionless variables where nondimensionalisations were taken as in *Arfaie et al.* [1993].

[8] Note that the form for the LUF model is expressed here in a series expansion about the poles and opposed to the ratio of Bessel's functions as is typical for Laplace-domain representations [*Stecki and Davis*, 1986]. This alternative approach was taken to ensure that the Laplace-domain model was in fact the Laplace transform of the time-domain model for the MOC. Note also that the series summation was truncated to 20 terms.

[9] In some texts [*Goodson and Leonard*, 1972; *Arfaie et al.*, 1993], $\mathbb{M}\mathfrak{R}$ is in fact termed the dissipation number, and represented by the symbol D_n.

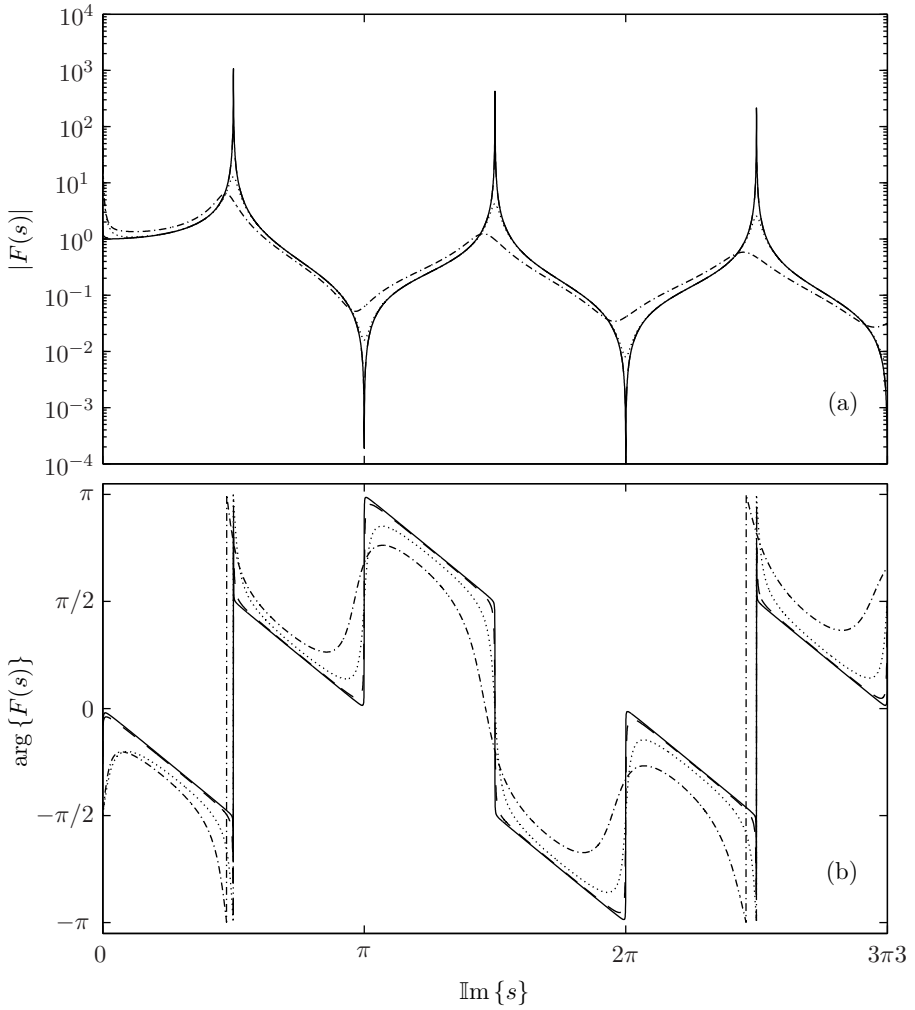

Figure 5.1: Plots of (a) $|F(s)|$ and (b) $\arg\{F(s)\}$, for the first three harmonics of F from (5.12), for both the linear TSF $(-)$,$(\cdots)$ and the LUF $(--)$,$(-\cdot-)$ for values of $\mathbb{M}\mathfrak{R} = 0.1, 0.001$ respectively.

of two different types of functions. Plots of the first three harmonics of (5.12), for both the TSF and the LUF are given for two values of $\mathbb{M}\mathfrak{R}$, in Figure 5.1, and the details of the real and imaginary parts of the first harmonic are given in Figure 5.2. The TSF function has a higher amplitude and sharper peaks than the smoother LUF model.

5.5.2 Analysis approach

As discussed in the analysis in Appendix A, along any contour parallel to the imaginary axis, (5.12) has harmonics spaced at approximately $s = a + \mathrm{i}\omega_n = 2(n+1)\pi/2, n = 1, 2, \ldots$ [*Zecchin et al.*, 2006]. Given this fact, the NILT parameters can be reformulated to intuitively relate to the function properties. That is, instead of using the number of discretisations (N) and the discretisation width ($\Delta\omega$), it is

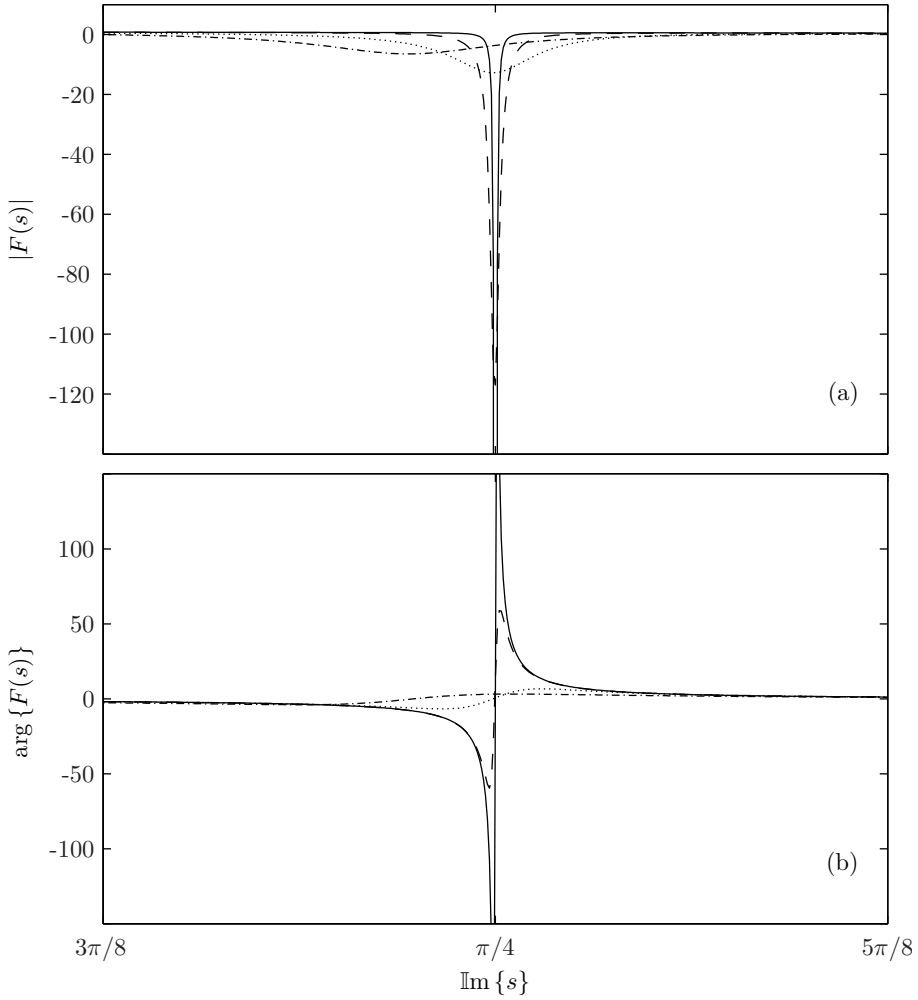

Figure 5.2: (a) Real and (b) imaginary parts of first harmonic of F from (5.12), for both the linearised TSF $(-)$,$(\cdots)$ and the LUF $(--)$,$(- \cdot -)$ for values of $\mathbb{M}\mathfrak{R} = 0.1, 0.001$ respectively.

more meaningful to consider the number of harmonics used in the inversion N_H, and the number of discretisations of each harmonic N_Δ, where the following relationships hold

$$N = N_H \cdot N_\Delta, \quad \Delta\omega = \frac{\pi}{2}\frac{1}{N_\Delta}.$$

In order to capture the time dependency of the accuracy of the NILT estimate $\tilde{f}(t, \mathbb{M}\mathfrak{R}|a, N_\Delta, N_H)$, the error measure was taken as the L_1 norm over a single pipe period. That is the error was measured as the L_1 norm over a window of length $\Delta t = 2$ (dimensionless pipe period) centred over the n-th period ($t = 4n - 2$). That is, for the sensitivity analysis, the adopted error measure, as a function of the pipeline dissipation number $\mathbb{M}\mathfrak{R}$, the NILT parameters (a, N_Δ and ,N_H), and the temporal pipe line period n is given by

$$e_n(\mathbb{M}\mathfrak{R}, a, N_\Delta, N_H) = ||f(\cdot, \mathbb{M}\mathfrak{R}) - \tilde{f}(\cdot, \mathbb{M}\mathfrak{R}|a, N_\Delta, N_H)||_{L_{1[4(n-1),4n]}} \tag{5.14}$$

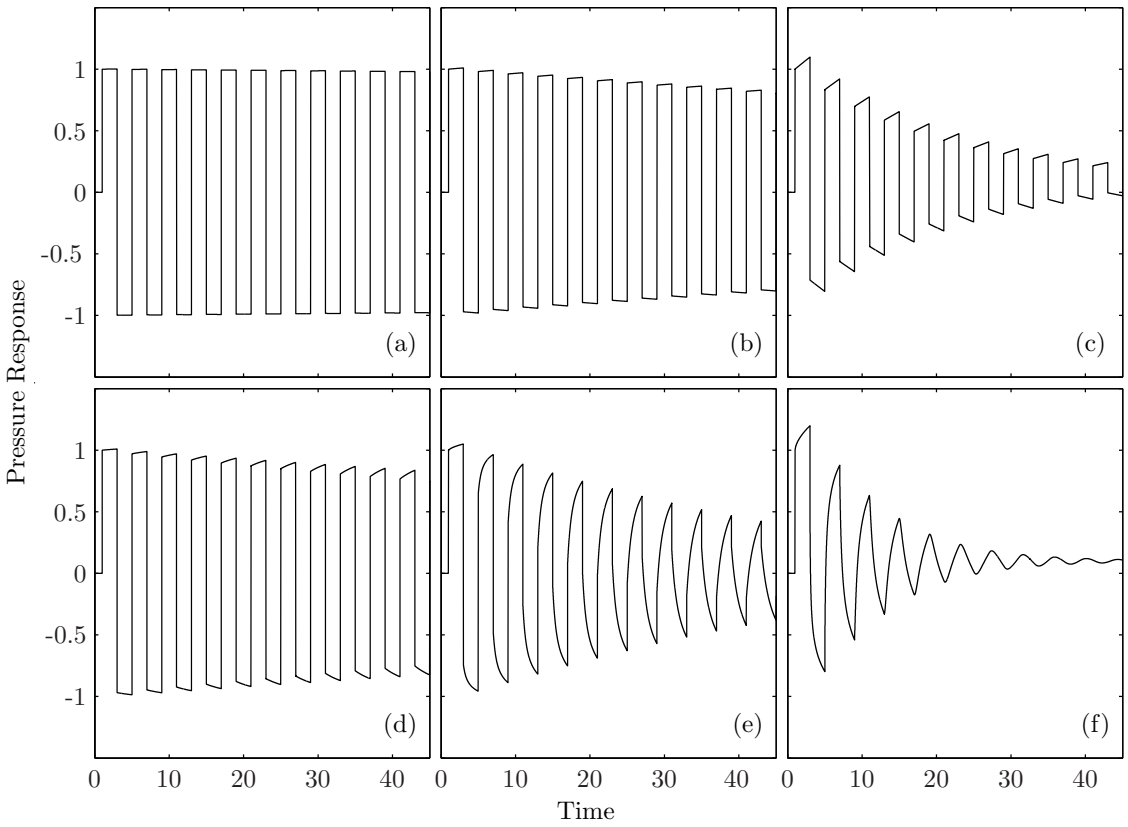

Figure 5.3: Time-domain representations of test functions for the LSF and linear TSF model [plots (a)-(c)] and Zielke's LUF model [plots (d)-(f)] for values of $\mathbb{M}\mathfrak{R} = 0.001$ [(a), (d)], $\mathbb{M}\mathfrak{R} = 0.01$ [(b), (e)], and $\mathbb{M}\mathfrak{R} = 0.1$ [(c), (f)]. The time-domain functions were computed using the MOC with 2000 spatial discretisations, and the LUF model was truncated to 20 terms.

which indicates the accuracy with which the NILT function approximates the function on the $f(t)$ on the interval $t \in [4(n-1), 4n]$.

Given this notation, the sensitivity analysis involves assessment of the error function $e_n(\mathbb{M}\mathfrak{R}, a, N_\Delta, N_H)$ for practical ranges of n and $\mathbb{M}\mathfrak{R}$ and feasible ranges of a, N_Δ and N_H, creating a five dimensional error surface. To undertake this analysis, the five-dimensional parameter space was discretised in each dimension according to the *a priori* observed sensitivity of e_n to the respective parameter. The following parameter ranges were used: $n \in \{1, 2, 5, 10\}$; $\mathbb{M}\mathfrak{R} \in \{0.001, 0.01, 0.1\}$; $a \in \{0.01, 0.02, \ldots, 0.15\}$; $N_\Delta \in \{21, 41, \ldots, 301\}$; $N_H \in \{10, 50, 100, 500, 1000\}$ leading to a total of 27 000 evaluations of e_n for the test functions.

The computation of $f(t, \mathbb{M}\mathfrak{R})$ was performed using the MOC where the dimensionless pipeline was discretised into 2000 reaches. Plots of the two kinds of test functions (LSF/linearised TSF and the LUF) for each value of $\mathbb{M}\mathfrak{R}$ are given in Figure 5.3.

5.5.3 Numerical results

The results of the sensitivity analysis computations are summarised in Figures 5.4 and 5.5 for the LSF/linear TSF and LUF models, respectively. These figures depict the behaviour of $e_n(\mathbb{MR}, a, N_\Delta, N_H)$ over the five-dimensional parameter space by the use of nested surface plots[10]. At the coarsest outer level, three nested surface plots are shown, one for each value of $\mathbb{MR} = 0.001$, 0.01, and 0.1. Each of these three plots shows 4×5 surface plots where each surface has a fixed value of n (varied along the horizontal axis) and N_H (varied along the vertical axis). At the finest scale, each surface shows the error $e_n(\mathbb{MR}, a, N_\Delta, N_H)$ over the range of N_Δ along the horizontal axis and a along the vertical axis. The lighter shade of a pixel indicates a lower value of e_n, and a darker shade indicates a higher value of e_n.

The first major observation was that there is a similar pattern for both test function types. The only differences being that the errors for the LUF model are generally smaller. This is attributed to the smoother nature of $F(s)$ for this model type (Figures 5.1 and 5.2) yielding an easier integration process. The same reasoning applied for the observed dependency of the error on the dissipation number $\mathbb{MR}$. That is, for any given parameter settings for a, N_Δ, N_H and values of n, e_n was consistently lower for higher dissipative test functions.

Concerning the number of harmonics N_H, clearly the greater the number of harmonics used in the approximation, the lower the error appears to be. A greater number of harmonics was observed to be particularly important for shorter time scales (*i.e.* lower values of n), as it is at these time scales where the pressure response is sharpest and the higher frequencies (higher order harmonics) play a more important role in the reconstruction of the time-domain signal. This phenomenon is observed in Figure 5.6 where the pressure response for different values of N_H is given for both test function types. The error is observed to manifest itself as oscillations in the approximation, particularly exaggerated at points of high rates of change. Similarly, for the less dissipative systems (*i.e.* lower values of $\mathbb{MR}$), a greater number of harmonics were generally needed for a comparative accuracy, this also attributed to the sharper time-domain signal resulting from the slower decay in the amplitudes of the harmonics.

The number of discretisations per harmonic N_Δ was one of the two parameters to be studies in detail (in Figures 5.4 and 5.5, N_Δ is varied from 21 to 301 along the local horizontal axis of each surface). As with N_H, the approximation is increasingly accurate for greater N_Δ. However, for lower N_Δ the discretisation of $F(s)$ is too

[10]That is, each pixel on each surface plot in Figures 5.5 for the LSF/linear TSF is interpreted as the value of e_n for a given period n, a given number of harmonics N_H, a given $\mathbb{MR}$, and given values of N_Δ and a.

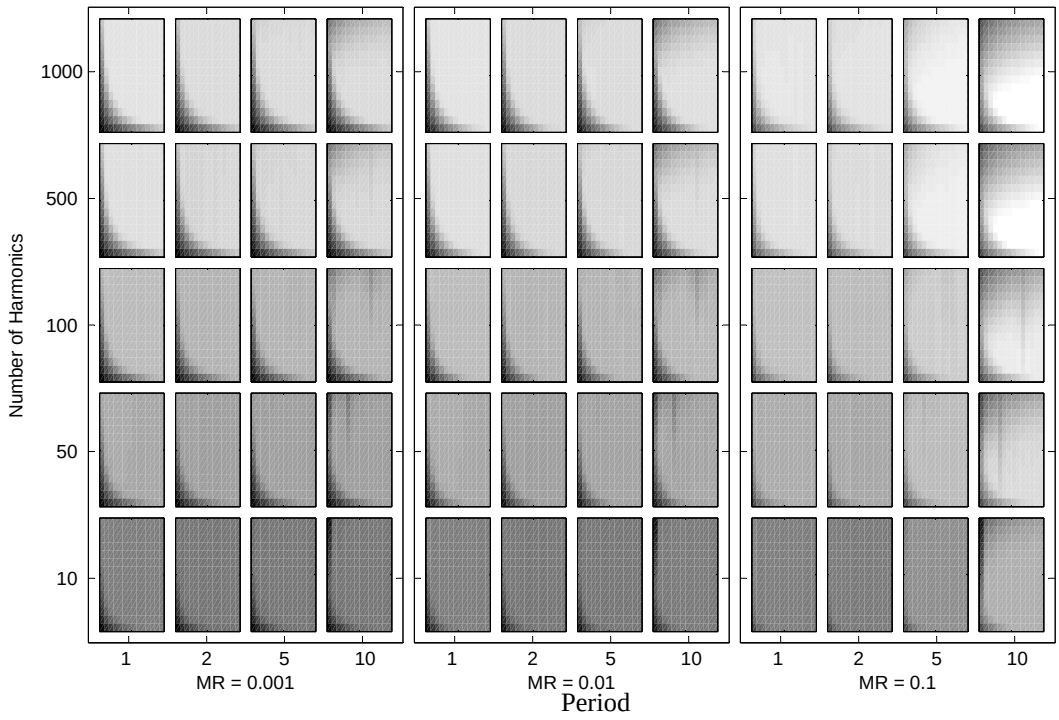

Figure 5.4: Sensitivity analysis of the NILT parameters for the dimensionless LSF/linear TSF model. Three plots of nested surface plots are shown, where each plot corresponds to a value of $\mathbb{MR} = 0.001, 0.01, 0.1$. Each nested surface plot shows the error $e_n(\mathbb{MR}, a, N_\Delta, N_H)$ over the range of N_Δ and a (on the local horizontal and vertical axes) for given values of the period n and N_H (on the outer horizontal and vertical axes). The lighter shade indicates lower values of e_n ($e_n < 0.01$ for white and $e_n > 10$ for black). For the nested surface plots, the value of N_Δ varies linearly from 21 to 301 in the horizontal direction, and the value of a varies linearly from 0.01 to 0.15 in the vertical direction.

coarse, and a bias is introduced into the NILT approximation. This bias arises from an inaccurate approximation in the integration of the low frequency component of the Laplace-transform and is particularly pronounced for lower values of $\mathbb{MR}$. This was to be expected as the lower $\mathbb{MR}$ test functions are more difficult to integrate due to the sharpness of the real and imaginary components of $F(s)$ (as in Figure 5.2). Examples of the bias induced from different harmonic discretisations are given in Figure 5.7.

The contour location a was the other parameter that has been studied in detail, as the NILT method was observed to be highly sensitive to this parameter (in Figures 5.4 and 5.5, a is varied from 0.01 to 0.15 along the local vertical axis of each surface). For low a, the integration contour lies closer to the poles of $F(s)$ meaning that the real and imaginary parts of $F(s)$ along this line contains sharper changes. This resulted in the necessity of a finer discretisation for a comparative accuracy in the

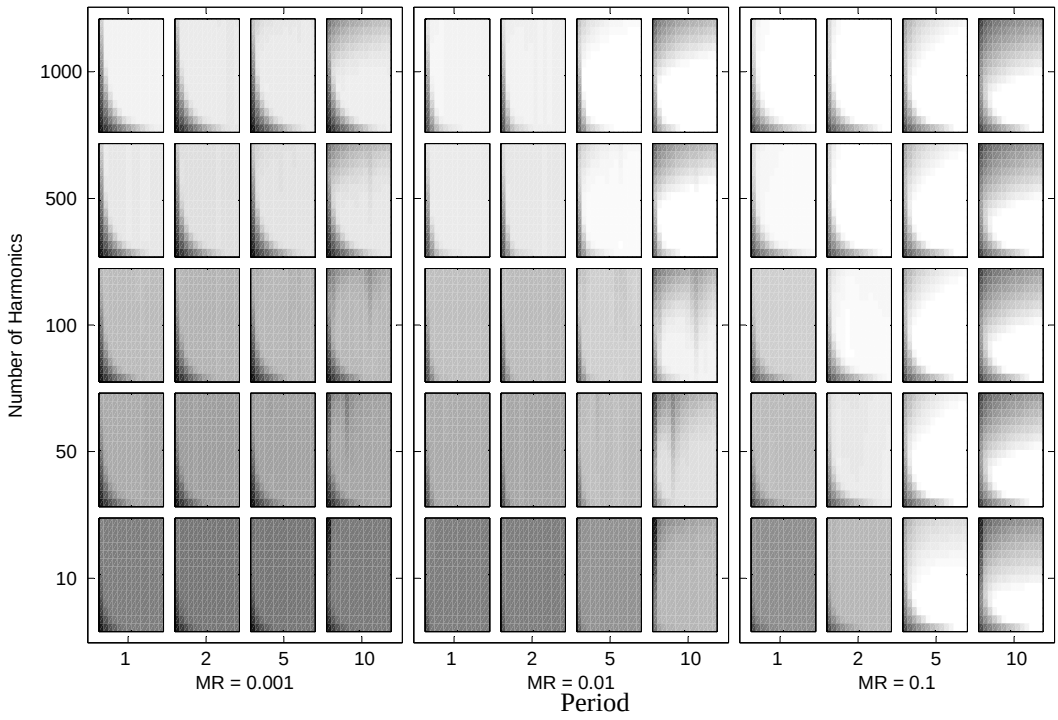

Figure 5.5: Sensitivity analysis of the NILT parameters for the dimensionless LUF model. Three plots of nested surface plots are shown, where each plot corresponds to a value of $\mathbb{MR} = 0.001, 0.01, 0.1$. Each nested surface plot shows the error $e_n(\mathbb{MR}, a, N_\Delta, N_H)$ over the range of N_Δ and a (on the local horizontal and vertical axes) for given values of the period n and N_H (on the outer horizontal and vertical axes). The lighter shade indicates lower values of e_n ($e_n < 0.01$ for white and $e_n > 10$ for black). For the nested surface plots, the value of N_Δ varies linearly from 21 to 301 in the horizontal direction, and the value of a varies linearly from 0.01 to 0.15 in the vertical direction.

numerical integration (this is seen in Figures 5.4 and 5.5 by the large e_n for low values of a and N_Δ). The error behaviour for low a, as observed in Figure 5.8, is seen to result from a bias in the function approximation, which, as in the paragraph above, results from the poor approximation of the integration over the low frequencies components.

The advantage with having a large is that as the integration contour moves further away from the Laplace functions poles, the function along the contour is smoother and easier to integrate with greater accuracy for a reduced number of points. Thus, a large a serves to reduce the bias in the inverse function, and reduce the oscillatory nature of the NILT resulting from dominant sinusoidal terms in the numerical approximation. However, for a too large, due to the e^{at} term in E_T [the truncation error (5.11)], the error increases significantly with time causing the inverse function to diverge dramatically for large times. This is observed from

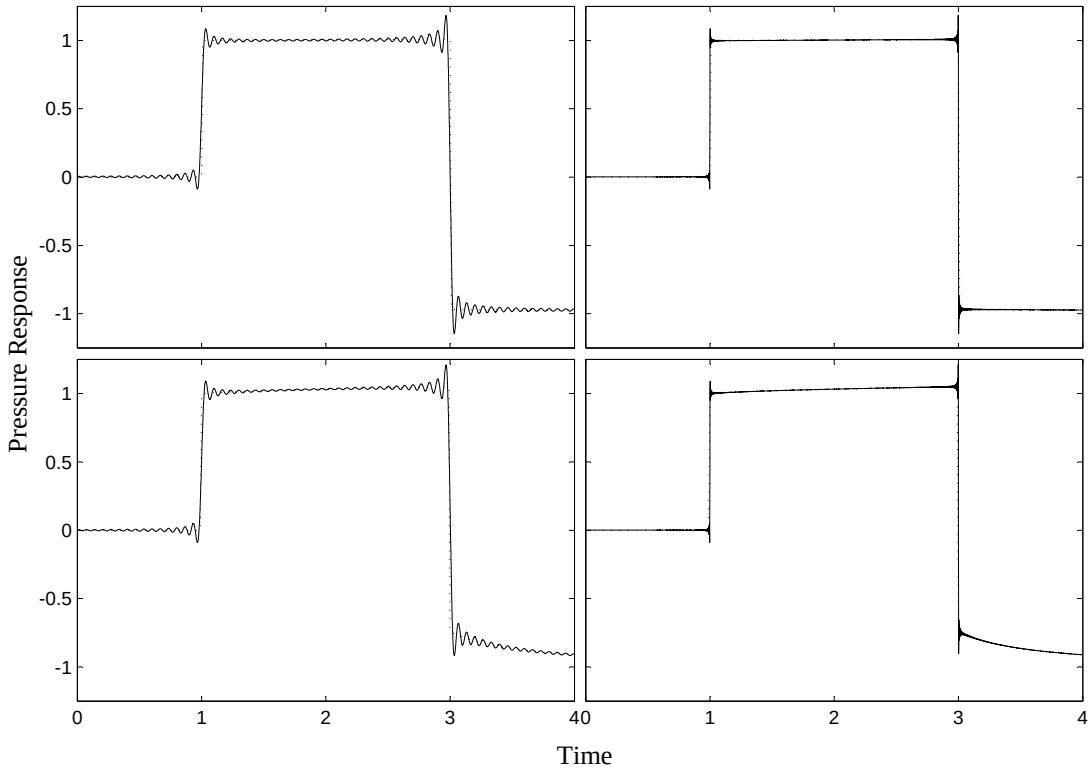

Figure 5.6: Time-domain comparisons of the NILT (−) with the MOC (· · ·) for the linear friction (top row) and Zielkie's friction (bottom row) test functions for harmonic numbers $N_H = 15$ (left column) and $N_H = 250$ (right column). The other parameter values are $\mathbb{M}\mathfrak{R} = 0.01$, $a = 0.07$, $N_\Delta = 41$.

Figures 5.4 and 5.5 by the increasing e_n for large a and time periods n. Examples of the truncation error resulting from high a are seen in Figure 5.8, a high amplitude oscillatory behaviour is observed for the linear TSF, and a large deviation is observed for the LUF.

In summary, the findings of the parametric analysis of the NILT parameters are as follows. For the test functions, reasonable errors were observed for the number of harmonics included in the integration as $N_H \in [500, 1000]$, where for systems with a higher dissipation rate (*i.e.* higher $\mathbb{M}\mathfrak{R}$), lower values were adequate, as the higher frequencies are not as important for these cases. An adequate number of discretisations for each harmonic was observed to be $N_\Delta \in [40, 60]$ where, again, lower values were sufficient for systems with higher dissipation rates. The most appropriate location of the contour was found to be $a \in [0.05, 0.08]$, where lower values performed better for larger t.

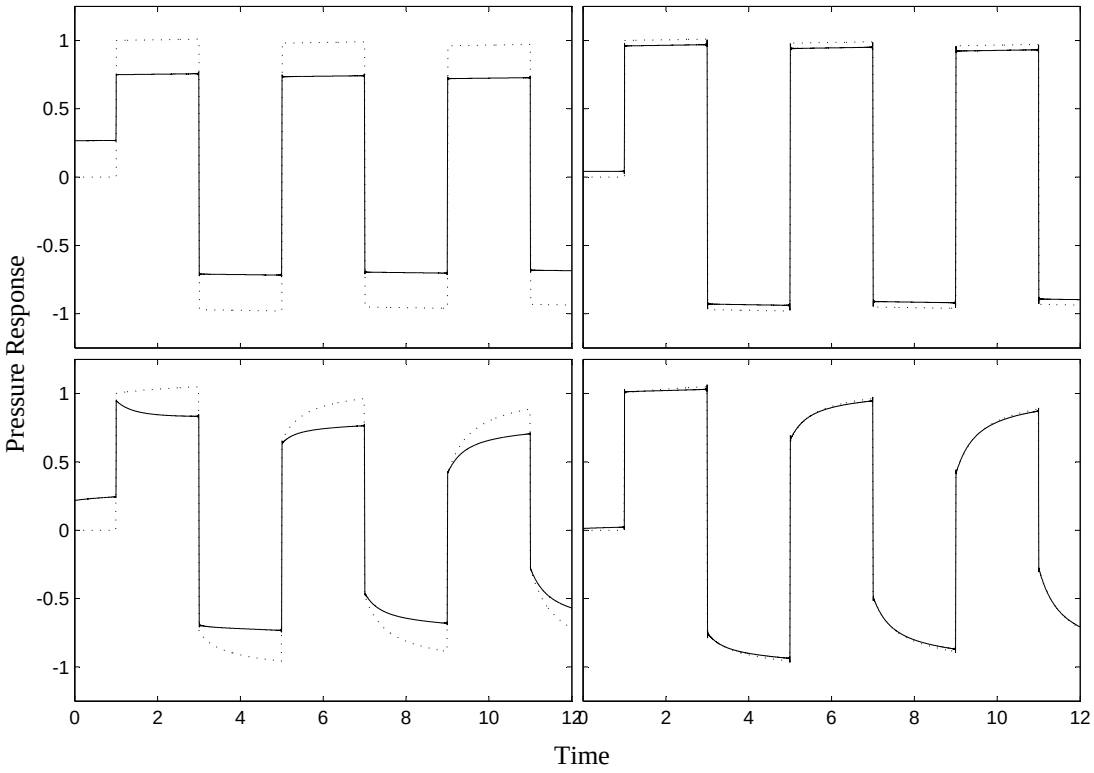

Figure 5.7: Time-domain comparisons of the NILT (−) with the MOC (· · ·) for the linear TSF (top row) and LUF (bottom row) test functions for harmonic discretisations $N_\Delta = 7$ (left column) and $N_\Delta = 21$ (right column). The other parameter values are $\mathbb{M}\mathfrak{R} = 0.01$, $a = 0.07$, $N_H = 500$.

5.6 Application to Networks

The primary interest within this chapter is the suitability of the linear numerical inverse Laplace transform (NILT) approach for the time-domain simulation of pipeline networks comprised of both linear and nonlinear pipes. The important issues pertaining to the suitability of the NILT method are (i) the accuracy of the method to approximate the true dynamics, and (ii) the relative computational efficiency of the method with respect to alternative simulation approaches.

As in Section 5.5, a computational transient hydraulic solver based on the MOC has been used here as both the representation of the true system (for the accuracy comparisons), and the alternative simulation approach (for the computational efficiency comparison). Four network case studies have been considered, including the 11-pipe network in Figure 5.9 from *Pudar and Liggett* [1992], the 35-pipe network in Figure 5.10 from *Pudar and Liggett* [1992], the 51-pipe network in Figure 3.6 from *Vítkovský* [2001], and the 94-pipe network from Figure 3.9 from *Datta and Sridharan* [1994]. Each of the networks have been simulated with the five different pipe types from Chapter 2, these being the linear LSF, LUF and viscoelastic (VE) models, and

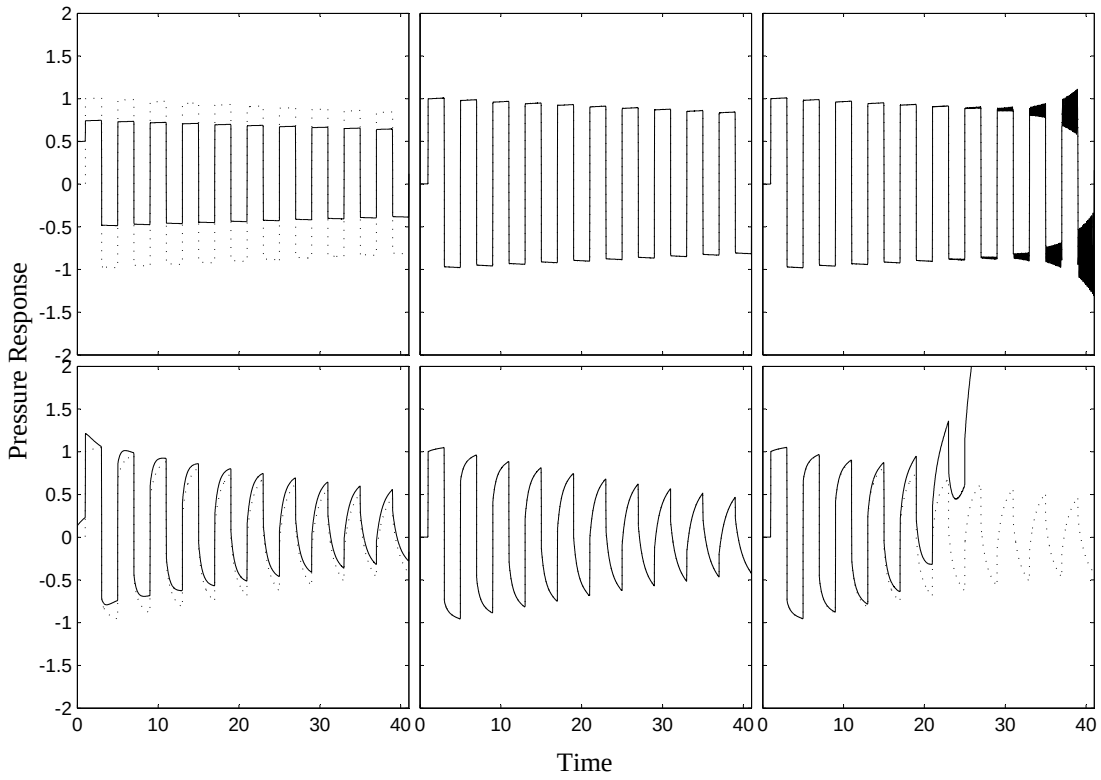

Figure 5.8: Time-domain comparisons of the NILT (−) with the MOC (· · ·) for the linear TSF (top row) and LUF (bottom row) test functions for contour locations $a = 0.001$ (left column), $a = 0.07$ (centre column), and $a = 0.3$ (right column). The other parameter values are $\mathbb{M}\mathfrak{R} = 0.01$, $N_H = 500$, and $N_\Delta = 41$.

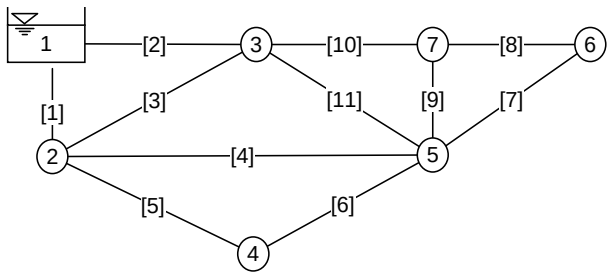

Figure 5.9: The 11-pipe network adapted from *Pudar and Liggett* [1992].

the two nonlinear TSF and turbulent-unsteady-friction (TUF) models. Examples of specific case studies are used to highlight important issues relating to the accuracy (Section 5.6.2) and computational efficiency (Section 5.6.3) of the NILT. Following the examples, general results observed for all case studies are discussed at length. Additional issues concerning the simulation of networks with step changes and the advantages of the discretisation-free simulation of the NILT are covered in Section 5.6.4. All computational procedures were undertaken as outlined in Appendix E.

NOTE:
This figure is included on page 134
of the print copy of the thesis held in
the University of Adelaide Library.

Figure 5.10: The 35-pipe network adapted from *Pudar and Liggett* [1992].

For all simulations, the time step was selected so as to achieve a Courant number of 1 in all computational reaches so as to avoid numerical errors. Section 5.6.4 explores the numerical efficiency/accuracy trade-off in selecting larger time-steps.

5.6.1 Parameter preliminaries

The parametric analysis of the previous section was applied to dimensionless single pipelines, thus necessitating a reinterpretation of the heuristics for the integration contour location a and the integration discretisation $\Delta\omega$ for use in networks. The parameters N_H (the number of harmonics included in the construction of the NILT solution) and N_Δ (the number of discretisations within a single harmonic) required no reinterpretation for the application to networks.

Firstly, the the Laplace variable s for the dimensionless system (5.12), is nondimensionalised by a factor of c/L. Therefore, the actual location of the integration contour a, for the dimensionless pipeline, is in fact $(c/L) \cdot a$ for a dimensional pipeline. For a multi-pipe network $(\mathcal{G}(\mathcal{N}, \Lambda), \mathcal{P})$, preliminary studies found the following heuristic to be suitable

$$a = \tilde{a} \cdot \min_{\lambda \in \Lambda} \left\{ \frac{c_\lambda}{L_\lambda} \right\}, \quad \tilde{a} \in [0.05, 0.08], \tag{5.15}$$

that is, the contour scaling is based on the minimum c/L value for the network.

The second issue of ambiguity in the reinterpretation of the parameter study from Section 5.5 results from the definition of a harmonic width by which $\Delta\omega$ is

computed from N_Δ. As with the contour location, the harmonic width of a dimensional pipeline is $c/L \cdot \pi/2$ where $\pi/2$ is the harmonic width of the dimensionless system. However, the frequency response within networks does not contain uniformly spaced harmonics, as is approximately the case for single pipelines, and the spacing between the harmonics along an integration contour can be highly irregular (*e.g.* see the results for the numerical examples in Chapter 3). To provide guidance to the integration discretisation, it is necessary to define a meaningful harmonic width $\Delta\omega_H$. From this perspective, it is important to adopt a harmonic width that is small enough to adequately discretise the harmonics with the thinnest width in frequency range. Preliminary studies showed that an appropriate heuristic for the network harmonic width was

$$\Delta\omega_H = \frac{\pi}{2} \cdot \min_{\lambda\in\Lambda} \left\{ \frac{c_\lambda}{L_\lambda} \right\} \tag{5.16}$$

which is basically the width between the harmonics for the pipe with the smallest c/L ratio (*i.e.* the pipe with the largest period).

To demonstrate the robustness of the method, the parameters were not calibrated to each case study, but the heuristics were used with $\tilde{a} = 0.07$ and $N_\Delta = 41$, and a range of N_H was used ($N_H \in \{250, 500, 1000, 2000\}$) to study the accuracy versus computational efficiency trade-off .

5.6.2 Accuracy studies

Five detailed examples are firstly presented and discussed below, where each example is based on the 51-pipe network whose pipes are comprised of one of the five pipe types[11]. These examples are used to explore the issues pertaining to each different pipe type. Following these examples, a detailed analysis of the results for all four networks is given, where general conclusions are drawn. The Courant number was maintained at 1 for all computational reaches within each MOC case study so as to avoid errors associated with numerical interpolation schemes.

Example 5.1. *Consider the 51-pipe network (details are given in Appendix D) with all pipes modelled according to the LSF model from Example 2.3. The hydraulic scenario is as defined in Example 4.7, except the network is excited into a transient state by temporarily halting the demand at nodes* $\{12, 17, 27, 30\}$ *for a period of*

[11]It is recognised that in the case of the laminar pipes, the assumption of laminar flow may be violated. This however is not of concern within this research, as the emphasis is on the ability of the NILT to approximate the MOC for a range of different network dynamics. That is, the comparative dynamical behaviour is of primary interest.

$\{1.0, 0.5, 0.3, 0.4\}$ *seconds. The pressure response of the network at node*[12] *25 for the first 100 s, as computed by the MOC on a temporal grid of* $\Delta t = 0.001$ *s, is given in Figure 5.11(a). Figures 5.11(b)-5.11(d) show the error functions for the NILT approximations for* $N_H = 250$, 500, *and* 1000, *where* $E_{N_H}(t) = \tilde{f}(t|a, \Delta\omega, N_H) - f(t)$. *For the other NILT parameters, heuristics from* (5.15)-(5.16) *were used.*

From Example 5.1, a consideration of the $E_{1000}(t)$ error function shows that an extremely accurate simulation for this pipe type with the NILT is achievable. Comparing the error functions $E_{1000}(t)$, $E_{500}(t)$, and $E_{250}(t)$ it is observed that there is an order of magnitude increase in the accuracy when doubling the number of harmonics included in the NILT from 250 to 500, and that the increase from 500 to 1000 harmonics yields a near indistinguishable error. The highest error in the NILT occurred in the first stages of the pressure response, where closer analysis shows that the error was associated with a Gibbs-type oscillation in the NILT approximation resulting from the sharpness of the pressure wave (as illustrated in Figure 5.6 for the test functions). For the larger time scales, the errors remain within reasonably small bounds (*i.e.* $|E_{1000}(t)| < 1$ kPa, $|E_{500}(t)| < 3$ kPa, and $|E_{250}(t)| < 10$ kPa).

Example 5.2. *Consider the 51-pipe network and hydraulic scenario from Example 5.1, but with the pipes modelled with the TSF model from Example 2.4. The pressure response of the network at node 25, as computed by the nonlinear MOC model, is given in Figure 5.11(a), and Figures 5.11(b)-5.11(d) show the error functions for the linear NILT approximations where the NILT parameters values are as in Example 5.1.*

Similarly with Example 5.1, the error functions $E_{250}(t)$, $E_{500}(t)$, and $E_{1000}(t)$ exhibit their maximum error in the early stages of the pressure response due to the sharpness of the pressure front at this time. However, a qualitativly different behaviour of the error functions is observed for this example as opposed to that in Example 5.1. For the smaller time points the errors appear to follow a trend (as opposed to being approximately uniformly distributed about the 0 level) and at the larger time points, the errors do not consist of high frequency oscillations as in Example 5.1, but they have a much lower variability, particularly in the case of the $E_{250}(t)$ function.

These observed differences can be explained by considering the dynamic behaviour of the pipe types. Firstly, the lower magnitude in the high frequency components in the error, observed in this example, result from the higher energy dissipation of the TSF pipes. This means that the energy in the higher frequencies did

[12]This node was randomly selected, and is used throughout the examples for the purpose of comparison.

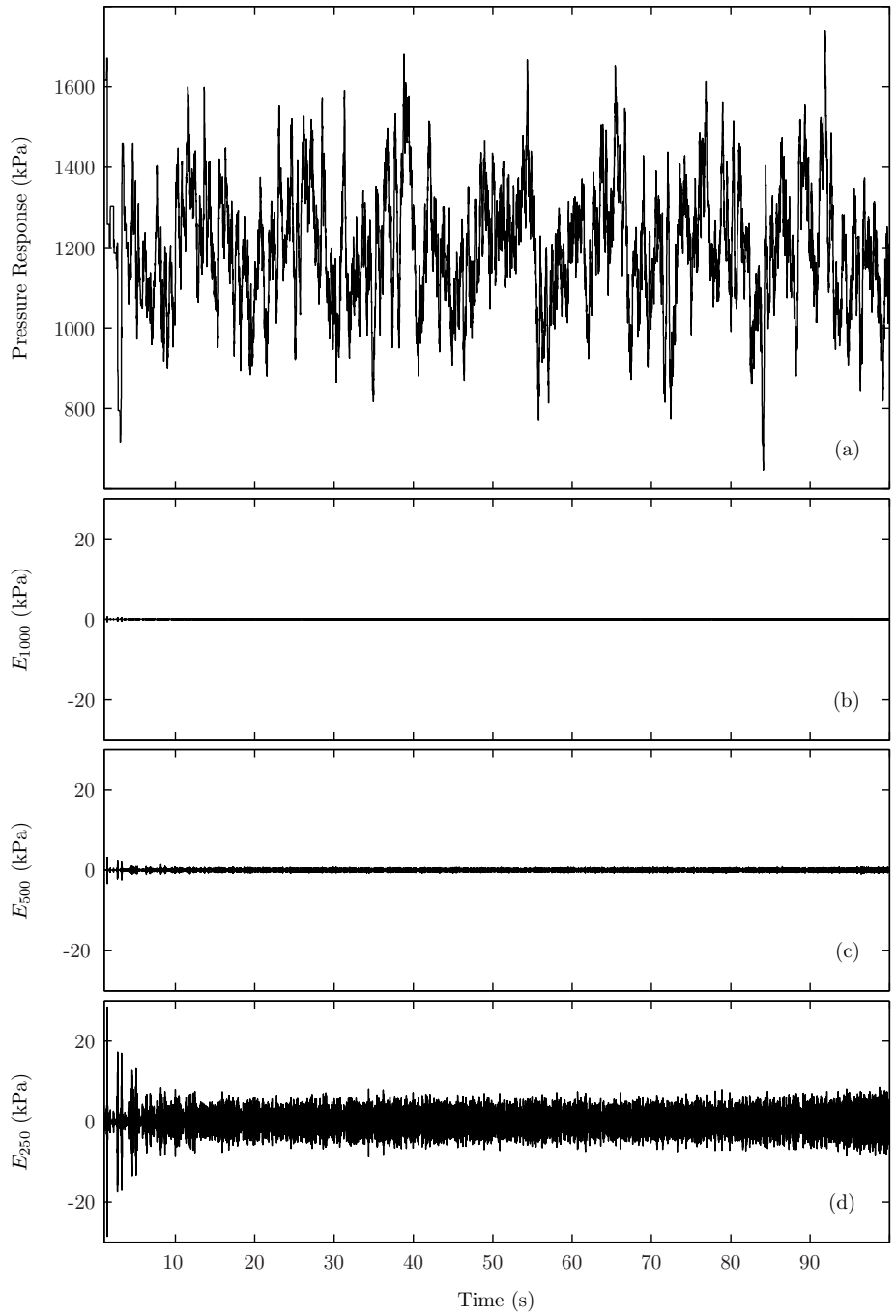

Figure 5.11: Comparison of MOC and the NILT for the laminar-steady-friction (LSF) 51-pipe network from Example 5.1 for the pressure response at node 25: (a) the pressure response $f(t)$ computed by the MOC, (b)-(d) the errors $E_{N_H}(t) = \tilde{f}(t|a, \Delta\omega, N_H) - f(t)$ for $N_H = 1000$, 500, and 250, respectively.

not persist into the longer time-scales, as was observed for the LSF pipes. Secondly, the bias (or trend) observed in the error was due to the fact that the NILT is a linear approximation. The nonlinear error associated with the NILT is dependent on the square of the size of the deviation of the flow from the operating point $|q(t) - q_0|^2$.

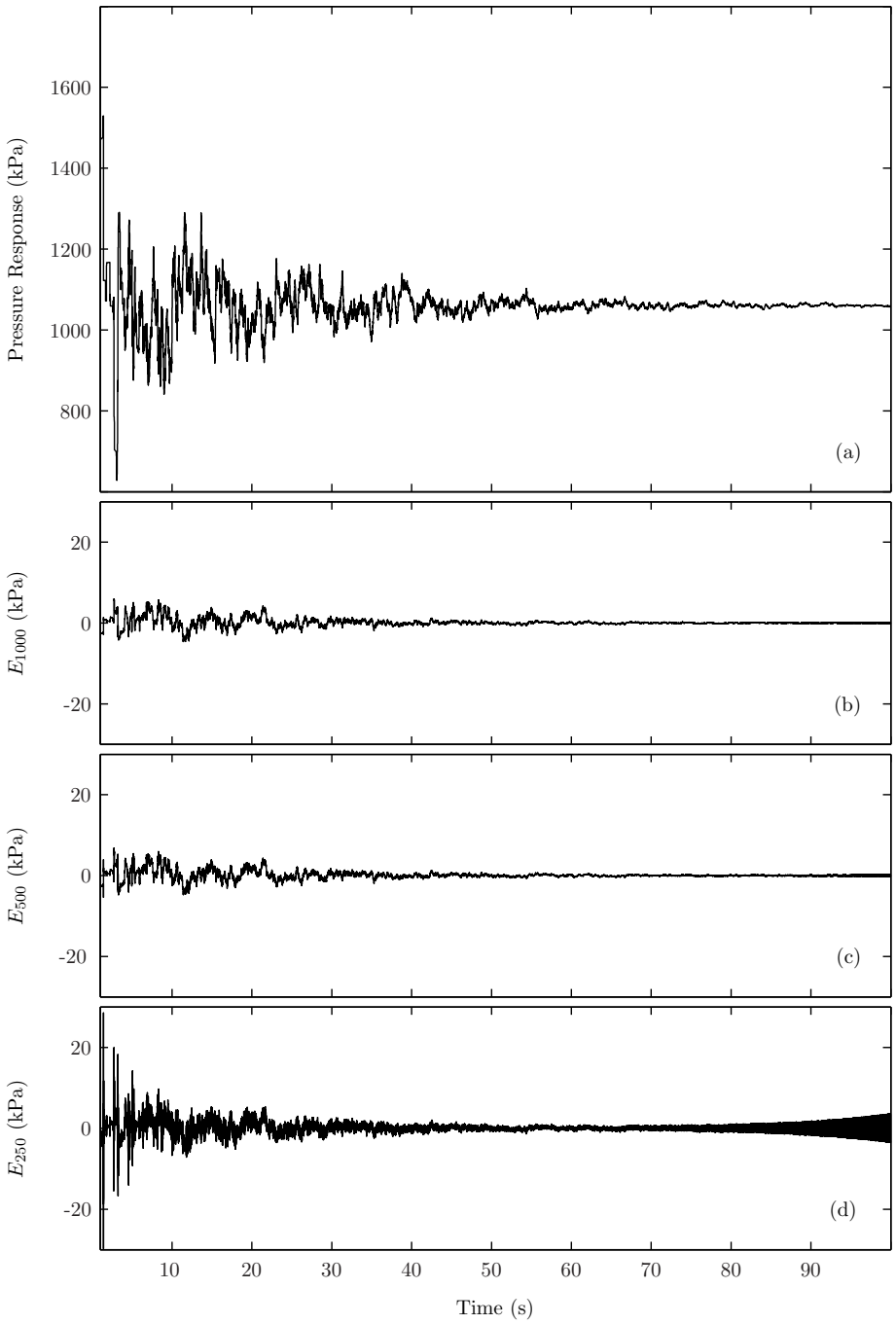

Figure 5.12: Comparison of MOC and the NILT for the turbulent-steady-friction 51-pipe network from Example 5.2 for the pressure response at node 25: (a) the pressure response $f(t)$ computed by the MOC, (b)-(d) the errors $E_{N_H}(t) = \tilde{f}(t|a, \Delta\omega, N_H) - f(t)$ for $N_H = 1000$, 500, and 250, respectively..

Therefore, when the deviation was large enough, the linear model yielded a biased approximation of the nonlinear model. This phenomena is also observed in Example 5.4.

The comparative behaviour of the error functions is also different for this exam-

ple as opposed to Example 5.1. It is seen that a reduction in the error is achieved by doubling the number of harmonics used in the NILT from 250 to 500. However, the reduction in error for $E_{1000}(t)$ in comparison to $E_{500}(t)$ is small. This observation is consistent with the discussion in the previous paragraph, in that the higher dissipation of this network means that the higher frequency components do not contribute much to the signal reconstruction in the NILT, but that the observed error is mainly associated with the nonlinearities. Finally, the error function $E_{250}(t)$ increases in magnitude for the larger time, resulting from the truncation error associated with $E_{250}(t)$.

Example 5.3. *Consider the 51-pipe network and hydraulic scenario from Examples 5.1 and 5.2, but with the pipes modelled with Zielke's LUF model from Example 2.6 truncated to 10 terms according to the Vítkovský et al. [2002] approximation. The pressure response of the network at node 25 for the MOC and the associated errors for the NILT approximations, are given in Figure 5.13 (the NILT parameters values are as in Example 5.1).*

As with Example 5.1, the accuracy that is achievable with the NILT for linear pipe types is demonstrated by the extremely low magnitude of the $E_{1000}(t)$ error function for Example 5.3. As the pipe type for this example is linear, there is no trend in the error functions as with Example 5.1. However, in comparison to Example 5.1, the errors are generally smaller in the larger time scales. This reduction in the error is attributed to the higher dissipation rate in the LUF pipes as opposed to the LSF pipes, resulting in a lower contribution to the time-domain behaviour from the higher frequencies.

Example 5.4. *Consider the 51-pipe network and hydraulic scenario from Examples 5.1-5.3, but with the pipes modelled with the Vardy-Brown TUF model from Example 2.7 truncated to 13 terms [Vardy and Brown, 2007]. The pressure response of the network at node 25 for the nonlinear MOC and the associated errors for the linear NILT approximations, are given in Figure 5.14 (the NILT parameters values are as in Example 5.1).*

As the pipe type for Example 5.4 is nonlinear, the behaviour of the error functions is qualitatively similar to that for Example 5.2, but with even less high frequency components in the larger time scales resulting from the increased dissipation rate from the unsteady friction contribution. As with this example, the decrease in the error for increasing the number of harmonics from 500 to 1000 is small as the error arises from the nonlinear dynamics in the MOC model

Example 5.5. *Consider the 51-pipe network and hydraulic scenario from Examples 5.1-5.4, but with the pipes modelled with the Kelvin-Voigt VE model from Example*

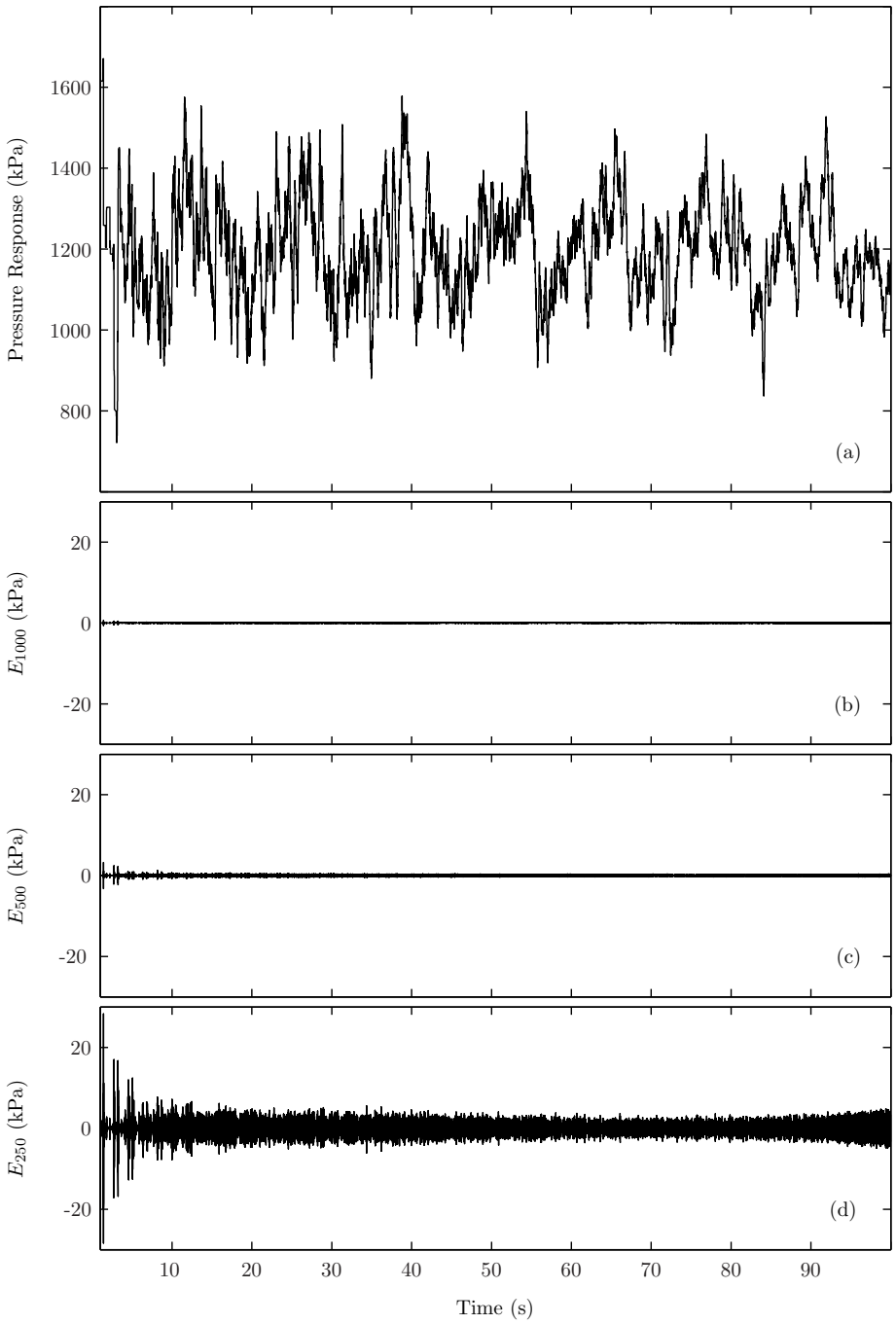

Figure 5.13: Comparison of MOC and the NILT for the laminar-unsteady-friction (LUF) 51-pipe network from Example 5.3 for the pressure response at node 25: (a) the pressure response $f(t)$ computed by the MOC, (b)-(d) the errors $E_{N_H}(t) = \tilde{f}(t|a, \Delta\omega, N_H) - f(t)$ for $N_H = 1000$, 500, and 250, respectively.

2.5 using the single Kelvin-Voigt element model for steel-cement-mortar lined pipes [Stephens, 2008]. The pressure response of the network at node 25 for the MOC and the associated errors for the NILT approximations, are given in Figure 5.14 (the NILT parameters values are as in Example 5.1).

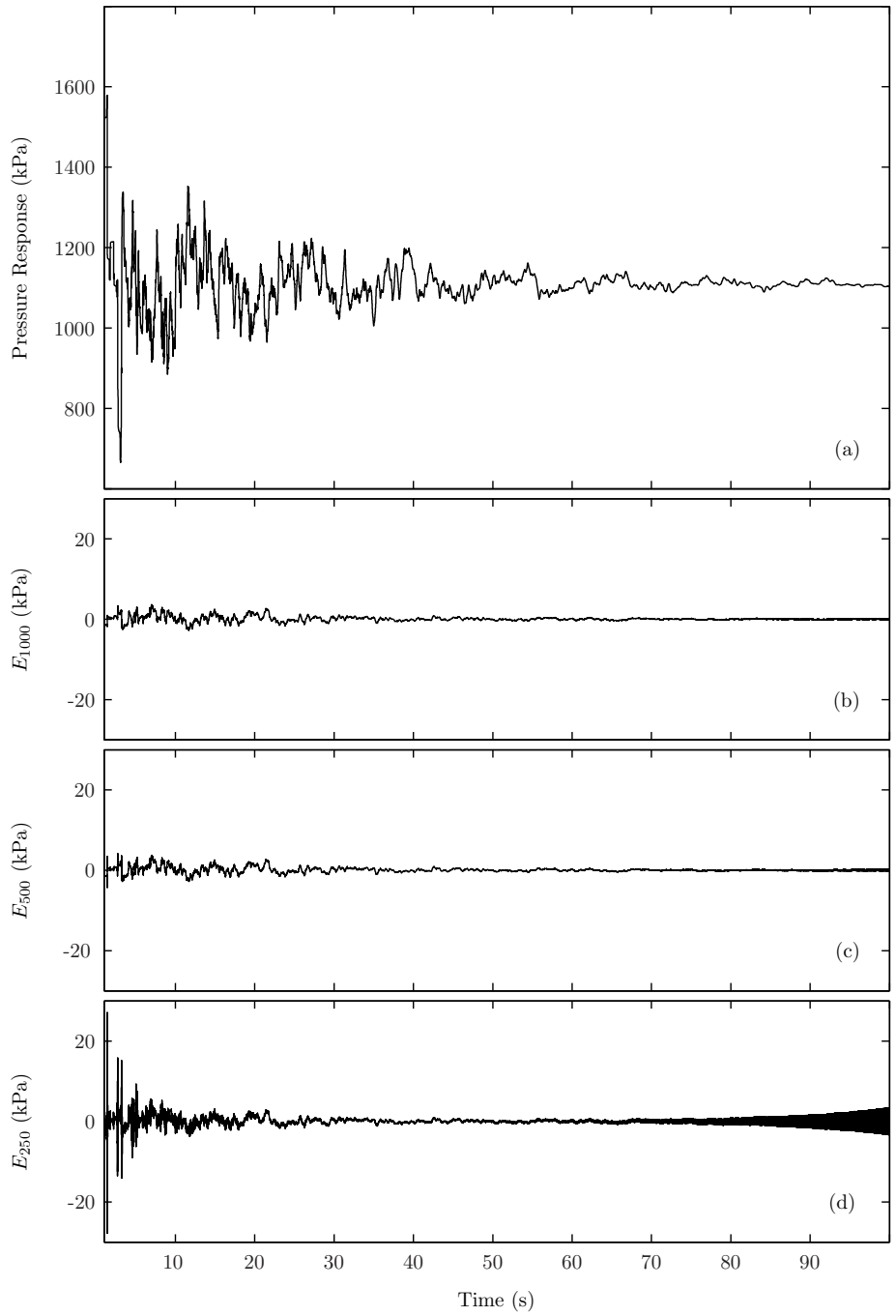

Figure 5.14: Comparison of MOC and the NILT for the turbulent-unsteady-friction 51-pipe network from Example 5.4 for the pressure response at node 25: (a) the pressure response $f(t)$ computed by the MOC, (b)-(d) the errors $E_{N_H}(t) = \tilde{f}(t|a, \Delta\omega, N_H) - f(t)$ for $N_H = 1000$, 500, and 250, respectively.

As with the Examples 5.1 and 5.3, the ability for the NILT to accurately model linear pipes is observed by the low magnitude of both the E_{1000} and E_{500} error functions. It is observed that the errors for this pipe type are the lowest of all pipe types. This is due to the higher dissipation in the VE model. As observed in

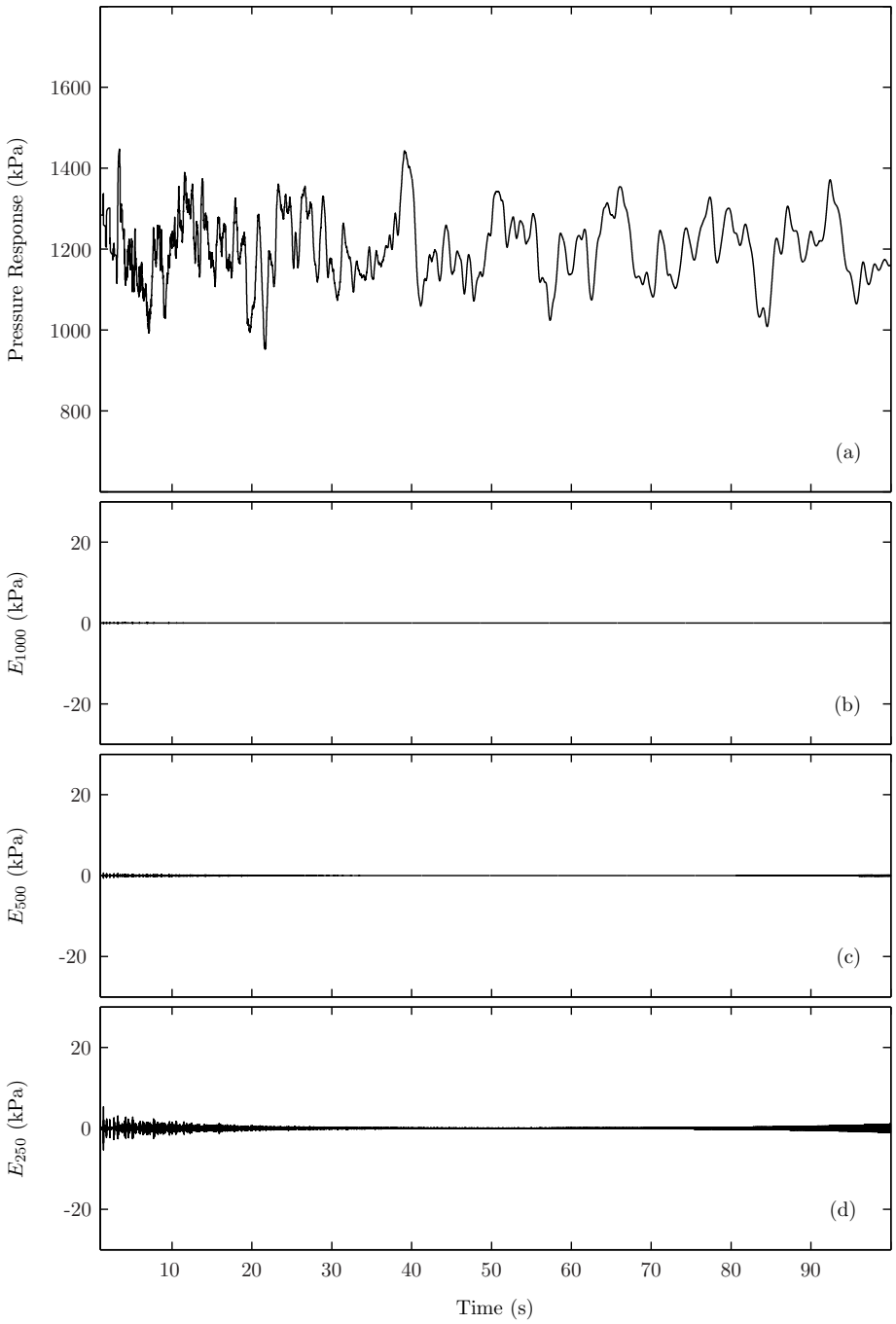

Figure 5.15: Comparison of MOC and the NILT for the viscoelastic (VE) 51-pipe network from Example 5.5 for the pressure response at node 25: (a) the pressure response $f(t)$ computed by the MOC, (b)-(d) the errors $E_{N_H}(t) = \tilde{f}(t|a, \Delta\omega, N_H) - f(t)$ for $N_H = 1000$, 500, and 250, respectively..

Section 5.5, a more dissipative model manifests itself as a smoother function in the Laplace-domain that decays faster for larger $|s|$. Such functions are more accurately numerically integrated.

Accuracy comparisons for 65 case studies (three to four different values of N_H

Table 5.1: The relative L_∞ norms for the MOC and the L_∞ errors for the NILT methods for varying N_H applied to the 11-pipe case studies for five different pipe types. The NILT errors as a percentage of the MOC norms are given in italic

Method[a]	L_∞ norms[b] and errors[c] for pipe types (kPa)				
	LSF	TSF	LUF	TUF	VE
MOC	665.91	395.66	429.47	355.16	379.65
1000	1.07 *(0.2%)*	3.93 *(1%)*	1.07 *(0.3%)*	2.99 *(0.8%)*	1.07 *(0.3%)*
500	6.15 *(0.9%)*	6.93 *(1.8%)*	6.14 *(1.4%)*	6.95 *(2%)*	6.11 *(1.6%)*
250	39.94 *(6%)*	40.47 *(10.2%)*	39.95 *(9.3%)*	40.45 *(11.4%)*	39.95 *(10.5%)*

[a] The numbers refer to the NILT simulations where the number specified corresponds to the value of N_H. [c] The norm values for the MOC are taken relative to the steady state value. [b] The L_∞ errors are based on the maximum of the norms from 5 randomly selected nodes (*i.e.* $\{2, 3, 4, 5, 6\}$).

for five network types in five different pipe types) are summarised in Tables 5.1-5.4. The results are presented in terms of a the L_∞ error of the pressure response between the NILT approximation and the MOC (defined as $||E_{N_H}(t)||_\infty$), where the error is taken as the maximum over 5 nodes for the 11-pipe network and 10 nodes for the other networks. To provide a comparative assessment of the magnitude of the errors relative to the excitation of the system, the L_∞ norm of the MOC with respect to the steady-state value[13] is also given. The use of this norm allows for a comparison of the maximum magnitude of the approximations error as a ratio of the maximum magnitude of transient fluctuations about the steady-state point. The errors as a percentage of the norms are given in Tables 5.1-5.4 in italics.

From Tables 5.1-5.4 it is observed that for the highest values of N_H (the number of harmonics included in the calculation of the NILT) the normalised error was less than 4% for all case studies, with some of the case studies achieving errors of less than 1% for the 11-pipe and 51-pipe networks. This level of accuracy for the nonlinear case studies is greater than expected, particularly given the relatively large transient perturbation from the steady-state operating point, as indicated by the relative L_∞ norm of the MOC. The errors for the 35-pipe and 94-pipe networks were consistently greater than the other two networks, and were also more sensitive to reductions in N_H, as the errors increased markedly more for the reduction from $N_H = 1000$ to $N_H = 250$. Values of $N_H = 500$ are adequate for high accuracy for

[13] That is, given the function $f(t)$ as computed by the MOC, the relative L_∞ norm is defined as $||f(t) - f_o||_\infty$ where f_o is the steady state value of $f(t)$.

Table 5.2: The relative L_∞ norms for the MOC and the L_∞ errors for the NILT methods for varying N_H applied to the 35-pipe case studies for five different pipe types. The NILT errors as a percentage of the MOC norms are given in italic

Method[a]	L_∞ norms[b] and errors[c] for pipe types (kPa)				
	LSF	TSF	LUF	TUF	VE
MOC	628.22	508.69	589.28	508.55	507.00
1000	15.06 *(2.4%)*	16.04 *(3.2%)*	14.92 *(2.5%)*	13.98 *(2.7%)*	13.87 *(2.7%)*
500	71.84 *(11.4%)*	71.82 *(14.1%)*	71.84 *(12.2%)*	71.80 *(14.1%)*	71.88 *(14.2%)*
250	134.65 *(21.4%)*	134.81 *(26.5%)*	134.65 *(22.8%)*	134.74 *(26.5%)*	134.00 *(26.4%)*

[a] The numbers refer to the NILT simulations where the number specified corresponds to the value of N_H. [c] The norm values for the MOC are taken relative to the steady state value. [b] The L_∞ errors are based on the maximum of the norms from 10 randomly selected nodes (*i.e.* $\{3, 5, 7, 8, 10, 11, 12, 13, 15, 19\}$).

Table 5.3: The relative L_∞ norms for the MOC and the L_∞ errors for the NILT methods for varying N_H applied to the 51-pipe case studies for five different pipe types. The NILT errors as a percentage of the MOC norms are given in italic

Method[a]	L_∞ norms[b] and errors[c] for pipe types (kPa)				
	LSF	TSF	LUF	TUF	VE
MOC	1208.89	948.20	1071.98	986.33	1062.28
1000	0.81 *(0.1%)*	33.60 *(3.5%)*	0.80 *(0.1%)*	16.47 *(1.7%)*	0.77 *(0.1%)*
500	6.81 *(0.6%)*	34.51 *(3.6%)*	6.75 *(0.6%)*	17.36 *(1.8%)*	6.56 *(0.6%)*
250	56.23 *(4.7%)*	60.51 *(6.4%)*	55.71 *(5.2%)*	54.23 *(5.5%)*	54.05 *(5.1%)*

[a] The numbers refer to the NILT simulations where the number specified corresponds to the value of N_H. [c] The norm values for the MOC are taken relative to the steady state value. [b] The L_∞ errors are based on the maximum of the norms from 10 randomly selected nodes (*i.e.* $\{3, 6, 8, 10, 15, 19, 22, 25, 28, 32\}$).

the 11-pipe and 51-pipe networks, but $N_H = 1000$ was needed for high accuracy in the 35-pipe network, and $N_H = 2000$ for the 94-pipe network[14].

[14]The increase to $N_H = 2000$ for the 94-pipe would indicate that N_H is dependent on the network size. However, no conclusive comment can be made here as $N_H = 500$ was adequate for the 51-pipe network, but $N_H = 1000$ was needed for the smaller 35-pipe network, this indicating that the necessary N_H is not just dependent on the network size.

Table 5.4: The relative L_∞ norms for the MOC and the L_∞ errors for the NILT methods for varying N_H applied to the 94-pipe case studies for five different pipe types. The NILT errors as a percentage of the MOC norms are given in italic

Method[a]	L_∞ norms[b] and errors[c] for pipe types (kPa)				
	LSF	TSF	LUF	TUF	VE
MOC	1363.40	937.44	1094.56	1642.57	1072.90
2000	50.32 *(3.7%)*	69.48 *(7.4%)*	44.50 *(4.1%)*	53.72 *(3.3%)*	40.89 *(3.8%)*
1000	154.22 *(11.3%)*	142.48 *(15.2%)*	137.86 *(12.6%)*	71.96 *(4.4%)*	126.70 *(11.8%)*
500	236.14 *(17.3%)*	204.71 *(21.8%)*	206.94 *(18.9%)*	126.22 *(7.7%)*	189.58 *(17.7%)*
250	293.31 *(21.5%)*	245.51 *(26.2%)*	260.39 *(23.8%)*	182.94 *(11.1%)*	240.13 *(22.4%)*

[a] The numbers refer to the NILT simulations where the number specified corresponds to the value of N_H. [c] The norm values for the MOC are taken relative to the steady state value. [b] The L_∞ errors are based on the maximum of the norms from 10 randomly selected nodes (*i.e.* $\{9, 16, 36, 39, 48, 56, 57, 62, 69, 71\}$).

With respect to the pipe types, as observed in the examples, the greatest errors occurred for the cases of the nonlinear TSF and TUF types. The TUF approximation was generally more accurate than the TSF. This observation is explained by the fact that the unsteady friction operator in the TUF model is in fact linear [*Vardy and Brown*, 2007], therefore, in comparison to the TSF model, a proportionally greater degree of the dissipation behaviour for the TUF is linear in nature.

As observed in the examples, the LSF generally yielded the highest error for the linear pipe types, in an absolute sense, but in a relative sense the errors as a percentage of the MOC norm were similar to those for the LUF and the VE, as the MOC norm for the LSF cases was typically higher than these other pipe types. The VE cases typically yielded the lowest error, once again being attributed to the higher energy dissipation rate of these pipe types (as is observed by the relatively smoother pressure response in Figure 5.15).

5.6.3 Computational efficiency studies

Differences in computational approaches between the NILT and the MOC

One of the advantages of the NILT as an efficient hydraulic simulator is that it does not require the computation of the complete network state as it deals only with the

composition of transfer functions from the input boundary conditions to the output nodal response variables. This is in contrast to all discrete methods that require a discretisation of the state and the computation of all lumped state variables at each time step. Despite its efficiency, an implication of these different approaches, however, is that the computational time of the NILT is dependent on the number of measurement points[15] N_m within the network, whereas the computational time of discrete methods, like the MOC, is not.

To explain this further, with respect to (5.5), the computation of $\psi_i(t)$ ($N_m = 1$), the pressure response at note $i \in \mathcal{N}_D$, is efficiently calculated by

$$\psi_i(t) = \mathcal{L}^{-1}\left\{\Psi_i(s)\right\}(t) = \mathcal{L}^{-1}\left\{\boldsymbol{H}_i(s)\begin{bmatrix}\boldsymbol{U}(s)\\ \hdashline \boldsymbol{\Theta}_d(s)\\ \hdashline \boldsymbol{\Psi}_r(s)\end{bmatrix}\right\}(t), \tag{5.17}$$

where $\boldsymbol{H}_i$ is the i-th row of the system matrix $\boldsymbol{H}$. That is, (5.17) involves only a vector multiplication (at each s) and the ILT of only a single Laplace-domain function $\Psi_i(s)$. This can be performed without evaluating any of the other response nodal states in $\boldsymbol{\psi}_D(t)$ or $\boldsymbol{\theta}_r(t)$. Similarly, to calculate the pressure response at any two nodes $\psi_i(t)$ and $\psi_j(t)$ ($N_m = 2$), the most efficient approach is

$$\begin{bmatrix}\psi_i(t)\\ \psi_j(t)\end{bmatrix} = \mathcal{L}^{-1}\left\{\begin{bmatrix}\Psi_i(s)\\ \Psi_j(s)\end{bmatrix}\right\}(t) = \mathcal{L}^{-1}\left\{\begin{bmatrix}\boldsymbol{H}_i(s)\\ \boldsymbol{H}_j(s)\end{bmatrix}\begin{bmatrix}\boldsymbol{U}(s)\\ \hdashline \boldsymbol{\Theta}_d(s)\\ \hdashline \boldsymbol{\Psi}_r(s)\end{bmatrix}\right\}(t), \tag{5.18}$$

which requires approximately double[16] the operations of (5.17) (two vector multiplications, and the ILT of two functions). Given this dependency on the number of state variable points of interest, it is important to include this as a parameter in the numerical computational studies.

Numerical experiments

As with Section 5.6.2, firstly a couple of specific examples are given, followed by a general analysis of the computational timings for all case studies from Section 5.6.2. Simulations were performed on a 2.13 GHz Linux machine were the CPU timings were evaluated by the `procstat` routine to ensure that the exact processing time of the simulation in the CPU was recorded.

[15] That is, spatial points of interest at which the transient response is to be computed.

[16] The computational requirements are only approximately double, as there is some computational saving in computing the NILT for two functions at a time (as opposed to two functions independently).

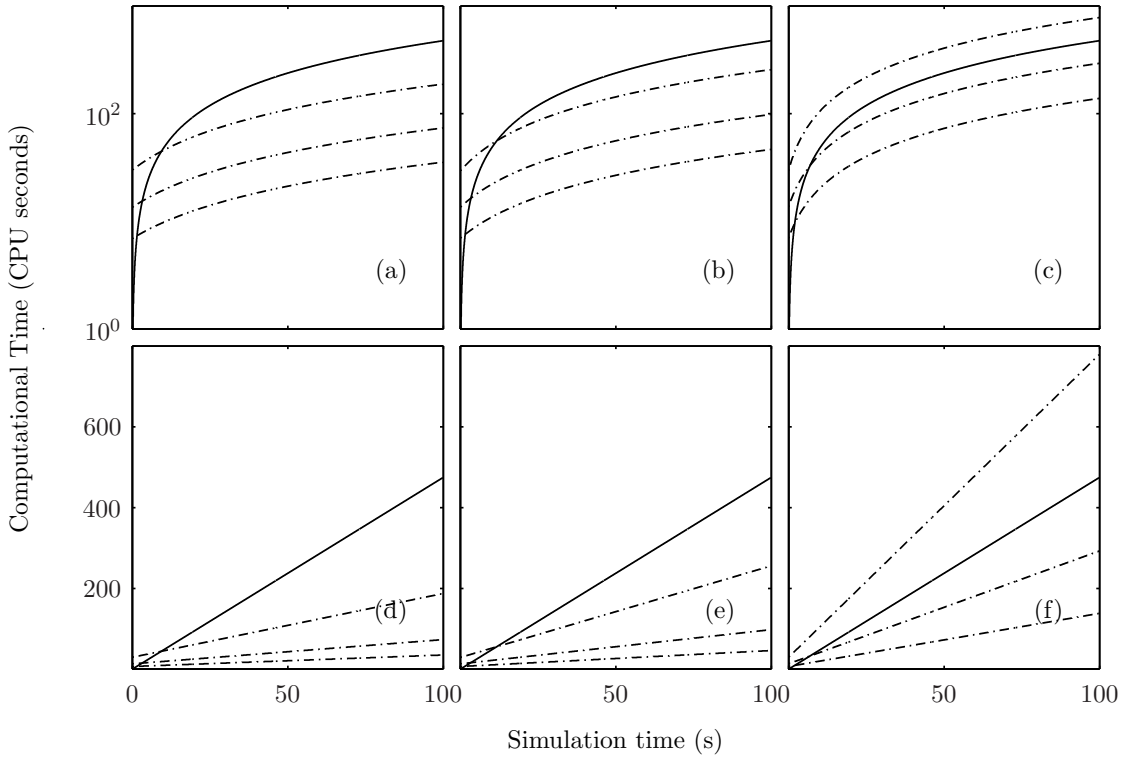

Figure 5.16: Computational times versus simulation time for the TSF 51-pipe network from Example 5.6 for the MOC (−) and the NILT (− · −). The three lines for the NILT correspond to N_H=250 (lowest), N_H=500 (middle), and N_H=1000 (highest). Figures (a)-(c) show the computational time on log scale for the case of the pressure response being computed at 1, 2, and 10 nodes, respectively. Figures (d)-(f) show the computational time in linear scale for the case of the pressure response being computed at 1, 2, and 10 nodes, respectively. Computational times are in CPU seconds. Simulations were performed on a 2.13 GHz Linux machine.

Example 5.6. *Consider the TSF 51-pipe network simulation in Example 5.2 consisting of* 10^5 *time point computations (*i.e. *100 s simulation time at a temporal discretisation of* $\Delta t = 0.001$ *s). The computational time for this simulation (in CPU seconds) for the MOC and the NILT are given in Figure 5.16 for* $N_m = 1$*, 2, and 10 nodal response variables.*

For this example, considering the computational requirements for a given number of nodal calculations in Figure 5.16, qualitatively, the computational time of the MOC and the NILT demonstrate an interesting behaviour. The MOC required minimal startup time, where the initialisation procedures simply involve setting the state variables to the initial steady-state values[17]. The increase in computational time for the MOC is linear with the simulation time t. The NILT however has a more computationally expensive overhead in the initialisation procedures, as observed

[17]The computation of the initial steady-state hydraulic solver is not included in these comparative timing studies as it is was the same for both the MOC and the NILT.

more clearly in the Figure 5.16(a)-(c) log scale plots. This is attributed to the fact that before any time-points can be computed, the complex coefficients $F(a + in\Delta\omega), n = 0, 1, \ldots, N_\Delta \cdot N_H$ must be determined by (5.5), which for this example involved the inversion of a complex 32×32 matrix. Therefore, as observed in Figure 5.16, the initial computational overhead is greater as N_H is increased. Similarly to the MOC, the computational time for the NILT increased linearly with t. As is clear on the Figure 5.16(d)-(f) linear plots, the startup time for the NILT is only a small portion of the overall computational time.

Doubling the number of measurement nodes from 1 to 2 [Figures 5.16(a) and 5.16(d) compared to Figures 5.16(b) and 5.16(e)] appear to have small impact on the increase in computational time (particularly for the smaller N_H). For these cases, the NILT is more computationally efficient for larger t. However, as the number of measurement nodes is increased to 10, the computational requirements of the NILT for all N_H are dramatically increased, such that, as seen in 5.16(e), the MOC is more efficient than the NILT with $N_H = 1000$ for all t.

Example 5.7. *Consider the TUF 51-pipe network simulation in Example 5.4 consisting of* 10^5 *time point computations (*i.e. *100 s simulation time at a temporal discretisation of* $\Delta t = 0.001$ *s). The computational time for this simulation (in CPU seconds) for the MOC and the NILT are given in Figure 5.16 for 1, 2, and 10 nodal response variables.*

For this example, the behaviour of the computational time as a function of the simulation time t for the TUF from Figure 5.17 is qualitatively similar to that for the TSF in Figure 5.16, with the significant quantitative difference being the computational times for the MOC. As a time-domain operator, the TUF involves the evaluation of a convolution to model the unsteady component to the fluid shear stresses at every spatial point. Under the efficient *Vardy and Brown* [2007] algorithm, this convolution is transformed into a single step difference equation in a finite number of states. Therefore, in comparison to the TSF, the TUF involves the storage of these additional states and the calculation of the difference equation at each time point for each spatial point. The computational impact of this is observed to be a near quadrupling of the computational cost of the TUF in comparison with the TSF.

In contrast, the Laplace-domain representation of the TUF does not require any additional states, but the *Vardy and Brown* [2007] algorithm serves to introduce a rational function[18] into the resistance transfer function. Therefore, in comparison with the TSF, the computational difference in the NILT method occurs only in the

[18]The introduced rational function is of the order of the number of introduced states.

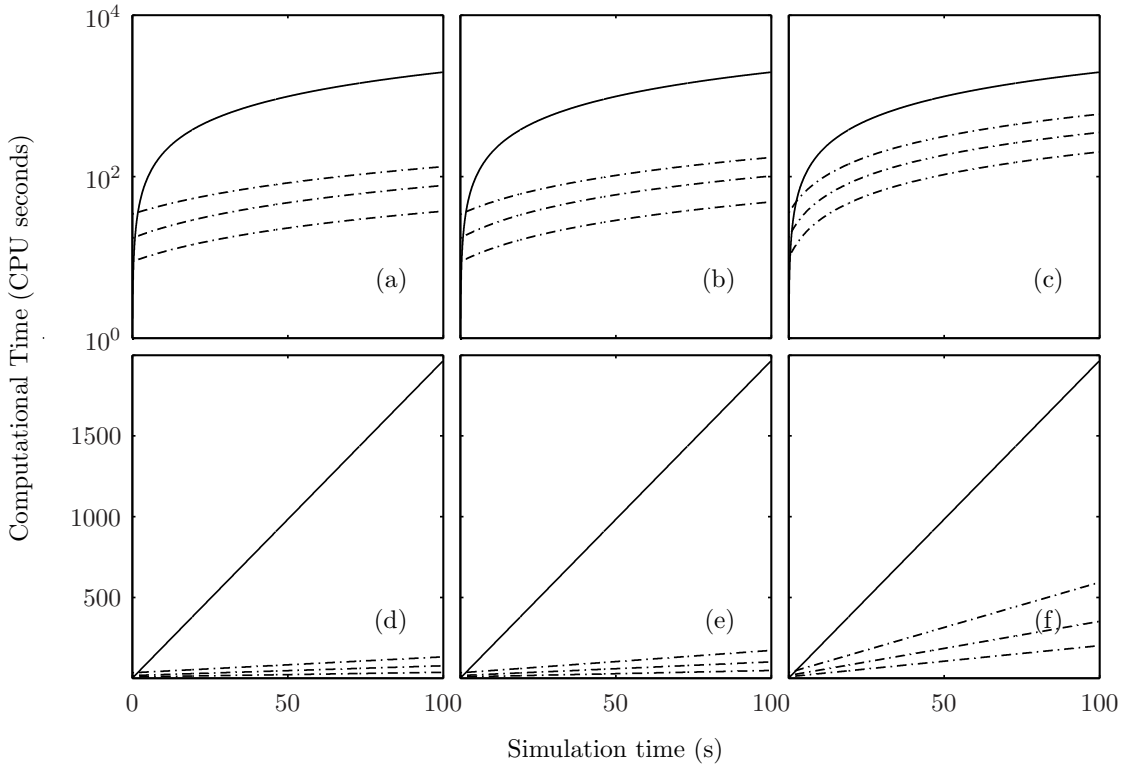

Figure 5.17: Computational times versus simulation time for the 51-pipe network from Example 5.6 with TUF pipes for the MOC(−) and the NILT (− · −). The three lines for the NILT correspond to N_H=250 (lowest), 500 (middle), and 1000 (highest). Figures (a)-(c) show the computational time on log scale for the case of the pressure response being computed at 1, 2, and 10 nodes, respectively. Figures (d)-(f) show the computational time in linear scale for the case of the pressure response being computed at 1, 2, and 10 nodes, respectively. Computational times are in CPU seconds. Simulations were performed on a 2.13GHz Linux machine.

initialisation time, that is, only when the $\boldsymbol{H}(s)$ matrix in (5.5) is computed. From Figure 5.17, it is clear that this cost is small. Consequently, once the complex coefficients $F(a + \mathrm{i}n\Delta\omega), n = 0, 1, \ldots, N_\Delta \cdot N_H$ have been computed, the computational cost of the NILT at each time point for the TUF is the same as for the TSF. This results in the NILT being more computationally efficient than the MOC for all cases as depicted in Figure 5.17.

To generalise the study, numerical timing experiments were performed on the four different networks from Section 5.6.2 in the five different pipe types, creating a total of 20 different network types. For the NILT, the timing experiments were performed for $N_H \in \{250, 500, 1000\}$ for the 11-pipe, 35-pipe and 51-pipe network and $N_H \in \{250, 500, 1000, 2000\}$ for the 94-pipe network, with $N_m \in \{1, 2, 5\}$ for the 11-pipe network and $N_m \in \{1, 2, 5, 10\}$ for the other three networks. Consequently, the study comprised 20 MOC simulations, and 245 NILT simulations, each at 10^5 time points.

Table 5.5: Computational times[a] of the NILT relative to the MOC for the 11-pipe case study. The computational times for the MOC simulations (in CPU seconds) are given in italics. All times are based on the computation of 10^5 simulation time points.

N_H	N_m	Relative computational times for pipe type				
		LSF	TSF	LUF	TUF	VE
1000	5	2.770[b]	2.793	0.938	0.700	1.354
	2	1.514	1.535	0.513	0.381	0.746
	1	1.105	1.094	0.370	0.278	0.529
500	5	1.398	1.392	0.465	0.350	0.664
	2	0.767	0.751	0.256	0.190	0.368
	1	0.549	0.549	0.183	0.140	0.269
250	5	0.690	0.700	0.232	0.175	0.338
	2	0.385	0.378	0.129	0.096	0.187
	1	0.279	0.272	0.092	0.069	0.134
(MOC		*74.1*	*75*	*223.5*	*297.4*	*154.9)*

[a]Simulations were performed on a 2.13GHz Linux machine were the CPU timings were evaluated by the procstat routine. [b] This means that the NILT took 2.770 times the computational time for the MOC, which in this case is $2.77 \times 74.1 = 205.3$ CPU seconds.

Tables 5.5-5.8 summarise the numerical experiments for the computational timing studies. Presented in italics are the computational times (CPU seconds) of the MOC for the different network types and, for convenience, the computational times of the NILT are presented as a ratio with the corresponding MOC time (*i.e.* relative computational times greater than 1 indicate that the MOC was more efficient than the NILT for the particular case).

At a first observation, Tables 5.5-5.8 show the expected result that the computational time of the NILT are approximately proportional to the number of harmonics N_H involved in the inversion process (*i.e.* a doubling of N_H is matched by a doubling of the computational time). In comparison, the computational cost of the NILT is not linear with the number of measurement nodes N_m, but the incorporation of each additional node for $N_m > 1$ costs approximately an additional 1/3 of the computational time required for the first node (*i.e.* at each time point, there are operations that need to be performed only once for all nodes).

As is clear in Tables 5.6-5.8, the NILT compares more favorably with the MOC for the larger networks, with relative computational times reaching as low as 0.008 (*i.e.* two orders of magnitude less time then the MOC). This is attributed to the fact that the MOC has an increasing computational expense for larger networks. In contrast, the only computational overhead associated with large networks for the

Table 5.6: Computational times[a] of the NILT relative to the MOC for the 35-pipe case study. The computational times for the MOC simulations (in CPU seconds) are given in italics. All times are based on the computation of 10^5 simulation time points.

N_H	N_m	Relative computational times for pipe type				
		LSF	TSF	LUF	TUF	VE
1000	10	0.775	1.132	0.347	0.206	0.415
	5	0.539	0.587	0.122	0.144	0.242
	2	0.262	0.310	0.068	0.076	0.134
	1	0.192	0.209	0.074	0.049	0.099
500	10	0.522	0.404	0.116	0.123	0.286
	5	0.337	0.230	0.073	0.065	0.161
	2	0.181	0.127	0.041	0.027	0.083
	1	0.129	0.092	0.030	0.020	0.070
250	10	0.141	0.141	0.045	0.032	0.073
	5	0.155	0.081	0.026	0.019	0.042
	2	0.077	0.045	0.014	0.010	0.024
	1	0.066	0.033	0.011	0.008	0.018
(MOC		*649.7*	*655.4*	*2106.7*	*2916.8*	*1280.7)*

[a]Simulations were performed on a 2.13GHz Linux machine were the CPU timings were evaluated by the `procstat` routine.

NILT is in the initialisation phase, which, from Examples 5.6-5.7, is observed to only contribute minimally to the overall computational time. The MOC was only faster than the NILT for the cases of the numerically simple LSF and TSF, and this was only observed for the longer NILT simulations (*i.e.* higher N_H and N_m).

With regard to the more numerically involved LUF, TUF and VE pipe types, the NILT was unconditionally more efficient (except for the 11-pipe $\{N_H, N_m\} = \{1000, 5\}$ case). The more expensive MOC times for these cases are clearly attributed to the increased number of numerical operations involved in evaluating the convolutions for the unsteady friction and viscoelastic operators. Using the N_H values from Tables 5.1-5.4, a NILT simulation with a small error[19] for 5 nodes was observed to be on average 50%, 17%, 14% and 40% of the MOC time for the 11-pipe, 35-pipe, 51-pipe and 94-pipe networks, respectively. As the LUF and the TUF pipe types were more numerically involved than the VE pipe type for the MOC[20], the

[19]Here *small* error is taken to mean: less that 1% for the 11-pipe network (*i.e.* the case $N_H = 500$); less that 3% for the 35-pipe network (*i.e.* the case $N_H = 1000$); less that 2% for the 51-pipe network (*i.e.* the case $N_H = 500$); and less than 4.1% for the 94-pipe network (*i.e.* the case $N_H = 2000$)

[20]A 10-term model was used for the LUF, a 13-term model for the TUF, and a single term model for the VE, as detailed in Examples 5.3-5.5 respectively.

Table 5.7: Computational times[a] of the NILT relative to the MOC for the 51-pipe case study. The computational times for the MOC simulations (in CPU seconds) are given in italics. All times are based on the computation of 10^5 simulation time points.

N_H	N_m	Relative computational times for pipe type				
		LSF	TSF	LUF	TUF	VE
1000	10	1.614	1.430	0.545	0.419	0.890
	5	0.753	0.613	0.260	0.238	0.524
	2	0.428	0.354	0.160	0.103	0.263
	1	0.327	0.334	0.100	0.068	0.177
500	10	0.638	0.562	0.203	0.203	0.313
	5	0.374	0.329	0.120	0.121	0.186
	2	0.216	0.189	0.069	0.059	0.110
	1	0.162	0.141	0.053	0.040	0.109
250	10	0.296	0.260	0.138	0.071	0.140
	5	0.174	0.152	0.073	0.060	0.083
	2	0.101	0.088	0.033	0.032	0.049
	1	0.076	0.066	0.025	0.025	0.038
(MOC		*456.7*	*523.2*	*1458.5*	*1911.9*	*874.7)*

[a]Simulations were performed on a 2.13GHz Linux machine were the CPU timings were evaluated by the `procstat` routine.

computational saving of the NILT is greater for these cases.

5.6.4 Additional studies

Two additional topics associated with the use of the NILT are now presented. These studies highlight limitations and advantages, respectively of the NILT for application to time-domain simulation of hydraulic networks. Firstly, the limitation is associated with the increase in the linearisation error in cases where the operating point for the network changes dramatically from that about which the NILT was linearised. Secondly, the ability of the NILT to jump to any time point (without the need to calculate the preceding time points) is demonstrated to have advantageous characteristics in comparison to the MOC.

Step changes in nonlinear networks

For the linear NILT approximation of the nonlinear network, the linearisation error within each pipe is dependent on $(q(t) - q_0)^2$, the square of the deviation of the flow from the operating state. As seen in Examples 5.2 and 5.4, when this deviation is

Table 5.8: Computational times[a] of the NILT relative to the MOC for the 94-pipe case study. The computational times for the MOC simulations (in CPU seconds) are given in italics. All times are based on the computation of 10^5 simulation time points.

N_H	N_m	Relative computational times for pipe type				
		LSF	TSF	LUF	TUF	VE
2000	10	1.826	2.047	0.527	0.389	0.828
	5	1.201	1.266	0.338	0.256	0.596
	2	0.739	0.803	0.236	0.169	0.429
	1	0.630	0.695	0.193	0.141	0.291
1000	10	0.938	1.059	0.369	0.271	0.521
	5	0.560	0.601	0.212	0.165	0.357
	2	0.363	0.422	0.144	0.106	0.237
	1	0.301	0.337	0.120	0.089	0.198
500	10	0.422	0.479	0.158	0.133	0.244
	5	0.304	0.299	0.101	0.074	0.178
	2	0.192	0.199	0.070	0.050	0.113
	1	0.129	0.164	0.058	0.042	0.092
250	10	0.203	0.216	0.088	0.066	0.139
	5	0.131	0.143	0.053	0.040	0.085
	2	0.085	0.093	0.035	0.026	0.056
	1	0.070	0.078	0.029	0.021	0.045
(MOC		*3235.0*	*2865.4*	*8313.0*	*11335.0*	*5385.1.0)*

[a]Simulations were performed on a 2.13GHz Linux machine were the CPU timings were evaluated by the `procstat` routine.

large, a trend is introduced into the error functions. This trend, arising from the linearisation error, is particularly prevalent in the cases were the system is excited into a different base level operating state. That is, cases where the steady state operating point is shifted from q_0 to $\acute{q}_0$. Such changes occur when the hydraulic configuration of a system is changed (*i.e.* a valve is closed) or when the hydraulic scenario has infinite energy boundary condition inputs, such as step changes in nodal flows. In such cases, the NILT will converge to a steady-state value that is different from the actual steady-state value of the nonlinear system. This issue only applies to nonlinear systems, and is explored in the following examples and discussion.

Example 5.8. *Consider the 11-pipe TUF network from Section 5.6.2, but with a step reduction of 60 L/s to the demand at node 6 (see Appendix D for details). The pressure response at node 2 as computed by the MOC and the NILT is given in Figure 5.18(a), where for the NILT a and* $\Delta\omega$ *are calculated by* (5.15)-(5.16) *and* $N_H = 1000$.

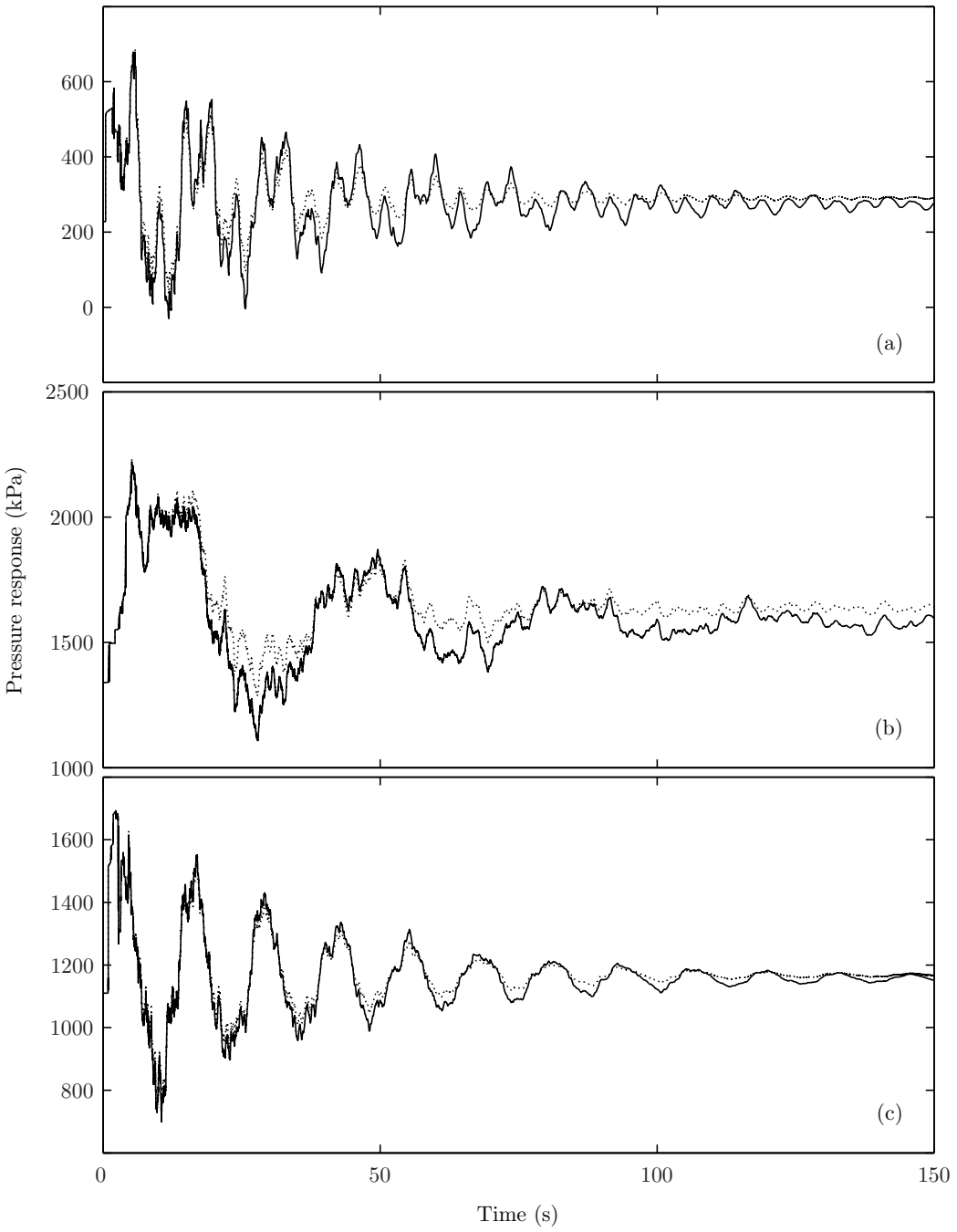

Figure 5.18: Comparison of the pressure response computed by the MOC (−) and the NILT($\cdots$) for the TUF networks with step inputs. The figures are of (a) node 2 for the 11-pipe network, (b) node 5 for the 35-pipe network, and (c) node 25 for the 51-pipe network.

For this example, in the early stages of the pressure response in Figure 5.18, the error between the NILT and the MOC are visible with the NILT tending to approximate higher pressures. As t is increased and the transient fluctuations dissipate, it is clear that the NILT is converging to a different steady-state value than the nonlinear MOC.

Example 5.9. *Consider the 35-pipe TUF network from Section 5.6.2, but instead of pulse perturbations at nodes* $\{3, 11, 14, 15\}$*, the demand at these nodes is reduced to zero in* $\{0.2, 0.05, 0.1, 0.01\}$ *s corresponding to flow changes of* $\{1.17, 0.17, 0.67, 0.25\}$ m^3/s*. The pressure response at node 5 as computed by the MOC and the NILT is given in Figure 5.18(b), where for the NILT a and* $\Delta\omega$ *are calculated by* (5.15)-(5.16) *and* $N_H = 1000$*.*

As seen in Figure 5.18(b), the first pressure surge for the network in Example 5.9 is captured with great accuracy, but the accuracy is greatly reduced within about the first 10 s of the simulation time. The loss in accuracy of the NILT is particularly manifest in the low pressure surge regions. As with Example 5.8, as t increases, it is clear that the NILT from Example 5.9 is converging to a different steady-state level than the MOC, but an interesting point to note is that exact nature of the fluctuating behaviour of the MOC is followed by the NILT. A qualitatively similar behaviour is observed in the following example.

Example 5.10. *Consider the 51-pipe TUF network from Section 5.6.2, but instead of pulse perturbations at nodes* $\{12, 17, 27, 31\}$*, the demand at these nodes is reduced to zero in* $\{0.01, 0.03, 0.1, 0.05\}$ *seconds corresponding to flow changes of* $\{0.11, 0.23, 0.28, 0.17\}$ m^3/s*. The pressure response at node 5 as computed by the MOC and the NILT is given in Figure 5.18(c), where for the NILT a and* $\Delta\omega$ *are calculated by* (5.15)-(5.16) *and* $N_H = 1000$*.*

Despite the large transient pressure surges, the NILT approximated the initial surges in the shorter time scales with high accuracy for all examples. It is only at longer time scales, where the cumulative nonlinear dissipation become significant, that the accuracy of the linear NILT is reduced. The additional nonlinear dissipation of the MOC explains the observation common to all examples that the NILT tended to underestimate the energy loss in the networks.

Discretisation-free simulation

Two main advantages of the NILT are that (i) the continuous distributed nature of the system is retained, and (ii) the function value $\tilde{f}(t)$ at any point t can be computed directly without first computing $\tilde{f}(\tau)$ for $\tau \in [0, t)$. Point (i) implies that the true wave propagation delays are correctly retained, and (ii) implies the accuracy of the simulation is not dependent on any form of refined computational grid. These properties do not hold for any discrete method.

As the Laplace-domain representation of a fluid line retains the distributed nature of the line, the resulting time-domain NILT also inherits the true distributed

nature of the line or network[21]. To explain this further, the pipeline transfer matrices (2.47)-(2.50) are derived for the true continuous spatial and temporal pipeline state-space and no discretisation approximations of this state space are required. As the spatial and temporal distributed properties are retained in the LT, they are also inherited by the ILT. Practically what this means for the time-domain ILT is that the distributed capacitive and resistance effects and, more importantly, the wave propagation delays are correctly captured by the ILT.

In comparison with discrete methods, the continuous state space requires both a spatial and a temporal discretisation. This need for discretisation leads to a trade-off between accuracy and computational efficiency. For applications with hyperbolic PDEs, it is fundamentally important that the computational grid is designed so as to correctly capture the wave propagation delays. As discussed earlier, if the computational grid is not correctly generated, numerical artefacts can be induced in the discrete simulation [*Wiggert and Sundquist*, 1977; *Goldberg and Wylie*, 1983; *Ghidaoui and Karney*, 1994; *Ghidaoui et al.*, 1998]. For the MOC, the numerical criteria to correctly capture wave propagation delays is that the Courant number $\mathbb{C}_r = 1$ for all grid reaches in all pipes. In circumstances where this is not possible[22], the weaker stability criteria of $\mathbb{C}_r \leq 1$ (for all reaches in all pipes) is used as the guideline for the design of interpolation schemes [*Wiggert and Sundquist*, 1977; *Goldberg and Wylie*, 1983]. The use of such schemes, however, is not preferable, as numerical dispersion is introduced into the scheme creating significant nonphysical artefacts [*Ghidaoui and Karney*, 1994]. A simple alternative that avoids such numerical artefacts is the wavespeed adjustment scheme [*Wylie and Streeter*, 1993; *Ghidaoui et al.*, 1998]. This scheme effectively changes the pipelines wavespeed to achieve the Courant condition of $\mathbb{C}_r = 1$ for all grid reaches in all pipes. The set back of this approach, however, is that the wave propagation delays are not exact [*Wylie and Streeter*, 1993].

Computational grid design is also clearly related to computational efficiency. That is, unlike the NILT, discrete methods must compute the state for the entire range $t = 0, \Delta t, \ldots, (\ n\ -\ 1\ )\ \Delta\ t$ to be able to compute the state at $t = n\Delta t = T$. Therefore, as the computational grid is refined, to compute the state at $t = T$ requires more time points, meaning a greater number of computations.

Two examples are presented in the following. Within these examples the MOC grid is coarsened to produce an algorithm that computes the pressure response on

[21]To be precise, this statement is strictly true only for the ILT, and it holds only approximately for the NILT. The intention here, however, is to highlight that the true wave propagation delays are retained in the NILT.

[22]Within a network context, it is typically computationally infeasible to achieve the Courant condition $\mathbb{C}_r = 1$ as the grid refinement can be too great.

$t \in [0, 100]$ in a similar computational time to the NILT with 10 measurement points, where the reduction factors are determined from Tables 5.6 and 5.7[23]. The relative accuracies of each method is discussed.

Example 5.11. *Consider the network and hydraulic scenario for the 35-pipe LUF network from Example 5.3. Consider changing the computational time step from* $\Delta t = 0.001$ *to* $\Delta t = \sqrt{10}/1000$*, which implies a increase in the length of the spatial reaches by a factor of* $\sqrt{10}$ *(this corresponds to an order of magnitude reduction in computational time). Applying the wavespeed adjustment scheme to this grid yields a maximum wavespeed change of 0.16% (*i.e. *1000 to 998.4 m/s). Figure 5.19(a) gives a comparison of the original MOC with the coarse MOC and Figure 5.19(b)-(d) compare the errors of the coarse grid MOC with the NILT error functions where* $\Delta E_{N_H}(t) = |E_{coarse}(t)| - |E_{N_H}(t)|$*. Positive values of* $\Delta E_{N_H}(t)$ *are associated with a higher error in the coarse grid MOC than the NILT. By scaling the relative computational times of the NILT with the original MOC in Table 5.6, the computational cost of the NILT with* $\{N_H, N_m\} = \{1000, 10\}$ *and the coarse MOC can be demonstrated as being approximately equal.*

Despite the fact that it is difficult to distinguish between the fine and course grid MOC in Figure 5.19(a), Figures 5.19(b)-(d) demonstrate that the error in the course grid MOC was up to near 200 kPa greater than the error in the NILT approximation. The magnitude of this error is suprising given that the maximum change in the wavespeed for the networks pipes was only 0.16%. The error of the coarse MOC remains around 50-70 kPa greater than $E_{1000}(t)$ for the longer time scales.

Example 5.12. *Consider the network and hydraulic scenario for the 51-pipe LUF network from Example 5.3. Consider changing the computational time step from* $\Delta t = 0.001$ *s to* $\Delta t = 0.005$ *s, which implies an increase in length of the spatial reaches by a factor of* 5 *(this corresponds to reduction in computational time by a factor of 25). Applying the wavespeed adjustment scheme to this grid yields a maximum wavespeed change of 0.44% (*i.e. *1000 to 1004.4 m/s). Figure 5.20(a) gives a comparison of the original MOC with the coarse grid MOC and Figure 5.20(b)-(d) compare the errors of the coarse grid MOC with the NILT error functions where* $\Delta E_{N_H}(t) = |E_{coarse}(t)| - |E_{N_H}(t)|$*. Positive values of* $\Delta E_{N_H}(t)$ *are associated with a higher error in the coarse grid MOC than the NILT. By scaling the relative computational times of the NILT with the original MOC in Table 5.7, the computational*

[23]Note that increasing the MOC time step by a factor of c actually decreases the computational time to compute the response in a given time window by a factor of c^2. This is because a coarsening in the time grid is accompanied by a coarsening in the spatial grid. Therefore, not only are $1/c$ of the time points computed, but also $1/c$ of the spatial points are computed too. Computing the NILT on the coarser grid simply reduces its computational time by a factor of c.

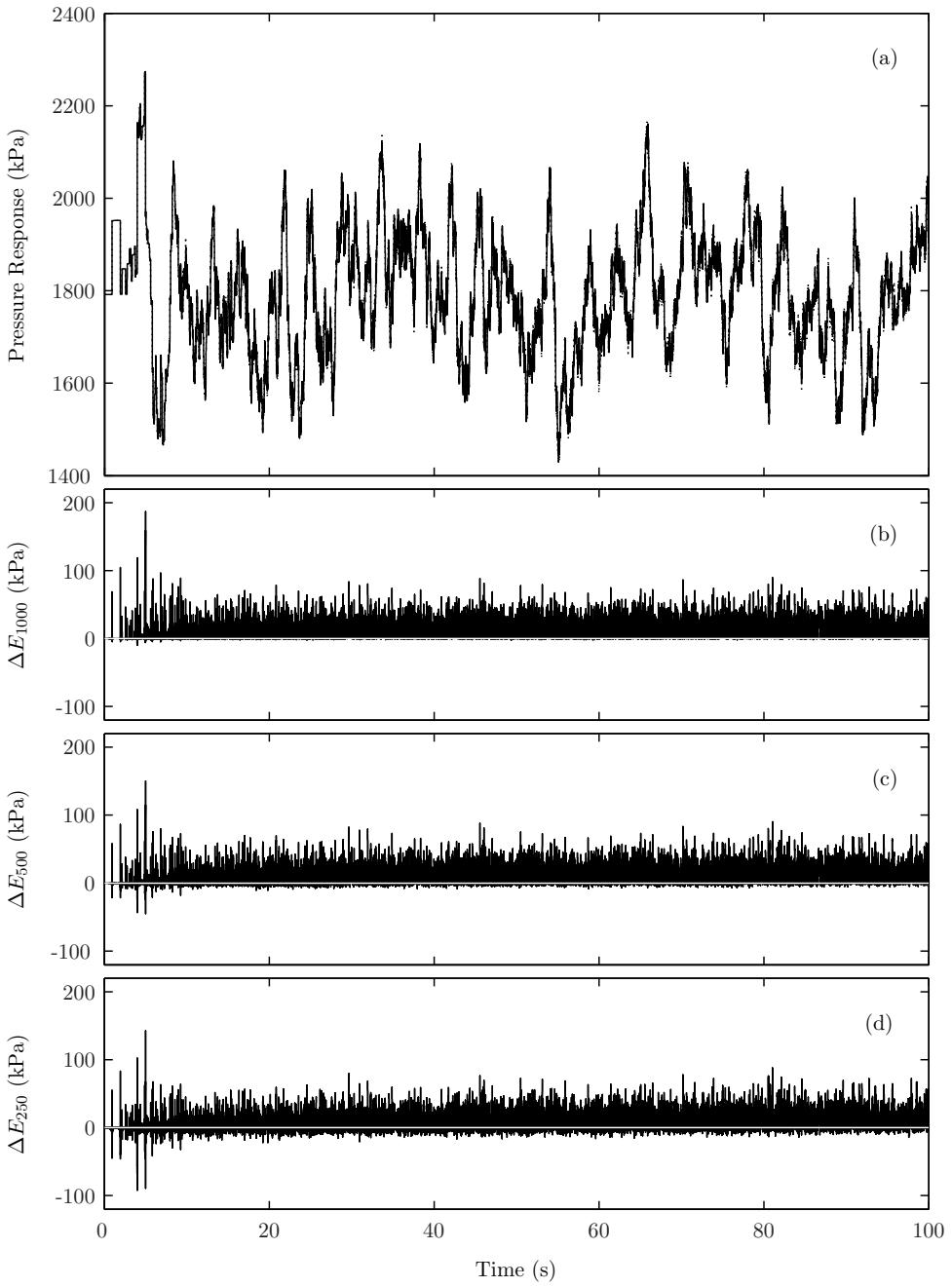

Figure 5.19: Comparison of the coarsely discretised MOC ($\Delta t = \sqrt{10}/1000s$) with the finely discretised MOC and the NILT for the LUF 35-pipe network from Example 5.11 for the pressure response at node 5: (a) the pressure response computed by the fine ($-$) and coarse ($\cdots$) MOC, (b)-(d) the error differences $\Delta E_{N_H}(t) = |E_{MOC}(t)| - |E_{N_H}(t)|$ for $N_H = 1000$, 500, and 250, respectively, where $E_{coarse}(t)$ is the error of the coarse MOC approximation.

cost of the NILT with $\{N_H, N_m\} = \{500, 10\}$ *and the coarse grid MOC can be demonstrated as being approximately equal.*

Example 5.12 yields similar results to Example 5.11, with the maximum wavespeed

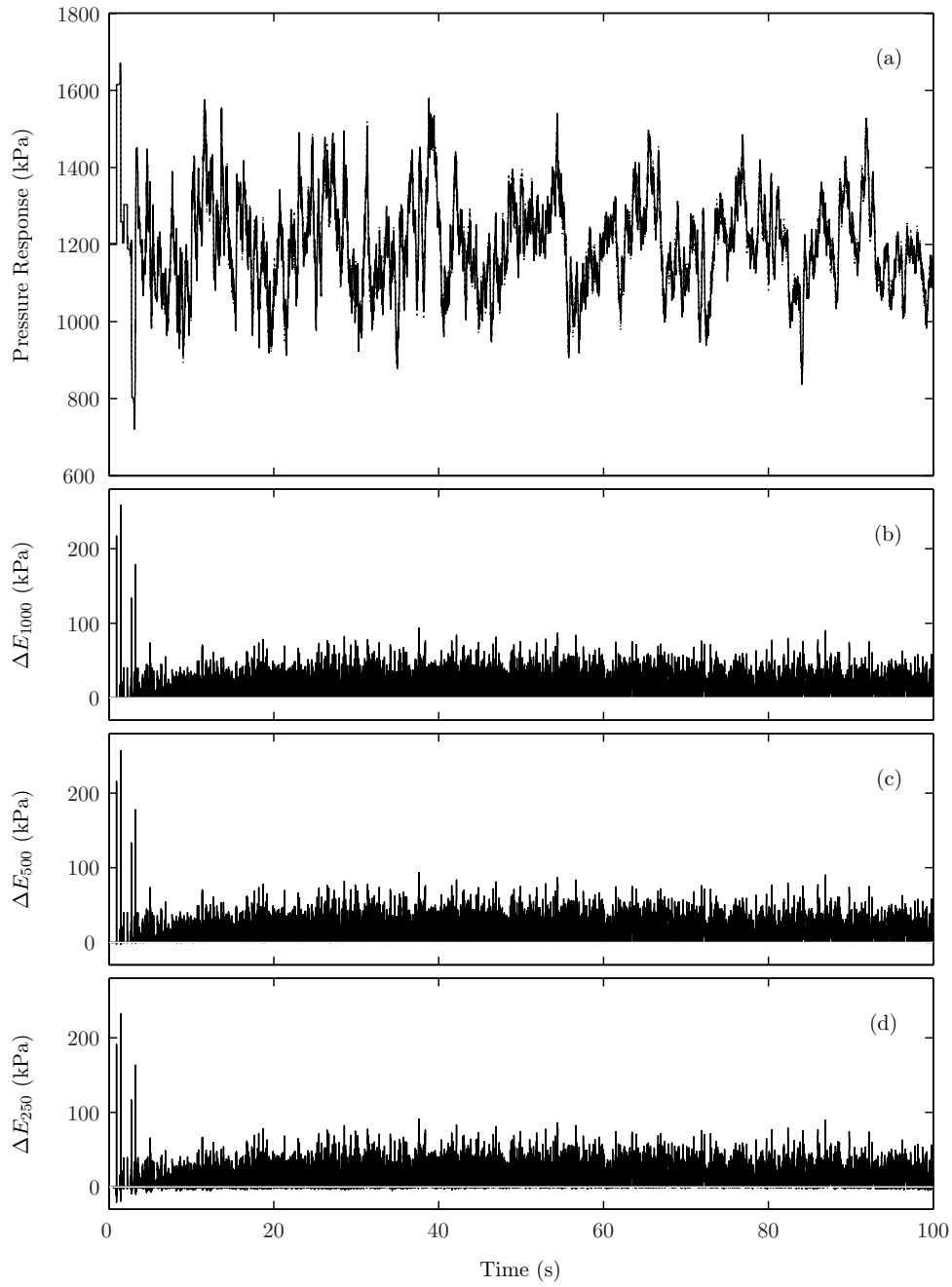

Figure 5.20: Comparison of the coarsely discretised MOC ($\Delta t = 0.005$ s) with the finely discretised MOC and the NILT for the LUF 51-pipe network from Example 5.12 for the pressure response at node 25: (a) the pressure response computed by the fine ($-$) and coarse ($\cdots$) MOC, (b)-(d) the error differences $\Delta E_{N_H}(t) = |E_{MOC}(t)| - |E_{N_H}(t)|$ for $N_H = 1000$, 500, and 250, respectively, where $E_{coarse}(t)$ is the error of the coarse MOC approximation.

change of 0.44% leading to a coarse grid MOC error of nearly 300 kPa greater than the NILT. More generally though, what both these examples demonstrate is the ability of the NILT to accurately compute a networks transient response on a time grid that does not coincide with natural discretisation of the network as defined by

a Courant number of $\mathbb{C}_r = 1$ for all reaches in all pipes.

5.7 Conclusions

The focus of this chapter has been on the use of the linear Laplace-domain network model from Chapters 3 and 4 as an alternative time-domain hydraulic simulator by way of the NILT. The use of the ILT in the development of time-domain models from their LTs has been extensive, however little attention has been given to full network models, with the exception being the IPREM [*Suo and Wylie*, 1989]. The approach presented here is entirely novel in that it couples the Laplace-domain input/output model in a computationally efficient way with the Fourier series NILT from *Abate and Whitt* [1995]. The parameters of the NILT have been studied in detail for simple test functions and robust parameter heuristics have been developed. These heuristics were successfully used in the application of the NILT to 20 different case studies (four different networks in five different pipe types). The focus of the studies were on the accuracy and computational efficiency of the proposed NILT.

For the cases of large finite energy perturbations (*e.g.* pulses and temporary flow stoppages), the NILT was found to provide highly accurate approximations for all case studies, even in networks with nonlinear pipe types. The accuracy was observed to be greater for the more highly dissipative networks. However, in cases of large infinite energy perturbations in nonlinear networks (*e.g.* step inputs), the NILT was observed to only accurately approximate the initial surge, with poorer accuracy in the longer time scales. The loss of accuracy in the longer time scales is attributed to the permanent shift in the operating point about which the NILT was linearised.

For large networks, NILT was found to be computationally efficient with respect to the MOC. This relative efficiency was observed to be especially true for the case studies with more complex pipe types involving convolution operations, as these operations exert little additional computational time on the NILT. In addition to the computational efficiency, the NILT possesses the desirable property that it correctly captures wave propagation delays without the need for fine computational grids. This property arises from the fact that the NILT does not involve the discretisation of the network state, and it is able to compute the network state at any time point without computing the state at the preceding time points. As such, the NILT represents an ideal approach for modelling networks involving pipes with greatly varying wavespeeds.

With regards to computational efficiency, an important point to note is that the NILT computes the hydraulic state variables only at points of interest selected by the

user (up to 10 points were considered within this chapter). However, an advantage of the MOC is that it provides the state variables at all computational points within the network. As such, the computational advantage of the NILT would diminish if the state variables at a large number of points required calculation.

Given the benefits and limitations of the NILT for time-domain simulation, it has the potential to play a meaningful role in applications, whether it be (i) general efficient modelling of linear networks, (ii) accurate and efficient initial surge calculations for nonlinear networks, (iii) a validation model for the assessment of computational grids for discrete models, and (iv) a complementary model to discrete models to efficiently investigate the dynamic behaviour of a network at resolutions finer time scales in between the time points of discrete computations.