Kalman Filtering and its Application to On-Line State Estimation of a Once-Through Boiler

A THESIS PRESENTED
BY
ZUBEIDA PATEL
TO
THE DEPARTMENT OF ELECTRICAL ENGINEERING

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF CAPE TOWN
MARCH 2021

PLAGIARISM DECLARATION

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This thesis has been submitted to the Turnitin module and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Student Number: PTLZUB003 **Signature:** Signed by candidate

DECLARATION ON THE INCLUSION OF PUBLICATIONS

I confirm that I have been granted permission by the University of Cape Town's Doctoral Degrees Board to include the following publications in my PhD thesis, and where co-authorships are involved, my co-authors have agreed that I may include the publications:

1. Z.Patel and E. Boje, "Application of the Continuous-Discrete Kalman Filter to State Estimation of the Convection Pass of a Once-Through Power Boiler," in *2019 American Control Conference (ACC)*, 2019, pp. 3167-3173, IEEE, 2019.

2. Z.Patel and E. Boje, "A Hybrid, Coupled Approach to the Continuous-Discrete Kalman Filter," *IEEE Control Systems Letters*, vol.5, no.3, pp. 827-832, 2020.

Student Number: PTLZUB003 **Signature:** Signed by candidate

Kalman Filtering and its Application to On-Line State Estimation of a Once-Through Boiler

Abstract

This thesis contributes to non-linear continuous-discrete Kalman filtering of multiplex systems through the development of two main ideas, namely, integration of the unscented transforms with linearly implicit methods and incorporation of simulation errors in the state estimation problem. The newly developed techniques are then applied to the technically relevant problem of state estimation on the main components of a utility boiler. State estimators in industrial systems are used as soft-sensors in monitoring and control applications as the most cost effective and practical alternative to telemetering all variables of interest. One such example is in utility boilers where reliable and real-time data characterising its behaviour is used to detect faults and optimise performance. With respect to the state-of-the-art, state estimators display limitations in real-time applications to large-scale systems. This motivates theoretical developments in state estimation as a first part in this thesis. These developments are aimed at producing more practical and efficient algorithms in non-linear continuous-discrete Kalman filtering for stiff large-scale industrial systems. This is achieved using two novel ideas. The first is to exploit the similarities between the extended and unscented Kalman filter in order to estimate the Jacobian required for linearly implicit schemes, thereby tightly coupling state propagation and continuous-time simulation. The second is to account for numerical integration error by appending a stochastic local error model to the system's stochastic differential equation. This allows for coarser integration time steps in systems that are otherwise only suited to relatively small step sizes, making the filter more computationally efficient without lowering its potential to construct accurate estimates. The second part of this thesis uses these algorithms to demonstrate the feasibility of on-line state estimation on the main components of a once-through utility power boiler that require in excess of a hundred state variables to capture its behaviour with adequate fidelity. Two separate models of the boiler are developed, a MATLAB® and a Flownex® model, comprising the economiser, evaporators, reheaters, superheaters and furnace. The mathematical MATLAB® model is better suited to real-time execution and is used in the filter. The more sophisticated model is based on a commercial thermal-hydraulic simulation environment, Flownex®, and is used to validate the mathematical modelling philosophies and construct filter observation data. After validating the performance of the filter against ground-truth data provided by the Flownex® model, the filter is demonstrated on historical plant data to illustrate its utility.

Contents

List of figures

List of Abbreviations

ATT	attemperator
CBF	cubature Kalman filter
CD	continuous-discrete
CFD	computational fluid dynamics
DD1	first-order divided difference estimator
DD2	second-order divided difference estimator
DIRK	diagonally implicit Runge-Kutta
EKF	extended Kalman filter
EM	Euler-Maruyama
EnKF	ensemble Kalman filter
GHQF	Gauss-Hermite quadrature filter
GPF	Gaussian particle filter
HCKF	hybrid coupled Kalman filter
HHV	higher heating value
HOLM	higher-order linearised matrix
HOT	higher-order terms
IAPWS	international association for the properties of steam and water
IF97	industrial formulation 1997
IVP	initial value problem
KBF	Kalman-Bucy filter
KF	Kalman filter
LKF	linear Kalman filter
MC	Monte Carlo
MCR	maximum continuous rating
MDE	moment differential equation
MGL	modified Gauss-Legendre
MSE	mean squared error
NTU	number of transfer units
ODE	ordinary differential equation
OPC	object linking and embedding for process control
PD	positive-definite
PDE	partial differential equation
PF	particle filter
PID	proportional integral derivative
R/H	reheater
RK	Runge-Kutta
SDE	stochastic differential equation

S/H	superheater
SIRK	semi-implicit Runge-Kutta
TH	Taylor-Heun
UKF	unscented Kalman filter
UT	unscented transform

Nomenclature

Constants and Parameters

a	unscented parameter
β	unscented parameter
η	negative complex number in the scalar test equation
γ	a characteristic of a numerical method
κ	unscented parameter
λ	stochastic order of strong convergence
θ	a scalar in the open interval between 0 and 1
L	number of observation points in the filter's simulation trajectory
N	number of integration steps in a single measurement step
d	number of noise sources
m	number of measurements
n	number of state variables
p	order of a numerical method
s	number of stages in a numerical method

Accents, Subscripts and Superscripts

$()'$	derivative with respect to the state
$()_i$	integration time instance
$()_k$	measurement time instance
$\bar{()}$	mean
$\hat{()}$	*a priori* state estimate
$\tilde{()}$	error

Number Sets

$\mathbb{C}^+$	the set of all complex numbers with positive real part and not including 0
$\mathbb{C}^-$	the set of all complex numbers with negative real part and not including 0
$\mathbb{R}^+$	the set of all positive real numbers not including 0
$\mathbb{R}^-$	the set of all negative real numbers not including 0

Operators

$\mathscr{L}$	Kolmogorov's forward operator
L^0, L^1	gradient operators

Thermodynamic Variable Subscripts

con	convection
f	fluid
g	gas (flue)

i	inner
in	into a discrete segment
m	tube wall
o	outer
out	out of a discrete segment
pf	pulverised fuel
rad	radiation

Thermodynamic Variables

$\dot{m}$	mass flow rate
$\dot{q}$	heat transfer rate
η	viscosity
λ	thermal conductivity
$\mathbb{h}$	heat transfer coefficient
μ	viscosity
v	velocity
ρ	density
ξ	frication factor
A_{ff}	free flow area
A_s	surface area
C_p	heat capacity
c_p	specific heat capacity
d	tube diameter
D_H	hydraulic diameter
h	specific enthalpy
K	forward loss coefficient
k	roughness coefficient
L_{flow}	flue gas path length
m	mass
Nu	Nusselt number
Pr	Prandtl number
Re	Reynolds number
T	temperature
V	volume

Variables and Function Variables

χ	unscented transform of the state variables
Δ	measurement step size
δ	integration step size
Λ	non-linear component of the measurement HOLM
$\mathcal{B}$	diffusion coefficient HOLM
$\mathcal{C}$	error function scaling matrix
$\mathcal{D}$	error function scaling matrix after regression analysis
$\mathcal{H}$	measurement HOLM
$\mathcal{J}$	drift HOLM
$\mathcal{L}$	unscented transform of the state-dependent diffusion coefficient
$\mathcal{N}$	normal/Gaussian distribution
$\mathcal{R}$	stability function
$\mathcal{U}$	uniform distribution
μ	logarithmic norm function

Ω	non-linear component of the diffusion coefficient matrix
ω	upper bound of the logarithmic norm of the Jacobian
Φ	increment function
φ	inverse solution matrix in the linearly implicit scheme
Π	state transition matrix
Ψ	measurement function
ρ	drift function of the state error covariance
σ	standard deviation
υ	upper bound of the defect
Υ	non-linear component of the drift HOLM
ϕ	scalar function of the *Itô* SDE
ϑ	butcher table element
ξ	derivative of the Wiener process
A	butcher table matrix
a	butcher table element
b	state-dependent diffusion coefficient
C	resultant butcher table matrix
c	butcher table element
D	a function in Richardson's extrapolation
e	global error function
F	linear drift function
f	drift function of the state variables
G	linear measurement function
H	measurement Jacobian
J	drift Jacobian
K	Kalman gain
l	local error function
$L(\cdot)$	loss function
M	solution matrix in the linearly implicit scheme
m	state mean of the UKF
P	state error covariance
p	probability density function
q	diffusion coefficient
Q_c	continuous-time process covariance
R	measurement covariance
r	defect
s	system inputs
U	scalar transformation function
u	measurement noise
v, w	process noise
W	unscented parameter matrix
x	state variables
X, Y and Z	arbitrary scalar random variables
Y	a sequence of observations
y	output measurements from the actual system
z	predicted output measurements

To my parents.

Acknowledgments

I would like to thank my supervisor for his constant guidance and for the many fruitful discussions which helped shape this thesis. I would furthermore like to thank the National Research Foundation (NRF) South Africa, the Eskom Tertiary Education Support Programme, the Vice Chancellor Research Fund, the Ada and Bertie Levenstein Fund and the Oasis Crescent Trust for their financial support. I would like to acknowledge the help of the Eskom Power Plant Engineering Institute (EPPEI) in acquiring the relevant data, drawings and software. A thank you to the EPPEI group for inviting me to participate in their colloquiums to help build my mechanical engineering knowledge base, and to the Mechatronics group for borrowing me their i9 PC and creating a pleasant working environment.

1 Introduction

Part 1 of this thesis develops theoretical aspects of the continuous-discrete (CD) Kalman filter (KF) for stiff large-scale systems. This is motivated by the practical application of the CD-KF to the problem of on-line state estimation of a once-through utility power boiler in Part 2 of this thesis. The contribution of Part 2 is to assess the viability of on-line state estimation on a once-through boiler and demonstrate its ability to detect common problems such as fouling in the convection pass. The use of the CD unscented Kalman filter (CD-UKF) to detect fouling is published in [3]. In this paper, however, implicit integration is more difficult and real-time implementation for a scale up of n=136 state variables is far from achievable. The various developments in Part 1 are therefore geared towards greater filter efficiency and practicality in such large-scale applications by designing more suitable algorithms based on the CD extended Kalman filter (CD-EKF), the CD-UKF and select ideas from Monte Carlo methods. The contribution of Part 1 is the development of new filter approaches, the CD hybrid coupled Kalman filter (CD-HCKF), published in [4], the CD-HCKF_Milstein and the CD-HCKF_EM. The former uses the relationship between the CD-EKF and CD-UKF to extract an equivalent of the system Jacobian, effectively coupling the filtering process to linearly implicit model simulation. This makes state estimation in complex stiff systems more practical and efficient by avoiding the need for analytic or numerical Jacobians, reusing matrices from the simulation in the filter and facilitating numerical/computational stability. The CD-HCKF_Milstein and CD-HCKF_EM are variants of the CD-HCKF and embed integration error from the linearly implicit solver in its moment differential equations, formulated using a Milstein and Euler-Maruyama (EM) discretisation of the stochastic differential equation (SDE). This helps maintain filter accuracy when larger integration step sizes are employed for faster simulation.

Part 1 begins with an introduction to Kalman filters (Chapter 2). Chapter 3 then discusses the solutions of general ordinary differential equations (ODEs) and contributes to the development of an accurate estimate of the local and global error for linearly implicit Runge-Kutta (RK) methods used in subsequent chapters. The novel CD-HCKF, applicable to stiff large-scale non-linear problems, is developed in Chapter 4, followed by its extension to an amended SDE in Chapter 5 that results in the CD-HCKF_Milstein and CD-HCKF_EM. Part 2 begins with an introduction to state estimation in power boilers (Chapter 6). Chapter 7 details the development of two models for the boiler under investigation. Chapter 8 uses these models and the algorithms developed in Part 1 to set up and design a filter for potential on-line state estimation of over a hundred state variables associated with the main components of the boiler. Chapter 9 then summarises the contributions and findings in Part 1 and 2 of this thesis.

Part I

2

Kalman Filtering

The filtering problem is a classic, centuries-old challenge grounded in Bayesian theory [5]. Considered as the stochastic equivalent to the deterministic Luenberger observer, it poses the question - what is the best estimate of the state a system given a set of observations? Its least squares solution was first put forward by Wiener and Kolmogorov for a stationary linear stochastic process with scalar observations [6]. It was Kalman who then reformulated the problem using a state-space approach under the assumption of white Gaussian noise, from which the recursive discrete-time Kalman filter (KF) and Kalman-Bucy filter (KBF) came to be. This was followed by the introduction of approximate non-linear filters such as the extended and unscented Kalman filter (EKF and UKF), [7],[8]. Other notable filters in this category include the cubature Kalman filter (CKF) [9] and the Gauss-Hermite quadrature filter (GHQF) [10] formulated using quadrature rules. These filters typically assume Gaussian noise models and are consequently referred to as Gaussian filters in the literature. Since there exists no closed form solution of the underlying dynamic equations or the governing stochastic differential equation, these non-linear filters approximate the probability density of the system trajectory via (an approximation of) its first two central moments. The Gaussian sum filter, first reported in [11], comes closer than the Gaussian filter to approximating the true density function of a non-linear system (not necessarily Gaussian) as a weighted sum of Gaussian functions. A non-Gaussian state estimator is not restricted solely to Gaussian noise models. An example is the particle filter (PF) [12] and is based on Monte Carlo (MC) methods. It is shown in [13] that in order for the PF to achieve its reported performance in the literature, the number of particles used in the filter has to be greater than $n^{2.5}$, where n is the number of state variables in the system. This problem is eased in [14] which sees a reduction in the number of particles to n^2, achieved by partitioning the state-space and employing the simplified Gaussian PF (GPF) [15]. Another Monte Carlo type method is the

ensemble Kalman filter (EnKF) [16]. Its noise models are assumed Gaussian and unlike the UKF, its sample points (and those of the PF), synonymous with the UKF sigma points, are not chosen deterministically. This filter is used mainly in meteorology and oceanography [17]. When tested in non-linear applications outside of these fields, the EnKF (as with the PF) is shown in [18] to require an ensemble size much greater than the number of sigma points used in the UKF (or the CKF) in order to achieve improved accuracy over the EKF; while for larger state vectors with $n > 100$ a performance limit is reached at a certain sample size, after which the EnKF exhibits the same accuracy levels as the EKF. It is further shown in [17] that for a linear problem, the EnKF is only able to compete with the KF if its ensemble size grows exponentially with the number of state variables. These Monte Carlo based filters are therefore rather expensive when considering their performance with respect to a general set of problems. Most of the aforementioned filters are, however, applied in a diverse range of fields to the task of estimation, lending themselves to continuous-discrete (explained in the next section) and square root versions [7],[19],[20],[21].

The UKF, GHQF, CKF and PF offer superior performance over the first-order EKF, but as mentioned before, suffer from the curse of dimensionality. Computational intensity increases linearly with the number of states in the case of the UKF and CKF, and exponentially in the case of the GHQF [22] and PF. Comparable performance of the third degree CKF and UKF, both of which belong to a class of sigma point filters, is shown in [23], however, the authors in [24] have reported the UKF to generate faster and more accurate estimates for the problem of target tracking when compared to the CKF. Although the UKF and CKF, which uses a set of $2n+1$ and $2n$ sigma points respectively, have shown to offer similar accuracy and efficiency with respect to a broad selection of problems, due to the manner in which the CKF defines its weighting points, its square root version [9] has the advantage of better numerical stability in comparison to the square root UKF [21]. Cheaper unscented transformations using $n+2$ sigma points are available, but lead to numerical problems for high dimensional state vectors [69] and should therefore be avoided in large-scale systems. The aforementioned sigma point filters form part of derivative-free filtering methods and have become prominent in situations where systems exhibit non-trivial Jacobians. Another derivative-free filter is developed in [25]. This approach replaces function derivatives with central divided difference approximations according to Stirling's Interpolation Formula and demonstrates its improved performance over its Taylor approximated equivalent. Work in [26] then goes on to compare the performance of the filters in [25], referred to as the first- and second-order divided difference estimators (DD1 and DD2), to the UKF, where it is shown that the DD1 filter predicts understated state covariances while performance of the DD2 filter and UKF are comparable in some cases. Like the UKF or CKF, the DD2 appears to be a viable filter choice when system Jacobians are difficult to calculate.

2.1 Continuous-Discrete Filters

The filter formulations in this thesis are based on the continuous-discrete (CD) system description - a key theme in this thesis. Here models of physical systems are in continuous-time while measurements are in discrete-time. The result is a cross-over between the respective continuous- and discrete-time filters, namely the CD fil-

ter whose description in the context of the KF and EKF can be found in [7] or [27]. In [28], the discrete-time GHQF, UKF and CKF is extended to their respective CD cases. Implementation aspects of the CD-EKF are discussed in [29], [30] and [31] and focuses on accurate numerical solutions of the filter's first two moment differential equations (MDEs). The first two papers make use of variable step size embedded Runge-Kutta (RK) solvers to minimize simulation error during moment propagation, while the latter focuses on second-order accurate implicit RK methods for stiff systems - the second key theme in this thesis. Stiffness is characterised by state variables each evolving at much different rates. In other words, the ratio of the biggest to the smallest eigenvalue of a stiff system's Jacobian is very large. This necessitates either relatively small integration mesh sizes or implicit solvers to ensure stability and reasonable numerical accuracy [32],[33]. The filtering problem in stiff systems is further exacerbated if there is a large number of state variables, a characteristic of large-scale systems - the third key theme in this thesis. Various implicit methods for propagating the MDEs of large-scale stiff systems are shown in [34] and applied to the CD-UKF in [35]. Here the MDEs of the CD-UKF are solved jointly, according to the philosophy in [19]. This paper also establishes a simpler, vectorised form of the CD-UKF. Other sophisticated implicit numerical methods that can be applied to CD filters of stiff non-linear problems can be found in [36], [37] and [38]. In [39] a simpler means of calculating the respective state error covariances of the CD-UKF is shown. This avoids using the MDEs in [19], and rather makes a direct application of the unscented transforms (UTs), useful when a purely CD-UKF filter is to be employed. Another alternative formulation to the CD-UKF from [19] is seen in [40] where the measurement update process is performed according to a singular value decomposition of the state covariance. Here the authors report faster execution in certain cases only. Accurate solutions of the MDEs via implicit RK methods, as applied to the CD-EKF in [29], is applied similarly to the CD-CKF in [41]. In [42] the process covariance of the CD-CKF is integrated according to a technique from [31] to help preserve its symmetry and positive-definiteness. This structure of the covariances matrix is easily lost in sigma-point based CD filter formulations, see [43] and [44]. Since there is no neat solution of the covariance propagation matrix (that will preserve its symmetry and positive-definiteness) as there is for the CD-EKF in [31], the first square root version of the CD-UKF was developed in [21] to help prevent numerical instability. Although the square root version of the CD-UKF ameliorates the effect of numerical error, leading to improved numerical stability (and a smaller chance of covariance matrices losing positive-definiteness), the presence of negative weights in the sigma points means that the problem is still more apparent than in the case of the CD-CKF [45]. One way to avoid the need for a positive-definite covariance matrix in the CD-UKF is to avoid the use of the Cholesky factorisation altogether and use the hyperbolic QR decomposition instead, as shown in [46]. This, however, does not avoid the issue of numerical instability since the non-symmetric and non-positive-definite matrices are still used by the filter.

2.2 Higher-Order Filters and Other Filter Types

Another variant of the CD filter (a relatively new development) involves filters grounded in higher-order truncations of the stochastic differential equation (SDE). Whereas classic filters can be formulated using first-order approximations, the work in [45] is the first account of a filter based on a higher-order approximation in the

stochastic sense. In [47], the *Itô*-1.5 filter in [45] is compared to classic filters and displays better performance over these filters for the examples under investigation. Such higher-order filters are investigated in this thesis. Other noteworthy sub-variants include adaptive filters, decentralised filters, sliding window filters and event triggered filters. Some of these filters depend on system architecture, such as decentralised filters, and are only relevant when local estimates can be formed at individual sensor nodes. Other filters, such as sliding window filters or event triggered filters, have the potential to be useful in an advanced application following the design of the proposed filters in this thesis. In the context of CD filtering, event triggered filters only make an observation once a certain condition is met. In [48] the condition is based on the estimation variance while in [49] an observation is made if the mean squared error (MSE) is above a certain threshold. This serves as a means to control the estimation error of the filter[48].

2.3 Conclusion

Based on the literature review, PFs and GHQFs and their variants are avoided in the filter design since they do not scale well. The CD-CKF is also not used in the filter design because of its similarity to the CD-UKF. Instead, the new filter approaches will adopt processes solely from the CD-EKF, the CD-UKF and Monte-Carlo-based filters. The aim is to develop more efficient and practical CD-KF approaches for stiff large-scale systems, which is the focus of Chapters 4 and 5. Before this can be done, details surrounding the numerical schemes used in this thesis are discussed, and an accurate representation of its local and global error estimates developed in Chapter 3.

3

Numerical Methods in Model Simulation

This chapter discusses single-step numerical Runge-Kutta (RK) type solutions of the ODE in (A.19). It is a well known fact that implicit methods work much better in comparison to explicit methods when applied to stiff problems [32]. Since the primary filtering application in this thesis involves inherently stiff dynamics, the main focus of this chapter is on implicit RK methods and how to accurately estimate their local and global error. The most common methods used for local error estimation is based on local extrapolation. See [51] and [52]. This method of local error estimation can be extended to the purely implicit case such as in [53] and [54], all of which are extensions of the original works in implicit error estimation in [55] and [32]. It is possible to differentiate between fully implicit, linearly-implicit (which includes Rosenbrock-type methods) and semi-implicit methods. The problem with fully implicit methods is the $n \times s$ system of non-linear equations that need to be solved for an n-dimensional system using a method with s stages. The next best thing is therefore to employ linearly implicit methods, which in some cases can be solved in an explicit manner, making it a much less laborious process than that of its purely implicit counterpart. Common global error estimation techniques makes use of two parallel integrations [56]. The most popular of these kinds of global error estimators, called Richardson's extrapolation, will be compared to the core developed global estimator in this chapter which is constructed by integrating the local error estimate as done in [57].

The main contribution of this chapter is to formulate an accurate expression for the local and global error estimates of a second-order accurate linearly implicit RK method. The techniques used in this process can be applied more generally to other implicit RK methods.

Consider the purely deterministic component of the SDE in(A.21) according to the initial value problem (IVP)

$$\frac{dx(t)}{dt} = f(x(t), t), \quad x(0) = x_0 \in \mathbb{R}^n, t \geq 0. \tag{3.1}$$

Where no analytic solutions exist, numerical solutions of IVPs can be employed based on either multistep or RK methods. Linear multistep methods are, however, not considered in this thesis due to the additional computational resources required to store past information.

3.1 Runge-Kutta Methods

Approximate solutions of ODEs based on RK methods are classified as either implicit or explicit. This can further be divided into a range of sub-categories such as Guass-Legendre, Radau and Lobatto; all of which exhibit different stability and accuracy properties [32].

3.1.1 Explicit RK

For an integration step size of δ, the general form of the explicit RK solution to (3.1) is given by

$$\begin{aligned} x_{i+1} &= x_i + \delta \sum_{j=1}^{s} \vartheta_j k_j, \\ k_j &= f\big(x_i + \delta \sum_{l=1}^{j-1} a_{jl} k_l, t_i + \delta c_j\big), \quad j = 1, 2, \ldots, s. \end{aligned} \tag{3.2}$$

The a_{jl}, ϕ_j and c_j coefficients are given by the generic Butcher tableau shown in Table 3.1.1.

0	0	0	0	...	0
c_2	a_{21}	0	0	...	0
c_3	a_{31}	a_{32}	0	...	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\ddots$	$\vdots$
c_s	a_{s1}	a_{s2}	...	$a_{s(s-1)}$	0
	ϑ_1	ϑ_2	...	ϑ_s	

Table 3.1.1: Generic Butcher tableau for explicit RK methods.

c_1	a_{11}	a_{12}	...	a_{1s}
c_2	a_{21}	a_{22}	...	a_{2s}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	a_{s1}	a_{s2}	...	a_{ss}
	ϑ_1	ϑ_2	...	ϑ_s

Table 3.1.2: Generic Butcher tableau for implicit RK methods.

Examples of explicit RK methods are Heun's method (Table 3.1.3), also referred to as the explicit trapezoidal method, and the explicit midpoint method (Table 3.1.4). Substituting the coefficients of the Butcher tableau in Table 3.1.1 into (3.2) it can be seen that such methods only require information about the current state of the system and can therefore be solved in a fairly straight forward manner. The problem with explicit methods, however, is that they do not exhibit strong stability properties and thus do not work well for stiff systems or

relatively big step sizes. Note that for explicit RK methods the step size is limited by the absolute value of the inverse of the system's largest eigenvalue. A step size greater than this will lead unstable behaviour.

0	0	0
1	1	0
	1/2	1/2

Table 3.1.3: Heun's method.

0	0	0
1/2	1/2	0
	0	1

Table 3.1.4: Explicit midpoint method.

3.1.2 Implicit RK

Implicit RK methods are proposed in the case of stiff systems due to their stronger stability properties. They are obtained according to the generic Butcher tableau in Table 3.1.2 and by replacing k_j in (3.2) with

$$k_j = f\Big(x_i + \delta \sum_{l=1}^{s} a_{jl} k_l, t_i + \delta c_j\Big), \quad j = 1, 2, \ldots, s. \tag{3.3}$$

An example is the implicit trapezoidal rule, also referred to as the Crank-Nicolson method, used frequently in this thesis, and whose Butcher tableau is shown in Table 3.1.5.

0	0	0
1	1/2	1/2
	1/2	1/2

Table 3.1.5: Implicit trapezoidal method.

1/2	1/2
	1

Table 3.1.6: Implicit midpoint method.

Implicit methods are typically implemented via Newton iterations where a linear system of sn $\times$ sn equations are solved at each iteration until convergence. See [58] for details on implementing implicit solutions.

Gauss-Legendre RK methods

Gauss-Legendre methods are a family of implicit collocation methods based on Gaussian quadrature where c_i for $i = 1, \ldots, s$ are the zeros of the Legendre polynomial $\frac{d^s}{dx^s}\left(x^s(x-1)^s\right)$ of order 2s. An example is the implicit midpoint method whose Butcher tableau is shown in Table 3.1.6. This method is also an example of a diagonally implicit RK (DIRK) method with generic Butcher tableau as shown in Table 3.1.7. DIRK methods reduce the number of linear equations that need to be solved at each Newton iteration from sn $\times$ sn (in the fully implicit case) to sn $\times$ n [59], [60].

3.1.3 Semi-Implicit RK

Although DIRK methods reduce the number of linear equations that need to be solved during implementation, the multiple Newton iterations are still a heavy computational burden. The class of semi-implicit RK (SIRK),

c_1	a_{11}	0	0	$\dots$	0
c_2	a_{21}	a_{22}	0	$\dots$	0
c_3	a_{31}	a_{32}	a_{33}	$\ddots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	0
c_s	a_{s1}	a_{s2}	a_{s3}	$\dots$	a_{ss}
	ϑ_1	ϑ_2	ϑ_3	$\dots$	ϑ_s

Table 3.1.7: Generic Butcher tableau for DIRK methods.

or linearly implicit RK methods, are solved in an explicit manner, requiring very little effort and has the general form

$$x_{i+1} = x_i + \sum_{j=1}^{s} \mathbb{a}_j k_j \tag{3.4}$$
$$k_j = \delta\Big(I - \gamma_j \delta J(x_i + \sum_{l=1}^{j-1} \mathbb{c}_{jl})\Big)^{-1} f\Big(x_i + \sum_{l=1}^{j-1} \mathbb{b}_{jl} k_l\Big), \quad j = 1, 2, \dots, s, \quad \mathbb{c}_{11} = \mathbb{b}_{11} = 0,$$

where $\mathbb{a}_j$, $\mathbb{b}_{jl}$, $\mathbb{c}_{jl}$ and γ_j are real coefficients obtained by Taylor expanding x_{i+1} and fitting it to (3.4) as shown in [61]. A single stage, A-stable, second-order accurate SIRK method is given in [62] and [31] as

$$x_{i+1} = x_i + \delta\Big(I - \delta\gamma_1 J_i\Big)^{-1} f(x_i), \tag{3.5}$$

where $J_i = \frac{\partial f(x)}{\partial x}|_{x=x_i}$ is the drift Jacobian and $\gamma_1 = 0.5$. Since this is only a two stage method, it can be derived directly by linearising the implicit trapezoidal method; and in fact has the same stability and accuracy properties as the implicit trapezoidal method [62], but requires less effort to solve. A 3 stage, third-order accurate, A-stable SIRK method presented in [61] is given as

$$\begin{aligned} x_{i+1} &= x_i + \mathbb{a}_1 k_1 + \mathbb{a}_2 k_2 + \mathbb{a}_3 k_3 \\ k_1 &= \delta\Big(I - \delta\gamma_1 J_i\Big)^{-1} f(x_i) \\ k_2 &= \delta\Big(I - \delta\gamma_1 J_i\Big)^{-1} f(x_i + \mathbb{b}_{21} k_1) \\ k_3 &= \delta\Big(I - \delta\gamma_1 J_i\Big)^{-1} (J_i)(\mathbb{b}_{31} k_1 + \mathbb{b}_{32} k_2), \end{aligned} \tag{3.6}$$

with coefficients $\gamma_1 = 0.43$, $\mathbb{b}_{21} = 0.75$, $\mathbb{b}_{31} = -0.27$, $\mathbb{b}_{32} = -0.11$, $\mathbb{a}_1 = \frac{16}{27}$, $\mathbb{a}_2 = \frac{11}{27}$ and $\mathbb{a}_3 = 1$. This method requires only 1 Jacobian evaluation (at a convenient point x_i), 1 n $\times$ n inversion matrix, and only 2 drift function evaluations. Such cheap semi-implicit methods in (3.5) and (3.6) with good stability properties are desirable over fully implicit or DIRK methods that require multiple Jacobian evaluations, multiple solutions of

linear systems equations (and hence multiple inversion operations), multiple function evaluations and multiple Newton iterations. In this thesis, all linearly implicit methods are desirable (fourth order schemes and higher as well), with a general requirement that $\sum_{l=1}^{j-1} \mathbb{C}_{jl} = 0$ for $j = 1, 2, \ldots, s$, such that the Jacobian used in the scheme is evaluated once only at x_i. One of the reasons for this is to promote efficiency and will become more evident when filters are visited in Chapter 4.

3.1.4 Embedded RK

Embedded RK methods are made up of two RK methods that are used to calculate the primary solution in conjunction with its local truncation error estimate. With reference to Table 3.1.8, the primary method of order p and the embedded method of order p-1 have the same a_{jl} and c_j coefficients, but different ϑ_j coefficients. This method of error estimation is known as local extrapolation. The solution according to the embedded method of order p-1 is given by $x^*_{i+1} = x_i + \delta \sum_{j=1}^{s} \vartheta^*_j k_j$. The local truncation error [32] is therefore written as

$$l_{i+1} = x_{i+1} - x^*_{i+1} = \delta \sum_{j=1}^{s} (\vartheta_j - \vartheta^*_j) k_j. \tag{3.7}$$

c_1	a_{11}	a_{12}	$\ldots$	a_{1s}
c_2	a_{21}	a_{22}	$\ldots$	a_{2s}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	a_{s1}	a_{s2}	$\ldots$	a_{ss}
	ϑ_1	ϑ_2	$\ldots$	ϑ_s
	ϑ^*_1	ϑ^*_2	$\ldots$	ϑ^*_s

Table 3.1.8: Generic Butcher tableau for embedded RK methods

3.2 Numerical Error Estimation

The ODE in (3.1) has discrete approximation step

$$x_{i+1} = x_i + \delta \Phi(x_i, t_i), \tag{3.8}$$

where Φ is the increment function and is a characteristic of the numerical method employed. Its local error is defined as the error between two consecutive steps, so assuming no error in the previous steps, it follows that

$$l_{i+1} = x(t_{i+1}) - (x(t_i) + \delta \Phi(x(t_i), t_i)), \tag{3.9}$$

where $x(t_{i+1})$ is the true (but typically unknown) solution of the ODE in (3.1). The global error is the total

accumulated error from time $t = 0$ to $t = t_{i+1}$ or

$$e_{i+1} = x(t_{i+1}) - x_{i+1}|_{x_0 = x(t_0)}. \tag{3.10}$$

In this thesis, the first step in obtaining the global error estimate is to obtain an estimate of the local error in (3.9). This is discussed in the next few sections.

3.2.1 Local Extrapolation

Local extrapolation, referred to in Section 3.1.4, is the most widely used method in *a posteriori* local error estimation and is documented in [63] and [64]. It consists of using two methods of order p and p^*, with $p^* < p$, and solutions x^*_{i+1} and x_{i+1}. The local error estimate of solution x_{i+1} is

$$l_{i+1} = x_{i+1} - x^*_{i+1}|_{x_i}. \tag{3.11}$$

The solution typically proceeds with the method of order p since it is already calculated. Local extrapolation is applicable to both explicit and implicit methods but for stiff problems it has been suggested in [55] to include a damping factor in the estimate to correct for large δ, thereby preventing over estimation of the error. This gives a local error of

$$\bar{l}_{i+1} = (I - \delta\gamma J_i)^{-1} l_{i+1}, \tag{3.12}$$

where γ is a characteristic of the implicit method ($\gamma = 0.5$ for the implicit trapezoidal method) and $J_i = \frac{\partial f(x)}{\partial x}|_{x=x_i}$ is the Jacobian. This method of local error estimation can be extended to obtain a more accurate estimate of the error for implicit methods. This is discussed in the next section.

3.2.2 An Implicit Local Error Estimate

The aim of this section is to establish a reasonable local error estimate associated with the second-order accurate RK method according to the approach detailed in [54]. This approach to local error estimation for implicit methods was originally applied to a fifth-order accurate method in [54]. Here it is applied to the second-order accurate implicit trapezoidal method whose Butcher tableau is shown in Table 3.2.1 and 3.2.2.

c_1	a_{11}	a_{12}
c_2	a_{21}	a_{22}
	ϑ_1	ϑ_2
	ϑ^*_1	ϑ^*_2

Table 3.2.1: General 2-stage RK method.

0	0	0
1	1/2	1/2
	1/2	1/2
	ϑ^*_1	ϑ^*_2

Table 3.2.2: Implicit trapezoidal method.

From these tables, the implicit trapezoidal method has solution

$$x_{i+1} = x_i + \delta\Big(\sum_{j=1}^{s} \vartheta_j k_j\Big) \tag{3.13}$$

and embedded solution

$$x^*_{i+1} = x_i + \delta\Big(\sum_{j=1}^{s} \vartheta^*_j k_j\Big). \tag{3.14}$$

The implicit trapezoidal method is chosen as one of the primary solutions to many of the upcoming problems in this thesis because of its simplicity and cheap implementation when considering its Taylor expanded form which results in a linearly implicit method developed in [62] and [31]. The method in [54] proceeds by introducing additional parameters, ϑ^*_0 and γ, into the embedded method's reference solution (the first iteration of the implicit method's solution), see [32] for details, and is given as

$$x^*_{i+1} = x_i + \delta\Big(\sum_{j=1}^{s} \vartheta^*_j k_j + \vartheta^*_0 f(x_i) + \gamma f(x^*_{i+1})\Big), \tag{3.15}$$

where ϑ^*_0 is a free parameter chosen to satisfy the order conditions and control the magnitude of the error estimate, or more specifically, to prevent under-estimation of the error, and γ is a property of the method. For the case of the implicit trapezoidal method, the embedded method (of order p-1) should be first-order. The values of ϑ^*_1 and ϑ^*_2 are therefore chosen to satisfy a first-order accurate method. The new local error using the amended embedded reference formula, (3.15), and including the damping factor introduced in (3.12) is

$$\bar{\bar{l}}_{i+1} = (I - \delta\gamma J_i)^{-1}\big(l_{i+1} - \delta\vartheta^*_0 f(x_i) - \delta\gamma f(x^*_{i+1})\big), \tag{3.16}$$

where l_{i+1}, x_{i+1} and x^*_{i+1} are given in (3.11), (3.13) and (3.14). It is noticed, however, that (3.16) has a term evaluated at the following time instance, $f(x^*_{i+1})$. Taking a step back and first linearising (3.15) around x_i gives

$$x^*_{i+1} = x_i + \delta\Big(I - \delta\gamma J_i\Big)^{-1}\Big(\sum_{i=1}^{s} \vartheta^*_i k_i + \vartheta^*_0 f(x_i) + \gamma f(x_i)\Big). \tag{3.17}$$

See Appendix B.1 for the complete linearisation that results in (3.17). The local error estimate when using the linearised reference formula in (3.17) instead of that in (3.15) can therefore be written as

$$\bar{\bar{l}}_{i+1} = (I - \delta\gamma J_i)^{-2}\Big(\delta(I - \delta\gamma J_i)\sum_{j=1}^{s} \vartheta_j k_j - \delta\sum_{j=1}^{s} \vartheta^*_j k_j - \delta\vartheta^*_0 f(x_i) - \delta\gamma f(x_i)\Big). \tag{3.18}$$

Note that the damping factor is again included in the expression for the local error above. Circling back to the approach outlined in [54], the following condition for a first-order accurate method needs to be satisfied in

order to determine the ϑ_o^* parameter:

$$C\vartheta^* = [1 - \vartheta_o^* - \gamma, \theta - \gamma]^T, \tag{3.19}$$

where $\vartheta^* = [\vartheta_1^*, \vartheta_2^*]^T$, $C \in \mathbb{R}^{s\times s}$ with $C_{jl} = c_l^{j-1}$ and $\theta \in \mathbb{R}^+ \neq \frac{1}{2}$. Here $\gamma = 0.5$ for the implicit trapezoidal method. The local error in this case, from (3.18), is

$$\bar{\bar{l}}_{i+1} = (I - \gamma\delta J_i)^{-2}\delta\Big(\sum_{j=1}^{s}\big(\vartheta_j(I - \delta\gamma J_i) - \vartheta_j^*\big)k_j - \vartheta_o^* f(x_i) - \gamma f(x_i)\Big), \tag{3.20}$$

where $J_i = \frac{\partial}{\partial x} f(x)|_{x=x_i}$ is the Jacobian. The parameters are thus chosen as $\vartheta_1^* = 0$ and $\vartheta_2^* = 1 - \vartheta_o^* - \gamma$ to satisfy (3.19). An appropriate value of ϑ_o^* is searched for by considering the linear test equation $y = e^z$, where $z = \delta\eta$ for $\eta \in \mathbb{C}^-$, and analysing the magnitude of the true relative local error $\left|\frac{x_{i+1} - x_{i+1}^*}{x_i}\right| = |\mathcal{R}(z) - e^z|$ on two domains, $\mathcal{A}(\rho)$ and $\mathcal{B}(\rho)$. Here $\mathcal{A}(\rho)$ represents the method's asymptotic region which constitutes the central portion of the method's order star; $\mathcal{B}(\rho)$ represents the region where accurate results can be obtained which constitutes a large region of $\mathbb{C}^-$; and $\mathcal{R}(z) = \frac{1+z/2}{1-z/2}$ is the stability function of the implicit trapezoidal rule [58] or its linearly implicit version (the Taylor-Heun method). The domains are chosen as $\mathcal{A}(\pi/3)$ and $\mathcal{B}(\pi/3)$ such that the local error is not heavily over-estimated and are described as

$$\mathcal{A}(\rho) = \{z \in \mathbb{C} : |z| \leq \rho\}, \tag{3.21}$$

$$\mathcal{B}(\rho) = a \leq (\rho - d)(\rho + d)/\rho, \tag{3.22}$$

where $z = a + jd$. See Fig. 3.2.1 for the respective domain plots and the method's order star (region of relative stability). With this a suitable value of ϑ_o^* can be searched for such that the maximum possible relative local error (the upper bound on the local error) is obtained within the specified bounds $\mathcal{A}(\pi/3)$ and $\mathcal{B}(\pi/3)$. The maximum local error occurs on the boundary of $\mathcal{B}(\pi/3)$. Searching along this boundary gives a maximum absolute value of $\max|\mathcal{R}(z) - \exp(z)| = 0.3482$. Therefore ϑ_o^* is chosen such that the maximum of the estimated local error is roughly equal to the maximum of true local error within the bounded domains, i.e., $\max|\mathcal{R}(z) - \mathcal{R}^*(z)| = \max|\mathcal{R}(z) - \exp(z)|$. According to [54], $\max|\mathcal{R}(z) - \mathcal{R}^*(z)| = \frac{\vartheta_o^*}{\gamma}$, which holds for all implicit RK methods and hence $\frac{\vartheta_o^*}{\gamma} = 0.3482$. With $\gamma = 0.5$ for the implicit trapezoidal method, it follows that $\vartheta_o^* = 0.17$.

3.2.3 An Implicit Local Error Estimate for the Linearly Implicit Second-Order Accurate Scheme

According to Table 3.2.2 the implicit trapezoidal method is

$$x_{i+1} = x_i + \delta(\frac{1}{2}f(x_i, t_i) + \frac{1}{2}f(x_{i+1}, t_{i+1})). \tag{3.23}$$

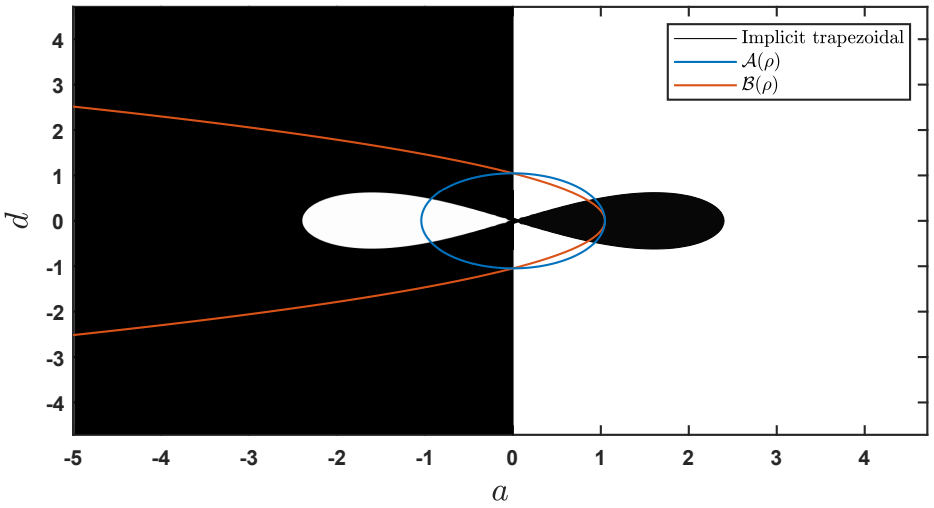


Figure 3.2.1: $\mathcal{A}(\pi/3)$, $\mathcal{B}(\pi/3)$ and the implicit trapezoidal order star (shaded region) with $z = a + jd$.

A Taylor expansion of (3.23) leads to the linearly implicit Taylor-Heun (TH) scheme in (3.5):

$$x_{i+1} = x_i + \left(I - J_i\frac{\delta}{2}\right)^{-1} f(x_i)\delta. \tag{3.24}$$

The embedded method of order p-1 (from Table 3.2.2) is given by

$$x^*_{i+1} = x_i + \delta(1 - \vartheta^*_o - \gamma)f(x_{i+1}, t_{i+1}), \tag{3.25}$$

which after linearising gives

$$x^*_{i+1} = x_i + \delta(1 - \vartheta^*_o - \gamma)\left(I - J_i(1 - \vartheta^*_o - \gamma)\delta\right)^{-1} f(x_i). \tag{3.26}$$

According to (3.20) the local error is therefore

$$\begin{aligned} l_{i+1} &= \left(I - J_i\frac{\delta}{2}\right)^{-2}\left[\delta f(x_i) - \delta(1 - \vartheta^*_o - \gamma)\left(I - J_i(1 - \vartheta^*_o - \gamma)\delta\right)^{-1} f(x_i) - \delta\vartheta^*_o f(x_i) - \delta\gamma f(x_i)\right] \\ &= \underbrace{\left(I - J_i\frac{\delta}{2}\right)^{-2}\left[I - (1 - \vartheta^*_o - \gamma)\left(I - J_i(1 - \vartheta^*_o - \gamma)\delta\right)^{-1} - \vartheta^*_o I - \gamma I\right]}_{C_i}\delta f(x_i), \end{aligned} \tag{3.27}$$

where the terms in brackets are grouped together as $C_i \in \mathbb{R}^{n\times n}$, and the parameters, $\vartheta^*_o = 0.17$ and $\gamma = 0.5$, were calculated/given in Section 3.2.2.

3.2.4 From Local to Global Error Estimates

Having an estimate of the local error according to (3.27), with parameters $\vartheta^*_o = 0.17$ and $\gamma = 0.5$ for the implicit trapezoidal method, the global error can now be estimated. Research in global error estimation is pre-

dominantly aimed at obtaining a more accurate and efficient means of its computation. Work such as that in [65] discusses a cheap method for obtaining global error estimates and simply reduces to the sum of the local error estimates. This is often, however, an exaggerated over bound on the global error. A more suitable approach discussed in [57] begins with the perturbed system dynamics

$$\frac{dy(t)}{dt} = f(y(t), t) + r(t), \tag{3.28}$$

where $y(t)$ is a nearby solution to the true solution $x(t)$, $e(t) = y(t) - x(t)$ is the global error function and $\bar{r}(t)$ is the defect. The dynamic global error function is

$$\frac{de(t)}{dt} = A(t)e(t) + \bar{r}(t), \tag{3.29}$$

where $A(t) = \int_0^1 \frac{df(x(t)+se(t),t)}{dt} ds$ is motivated in [57]. This term is equivalent to $A(t) = J(t) + O(e(t))$, where $J(t) = \frac{df(x(t),t)}{dt}$. Higher-order terms can be absorbed into the defect, so without a loss of generality (3.29) can therefore be written as

$$\frac{de(t)}{dt} = J(t)e(t) + r(t). \tag{3.30}$$

For a well conditioned system with $\omega < 0$, $\mu(J(t)) < \omega$ (where $\mu(J(t))$ is the logarithmic norm of matrix $J(t)$) and $||r(t)|| < v \in \mathbb{R}^+$. It follows that

$$||e(t)|| \leq \frac{v}{\omega}(\exp(\omega t) - 1), \tag{3.31}$$

which converges as $t \to \infty$. For a system that is not well conditioned, the stepwise frozen version of (3.30),

$$\frac{de(t_i)}{dt} = \frac{\partial f(x(t_i), t_i)}{\partial x} e(t_i) + r(t_{i+1/2}), \tag{3.32}$$

must be integrated over δ to obtain global error estimates at each time step. In this thesis, the defect $r(t_{i+1/2})$ is set to the local error estimate rate evaluated at the midpoint between two successive integration steps. Using the implicit midpoint method as suggested in [57] the solution of the global error estimate in (3.32) is

$$\begin{aligned} e_{i+1} &= \big(I - \delta J(x(t_i))/2\big)^{-1}\big(e(t_i) + \delta J(x(t_i))e(t_i)/2 + \delta r(t_{i+1/2})\big) \\ &= \big(I - \delta J(x(t_i))/2\big)^{-1}\big(e(t_i) + \delta J(x(t_i))e(t_i)/2 + l_{i+1/2}\big). \end{aligned} \tag{3.33}$$

The local error in 3.33, $l_{i+1/2}$, is simplified to $\frac{1}{2}l_{i+1} + \frac{1}{2}l_i$ such that

$$\begin{aligned} e_{i+1} &= \big(I - \delta J_i/2\big)^{-1}\Big(e(t_i) + \delta J_i e(t_i)/2 + \delta r(t_{i+1/2})\Big) \\ &= \big(I - \delta J_i/2\big)^{-1}\Big(e(t_i) + \delta J_i e(t_i)/2 + \frac{1}{2}l_i + \frac{1}{2}l_{i+1}\Big). \end{aligned} \tag{3.34}$$

3.2.5 Richardson's Extrapolation

In order to compare the above method to another popular method of global error estimation found in the literature, consider the method referred to as Richardson's extrapolation. This method makes use of two state solutions evaluated at the same time instance but with one of the solutions computed using double the integration step size. From [66] the first-order explicit Euler solution can be written in the form

$$x(t_i) - x_i = \delta D(t_i) + O(\delta^2), \tag{3.35}$$

where $D(t)$ satisfies

$$D'(t_i) = J_i D(t_i) + \frac{1}{2}x''(t_i). \tag{3.36}$$

According to Richardson's error estimate, the global error estimate for the Euler scheme in (3.35) can be written as

$$x(t_i) - x_i = x_i - x_i^{2\delta}, \tag{3.37}$$

where $x_i^{2\delta}$ is the numerical solution using the same method as x_i but with double the time step. It can be shown from [66] that (3.36) holds for RK methods of order p such that

$$x(t_i) - x_i = \delta^p D(t_i) + O(\delta^{p+1}). \tag{3.38}$$

So for a second-order RK method the global error at time t_i is

$$e_i = x(t_i) - x_i = \frac{1}{3}(x_i - x_i^{2\delta}). \tag{3.39}$$

3.3 Case Studies

Two examples are presented in the following section to compare error estimates using Richardson's extrapolation with estimates constructed according to (3.27) and (3.34), calculated explicitly for the second-order accurate linearly implicit trapezoidal method. The true state variables in each case are simulated using the linearly implicit trapezoidal method with very small time steps (alternatively, a first-order Euler approximation will suffice for this purpose). For notational simplicity, explicit time and state dependence are dropped in model equations.

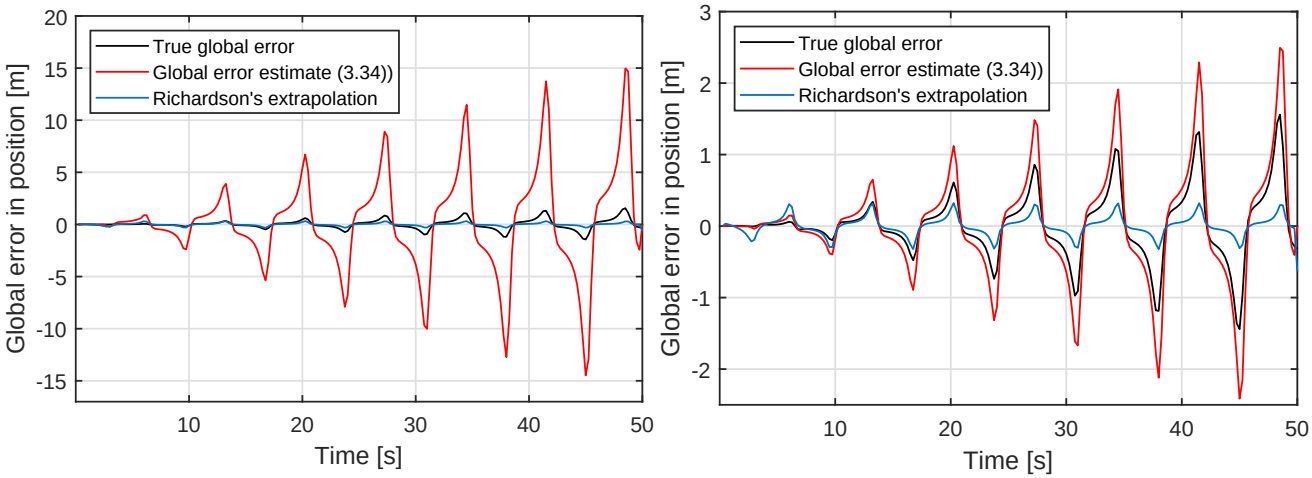
3.3.1 Example 1: The Van der Pol Oscillator

The Van der Pol oscillator is a common benchmark problem described by a stiff non-linear differential equation [31] that exhibits limit cycles for $\varepsilon > 0$ and damped oscillations for $\varepsilon < 0$. Its state vector comprises its

position, x_1 [m], and velocity, x_2 [ms^{-1}]. The system model is given by

$$d\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ \varepsilon(1 - x_1^2)x_2 - x_1 \end{pmatrix} dt\,, \quad x_0 = (0.5\ 0.5)^T, \tag{3.40}$$

for $\varepsilon = 1.4$. Shown in Fig. 3.3.1a are the plots comparing global error estimates of the model's TH solution in (3.34) and Richardson's extrapolation in (3.39). A factor of δJ_i is heuristically appended to the local error formulation in (3.27). The resulting global error plots are shown in Fig. 3.3.1b. Notice an improvement in the global error estimates of (3.34) when appending the correction factor. The correction factor is motivated by noting that the local error in (3.27) is calculated using a lower order method, and although this is done in the literature, results in an error of lower order. This is explained further in what follows.



(a) $\delta = 0.25$ s, without the correction factor.

(b) $\delta = 0.25$ s, with the correction factor.

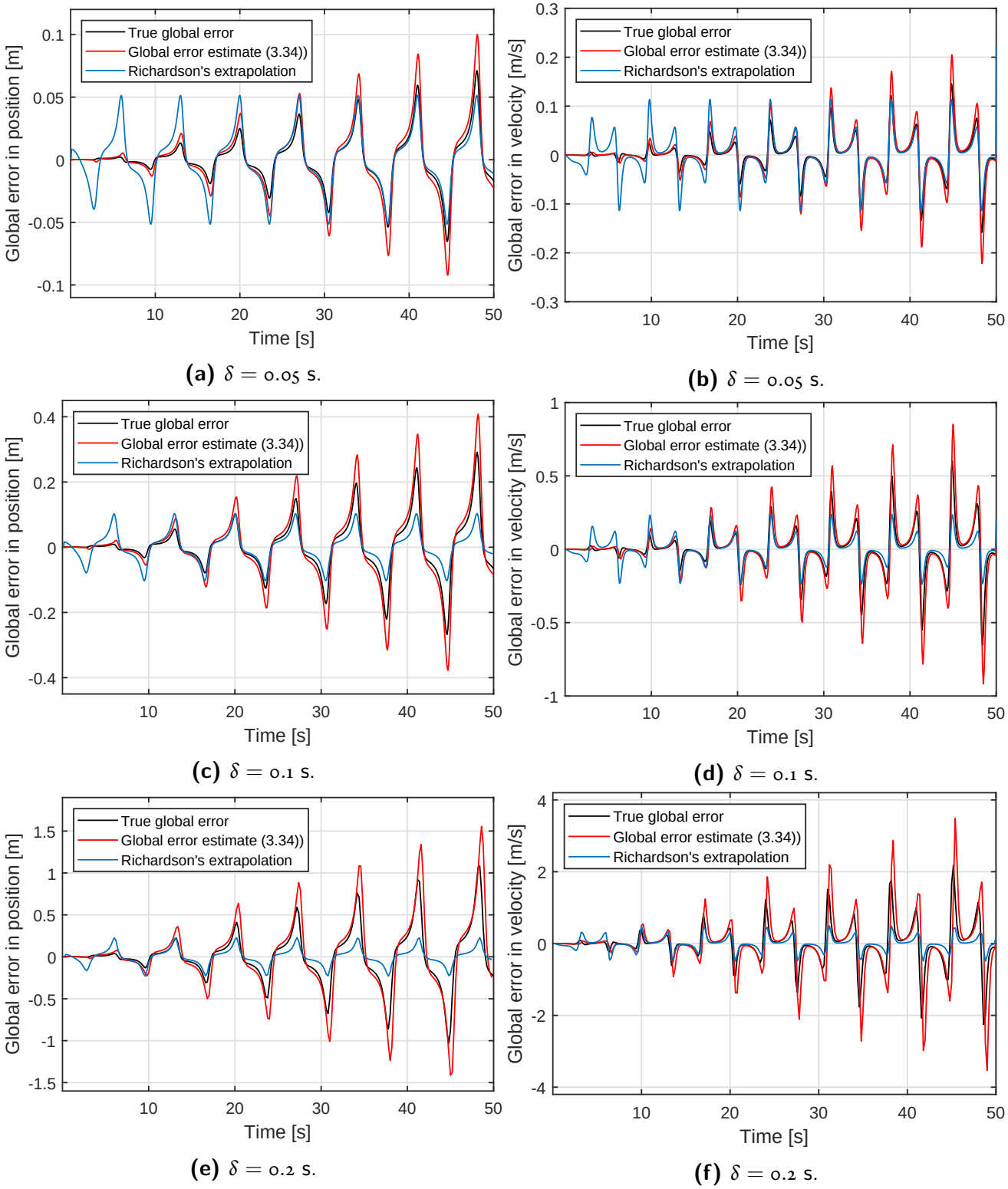
Figure 3.3.1: Example 1: Global error in position, with and without the correction factor.

Correction Factor

Local extrapolation uses a method of order p and p-1 to calculate the error, while the simulation proceeds with a method of order p. This results in a local error of $\mathcal{O}(h^p)$, an order lower than it should be which is $\mathcal{O}(h^{p+1})$. Consider the simple explicit case. The local truncation error of the first-order Euler scheme is $\propto \frac{\delta^2}{2} f'(x(t), t)$, while that of a second-order Euler scheme is $\propto \frac{\delta^3}{3!} f''(x(t), t)$ (and alike for higher-order methods), where $f'(x(t), t)$ and $f''(x(t), t)$ are the first and second derivatives with respect to the state vector. The approximate difference between the first- and second-order method is the factor of $\delta J(x(t), t)$. Appending this factor to the local error estimate in (3.27) gives

$$l_{i+1} = \delta^2 J_i C_i f(x_i). \tag{3.41}$$

Although this is a crude approximation, it suffices to fix the order of magnitude of the error estimates as shown in Fig. 3.3.2. To further verify this, another example is presented.



(a) $\delta = 0.05$ s.

(b) $\delta = 0.05$ s.

(c) $\delta = 0.1$ s.

(d) $\delta = 0.1$ s.

(e) $\delta = 0.2$ s.

(f) $\delta = 0.2$ s.

Figure 3.3.2: Example 1: Global error in position and velocity for various values of δ. Includes the correction factor.

3.3.2 Example 2: The Vertically Falling Body Problem

The vertically falling body is a highly non-linear problem characterised by two opposing forces, drag and gravity [31]. The system model is given by

$$d\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -x_2 \\ \gamma_g - c_b e^{-\lambda_\rho x_1} x_2^2 \end{pmatrix} dt\,, \quad x_0 = (180000\ 4000)^T, \tag{3.42}$$

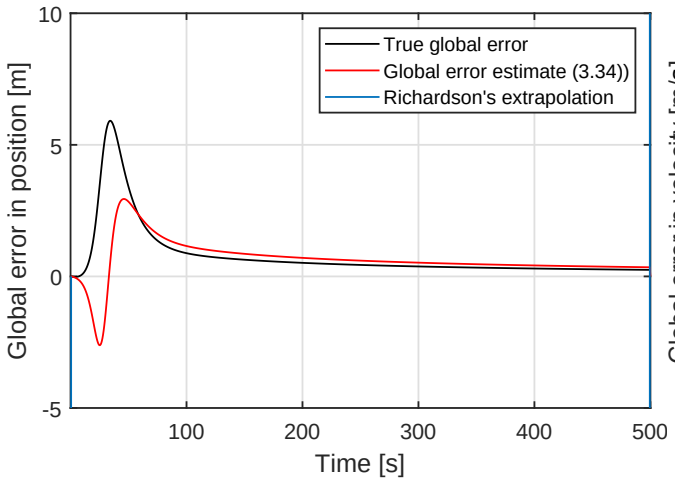
where x_1 [m] is the position of the body, x_2 [ms^{-1}] is its velocity, $\gamma_g = 9.8$ [ms^{-2}] is the gravitational acceleration constant, $c_b = 10^{-3}$ is a ballistic constant and $\lambda_\rho = 5 \times 10^{-5}$ is an atmospheric density constant. The correction factor is applied to the local estimates according to (3.41) and results in the global error plots in Fig. 3.4.1.

3.3.3 Discussion

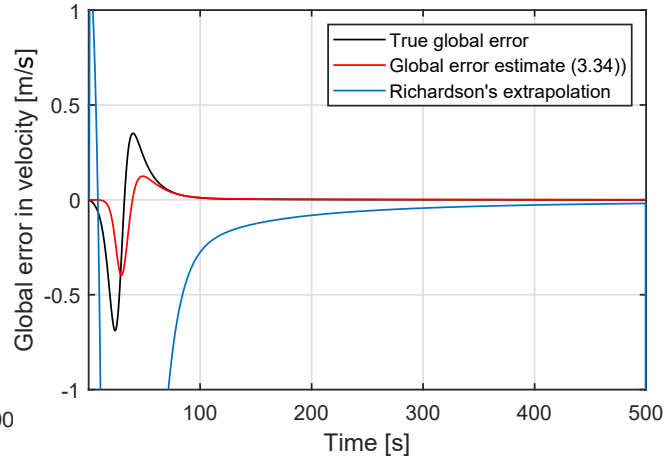
It is evident in all two examples that the global error estimates obtained using (3.34) are closer to the true errors and demonstrate better convergence behaviour compared to the global error estimates obtained via Richardson's extrapolation. The global error estimate in (3.34) is therefore a good approximation to the true global error of the system in each case. From this, it can be deduced that the local error estimate in (3.41), which includes the correction factor, is a good approximation to the true local error.

3.4 Conclusion

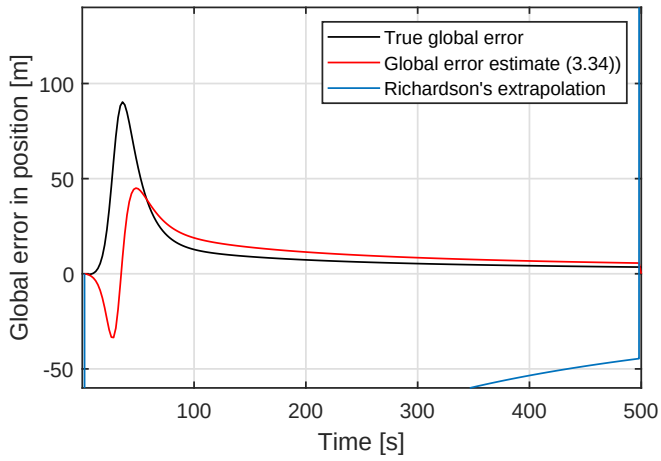
In this chapter, the local and global error estimates of the second-order accurate linearly implicit TH scheme were developed. The error estimation techniques used in this chapter are, however, not limited solely to the TH scheme and can be extended to higher-order implicit methods. The need for a good estimate of the numerical integration error will become evident when the state estimation problem with a stochastic simulation error term in the SDE is visited in Chapter 5.



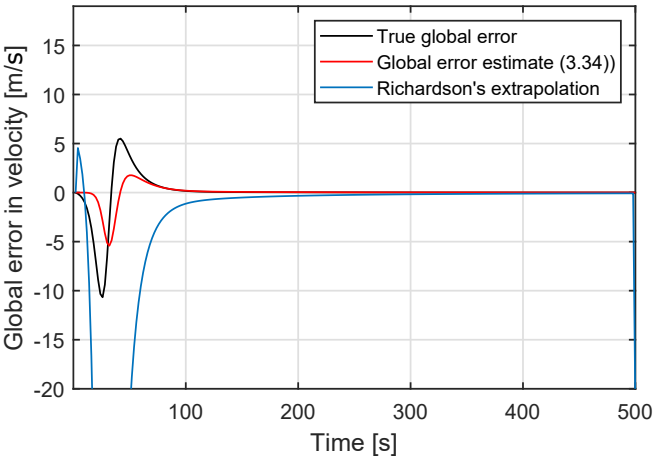
(a) $\delta = 0.5$ s.



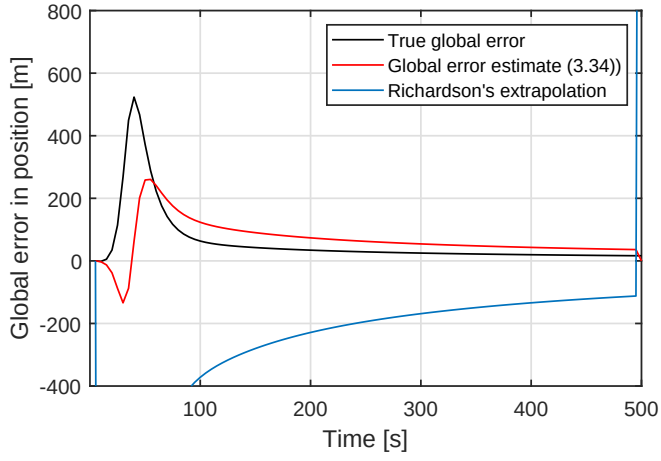
(b) $\delta = 0.5$ s.



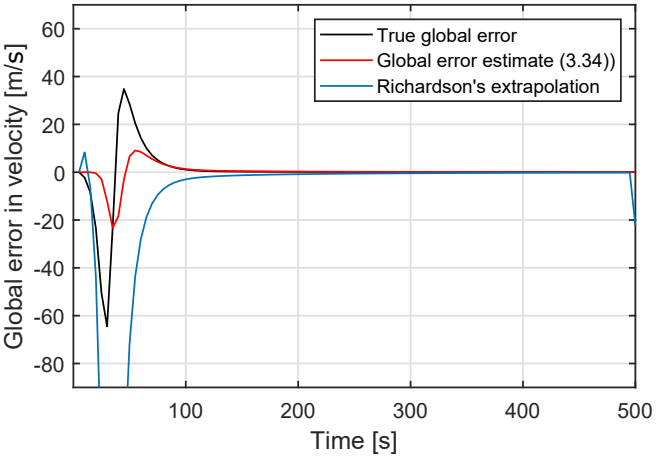
(c) $\delta = 2$ s.



(d) $\delta = 2$ s.



(e) $\delta = 5$ s.



(f) $\delta = 5$ s.

Figure 3.4.1: Example 2: Global error in position and velocity for various values of δ. Includes the correction factor.

4

Non-Linear Kalman Filters

In linear estimation theory, the Kalman-Bucy filter is the continuous-time variant of the discrete-time KF. Developed in the early 1960s, both the discrete-time KF and the Kalman-Bucy filter are the predecessors of a variety of different KFs. These include approximate filters for non-linear problems such as the EKF and UKF as well as their continuous-time and CD versions. Although the first-order EKF, or EKF for short, is a popular non-linear filter, an immediate drawback is that it relies on expressions for the system Jacobians. Analytic calculations of system Jacobians are in many cases not trivial (or grossly resource intensive) [8] and simplifications could potentially lead to weaker forms of stability [67]. The UKF, on the other hand, is a derivative-free approach to Kalman filtering [8] that has been shown in various applications with severe non-linearity to offer better accuracy in comparison to the EKF. Instead of Jacobians, it uses sets of unscented transforms (UTs) to propagate the filter moments. Although the manner in which the UKF and EKF compute their moments appear very different, it is shown in [68] that the UKF and second-order EKF effectively result in identical expressions for the mean with subtle differences between their state covariance matrices. The authors in [68] then develop a second-order EKF that uses the UTs to circumvent the need for a direct analytic calculation of the Jacobians and Hessians or their numerical equivalents. These Jacobians and Hessians are not, however, formed explicitly. One of the highlights of this chapter is therefore to extract these state derivative matrices explicitly. This is useful because work in this thesis is based on the CD-KF formulation. The Jacobians can be used in the propagation stage of the CD-KF formulation to employ single-step linearly implicit numerical integration methods that are less complex than standard implicit methods. Implicit methods are used because they are better at solving stiff problems. Efficient CD filtering of stiff systems are considered in [31] where the cheap second-order Taylor-Heun (TH) and the Modified Gauss-Legendre (MGL) schemes are formulated for the CD-EKF.

Furthermore, the MGL scheme offers symmetric and positive-definite (PD) representations of the state covariance matrices in the filter's state propagation stage where it is applied [31]. This representation results in better numerical stability of the filter due to less accumulated round-off error associated with limited computational precision. This is not the first account of imposing symmetry in order to improve the numerical stability of a filter. In much earlier years a more robust version of the state covariance measurement update equation of the discrete-time EKF was formulated [69], [70]. The CD-UKF, on the other hand, does not have symmetric solutions of the state covariance propagation and measurement update stages and is therefore more prone to numerical instability. It is for this reason that the square root filter was initially developed [21] and is still relevant today, as in [46], for example. The square root filter propagates the square root of the state covariance matrix and offers better numerical stability on the grounds that less error is propagated when square rooted than in the standard case. The need for square root methods in this thesis is avoided by adopting symmetric EKF processes in the state covariance propagation and measurement update stages. This results in what will be referred to as a hybrid UKF-EKF filter, i.e., a combination of both EKF and UKF processes. The concept of hybrid filters has been published in [35]. The hybrid attribute of the filter developed in this chapter, however, differs from that in [35] by which parts of the CD-EKF and CD-UKF it uses.

The contribution of this chapter is therefore the new CD hybrid coupled Kalman filter (CD-HCKF) which details as follows. The UTs are used to extract a stand-alone representation of the system's state derivative matrices. These are in essence a linear combination of the Jacobian and higher-order state derivatives. These are called the drift and measurement higher-order linearised matrices (HOLMs); synonymous with the drift and measurement Jacobians. These HOLMs are formed because they can be done in a simple, almost black-box approach and because the UTs for state mean propagation have already been calculated in any case. The benefit of forming these HOLMs is that they can then be used in place of the Jacobians in parts of the standard CD-EKF. Parts of the CD-EKF formulation are used for two reasons. The first is to take advantage of the MGL scheme developed in [31] for state covariance propagation and the second is to take advantage of the Joseph stabilised version of the state covariance update equation of the CD-EKF. These are both symmetric and PD representations of the state covariance matrices and hence offer better numerical stability in comparison to UKF equivalent processes. Another benefit of forming the HOLMs is that the drift Jacobian can then be replace with the drift HOLM in linearly implicit integration schemes and used during state mean propagation to attain A-stable solutions of the differential equation. In doing so, computational efficiency is promoted since no additional function evaluations or coefficients need to be calculated iteratively for fully implicit RK methods.

4.1 The Continuous-Discrete Kalman Filter

There are three ways of looking at the filtering process: as a deterministic least-squares problem, as a way to maximise the likelihood of measurements occurring, or as a problem in Bayesian estimation. In Bayesian estimation, either the probabilistic minimum variance or statistical maximum likelihood estimate can be used. The general consensus is to adopt the minimum variance Bayesian framework [7], [27]. Here the solution of the

CD filtering problem lies in the solution of the conditional probability density function

$$p(x(t), t|Y(t_k)), \tag{4.1}$$

where $x(t) \in \mathbb{R}^{\mathrm{n}}$ is a Markovian state process and $Y(t_k) = \{y(t_0), \ldots, y(t_k)\}$ is a sequence of discrete observations with continuous-time equivalent $Y(t) = \{\int_{t_0}^{t} y(s)ds\}$. The optimal estimate, found by minimising the expected loss function $L(\tilde{x}(t))$, with $\tilde{x}(t) = x(t) - \hat{x}(t)$, is found to be the conditional mean given as

$$\begin{aligned}\hat{x}(t) &= \mathrm{E}\{x(t)|Y(t)\} \\ &= \underbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\ldots\int_{-\infty}^{\infty}}_{\mathrm{n\ times}} x(t)p(x(t)|Y(t_k))dx_1dx_2\ldots dx_{\mathrm{n}},\end{aligned} \tag{4.2}$$

and its conditional covariance matrix

$$\hat{P} = \mathrm{E}\{(x(t) - \hat{x}(t))(x(t) - \hat{x}(t))^T|Y(t_k)\}. \tag{4.3}$$

The density function in (4.1) is Gaussian and characterised solely by its mean and covariance only in the linear case. The corresponding state estimator is therefore optimal only in the linear case. In the continuous-time, non-linear case the system is governed by the SDE of *Itô*-type in (A.21) and is embedded in the filter by assuming a constant or simply time-varying diffusion coefficient such that

$$dx(t) = f(x(t), t)dt + q(t)dw(t), \tag{4.4}$$

and its non-linear discrete-time output equation at each observation instance as

$$y(t_k) = \Upsilon(x(t_k), t_k) + u_k, \tag{4.5}$$

where $f(x(t), t) \in \mathbb{R}^n$ is the drift function of the state, $w(t) \in \mathbb{R}^{\mathrm{d_2}\times 1}$ is a Wiener process, $y(t_k) \in \mathbb{R}^m$ is a vector of outputs, $\Upsilon(x(t_k), t_k)$ is the measurement function and $u_k \in \mathbb{R}^m$ is the white measurement noise with covariance matrix $R_k \in \mathbb{R}^{\mathrm{m}\times\mathrm{m}}$. The diffusion matrix (rate of variance) is expressed as $Q_c(t) = q(t)Q(t)q(t)^T$ with $q(t) \in \mathbb{R}^{\mathrm{n}\times\mathrm{d_2}}$ being the diffusion coefficient that scales the noise. The variance of the Wiener increment can be written as $E\{dw(t)dw^T(t)\} = Q(t)dt$, where $Q(t)$ $[\mathrm{s}^{-1}]$ is the noise spectral density taken as $I_{\mathrm{d_2}}$ (the identity matrix of dimension $\mathrm{d_2}$). Here $x(t_0) \sim \mathcal{N}(\bar{x}(t_0), P(t_0))$ for $t_0 > t$ and $x(t_0)$, $w(t)$ and u_k are independent.

Between observations, the propagation of the density in (4.1) can be obtained using various methods. This is either by direct discretisation of the *Itô*-Taylor expansion in (A.31) (this method is used in Chapter 5) or via Kolmogorov's forward equation given as

$$dp(x(t)|Y(t_k) = \mathscr{L}(p)dt, \tag{4.6}$$

where $\mathscr{L}$ is the forward operator given as

$$\mathscr{L}(\cdot) = -\sum_{i=1}^{n} \frac{\partial(\cdot f_i)}{\partial x_i} + \frac{1}{2}\sum_{i,j=1}^{n} \frac{\partial^2(\cdot (qQq^T)_{ij})}{\partial x_i \partial x_j}. \tag{4.7}$$

Another method is via *Itô*'s lemma in (A.26). Given $\phi(x(t))$, or ϕ in compact notation, a twice continuously differential scalar function (it suffices to show this for the scalar case) it follows then, and is shown in [7] that

$$d\phi = \dot{\phi}dt + \phi'^T dx + \frac{1}{2}\text{tr } q(t)Q(t)q^T(t)\phi'' dt, \tag{4.8}$$

where ϕ' is the gradient vector and ϕ'' the matrix of partial differential equations (see (A.25)). It follows then by substituting (4.4) into (4.8) and taking the expectation that

$$d\bar{\phi} = \text{E}\{\phi'^T f(x(t),t)\}dt + \frac{1}{2}\text{tr E}\{q(t)Q(t)q(t)^T\phi''\}dt. \tag{4.9}$$

The same result is obtained using Kolmogorov's forward equation in (4.6) and can similarly be obtained using a first-order [1] discretisation of the *Itô* Taylor expansion in (A.31). The propagation equations for the first and second moments, $\bar{x}(t) = \text{E}\{x(t)\}$ and $P(x(t),t) = \text{E}\{(x(t)-\bar{x}(t))(x(t)-\bar{x}(t))^T\} = \text{E}\{x(t)x^T(t)\} - \bar{x}(t)\bar{x}^T(t)$, follow by substituting $\phi = x(t)$ and $\phi = (x(t)-\bar{x}(t))(x(t)-\bar{x}(t))^T(t)$ respectively in (4.9). The resulting differential equations are

$$\frac{d\bar{x}(t)}{dt} = \bar{f}(x(t),t) \tag{4.10}$$

$$\begin{aligned}\frac{dP(x(t),t)}{dt} &= (\text{E}\{x(t)f^T(x(t),t)\} - \bar{x}(t)\bar{f}^T(x(t),t)) \\ &\quad + (\text{E}\{f(x(t),t)x^T(t)\} - \bar{f}(x(t),t)\bar{x}^T(t)) + \text{E}\{q(t)Q(t)q(t)^T\}.\end{aligned} \tag{4.11}$$

The measurement update equations are obtained from the conditional density function according to Bayes' theorem, whose details can be found in [7]. At an observation instance t_k it is given in [7] that

$$p(x(t),t_k|Y(t_k)) = \frac{P(y_k|x)P(x(t),t_k|Y(t_{k-1}))}{\int P(y_k|x)P(x,t_k|Y(t_{k-1}))dx}, \tag{4.12}$$

and $p(y_k|x) = \frac{1}{(2\pi)^{m/2}}|R_k|^{1/2}\exp\left(-\frac{1}{2}(y_k - \Upsilon(x(t),t_k))^T R_k^{-1}(y_k - \Upsilon(x(t),t_k))\right)$. The expectation of the scalar function in this case is found by integrating it over all possible states giving

[1] Higher-order discretisation schemes are discussed in Chapter 5.

$$\begin{aligned}\phi(x(t), t_{k|k}) &= \int \phi(x(t_{k|k}))p(x(t), t_k|Y(t_k))dx \\ &= \frac{\mathrm{E}\{\phi(x(t_{k|k}))p(y_{k-1}|x(t_k))\}}{\mathrm{E}\{p(y_{k-1}|x(t_k))\}},\end{aligned} \tag{4.13}$$

Again substituting $\phi = x(t)$ and $\phi = (x(t) - \bar{x}(t))(x(t) - \bar{x}(t))^T$ respectively in (4.13) gives

$$\bar{x}(t_{k|k}) = \frac{\mathrm{E}\{x(t_k)p(y_{k-1}|x(t_k))\}}{\mathrm{E}\{p(y_{k-1}|x(t_k))\}} \tag{4.14}$$

$$P(t_{k|k}) = \frac{\mathrm{E}\{x(t_k)x^T(t_k)p(y_{k-1}|x(t_k))\}}{\mathrm{E}\{p(y_{k-1}|x(t_k))\}} - \bar{x}(t_{k|k})\bar{x}^T(t_{k|k}). \tag{4.15}$$

4.2 The Optimal Linear Kalman Filter

The moment equations in (4.10) and (4.11) and the observation equations in (4.14) and (4.15) can be evaluated in the linear case. Consider the linear equivalent set of equations to (4.4) and (4.5):

$$\begin{aligned}dx(t) &= F(t)x(t) + q(t)dw(t) \\ y(t_k) &= G(t)x(t) + v_k,\end{aligned} \tag{4.16}$$

where $F(t)$ and $G(t)$ are the linear drift and measurement matrices. It is easily shown that the exact moment equations in this case are

$$\frac{d\bar{x}(t)}{dt} = F(t)\bar{x}(t) \tag{4.17}$$

$$\begin{aligned}\frac{dP(x(t), t)}{dt} &= (\mathrm{E}\{x(t)x^T(t)F^T(t)\} - \bar{x}(t)\bar{x}^T(t)F^T(t)) \\ &\quad + (\mathrm{E}\{F(t)x(t)x^T(t)\} - F(t)\bar{x}(t)\bar{x}^T(t)) + \mathrm{E}\{q(t)Q(t)q(t)^T\} \\ &= P(x(t), t)F^T(t) + F(t)P(x(t), t) + q(t)Q(t)q^T(t)\end{aligned} \tag{4.18}$$

The observation equations in (4.14) and (4.15) are derived in [7] as

$$\bar{x}(t_{k|k}) = \bar{x}(t_{k|k-1}) + K(t_k)(y(t_k) - G(t_k)x(t_{k|k-1})), \tag{4.19}$$

$$P(t_{k|k}) = P(t_{k|k-1}) - K(t_k)G(t_k)P(t_{k|k-1}) \tag{4.20}$$

$$= (I - K(t_k)G(t_k))P(t_{k|k-1})(I - K(t_k)G(t_k))^T + K(t_k)R_kK(t_k)^T, \tag{4.21}$$

where the Kalman gain is $K(t_k) = P(t_{k|k-1})G^T(t_k)(G(t_k)P(t_{k|k-1})G^T(t_k) + R_k)^{-1}$ and the output measurement vector from the actual system is $y(t_k)$.

4.3 Approximate Non-Linear Kalman Filters

In the non-linear case, the right hand side of (4.10) can be expanded using the Taylor series. For a scalar state variable, x, the expansion of the drift function around the mean is $f(x(t),t) = f(\bar{x}(t),t) + f'(\bar{x}(t),t)(x(t) - \bar{x}(t)) + \frac{1}{2}f''(\bar{x}(t),t)(x(t) - \bar{x}(t))^2 + \ldots$, where $f'(x(t),t)$ is the first derivative with respect to the state and $f''(x(t),t)$ is the second derivative with respect to the state. Taking the expectation of the expanded $f(x(t),t)$ function gives

$$\bar{f}(x(t),t) = f(\bar{x}(t),t) + \frac{1}{2}P(x(t),t)f''(\bar{x}(t),t) + \ldots \tag{4.22}$$

To evaluate (4.11), first consider that

$$\mathrm{E}\{x(t)f^T(x(t),t) - \bar{x}(t)\bar{f}^T(x(t),t)\} = P(x(t),t)f'(\bar{x}(t),t) + \ldots \tag{4.23}$$

4.3.1 The First-Order CD-EKF

Substituting the first-order expansion terms from (4.22) and (4.23) into (4.10) and (4.11) give the first-order CD-EKF moment equations

$$\frac{d\bar{x}(t)}{dt} = f(\bar{x}(t),t), \tag{4.24}$$

$$\frac{dP(x(t),t)}{dt} = P(x(t),t)J^T(x(t),t) + J(x(t),t)P(x(t),t), + q(t)Q(t)q(t)^T, \tag{4.25}$$

where $J(x(t),t) = \left.\frac{\partial f(x,t)}{\partial x}\right|_{x=\bar{x}(t)}$ is the state Jacobian evaluated at the mean state trajectory. The observation equations are the same as those given in (4.19) to (4.21), but with $G(t)$ replaced by the measurement Jacobian evaluated at the *a priori* mean vector, $H(x(t_k),t_k) = \left.\frac{\partial \Psi(x,t)}{\partial x}\right|_{x=\bar{x}(t_k)}$, such that

$$\bar{x}(t_{k|k}) = \bar{x}(t_{k|k-1}) + K(t_k)(y(t_k) - H(x(t_k),t_k)x(t_{k|k-1})), \tag{4.26}$$

$$P(t_{k|k}) = P(t_{k|k-1}) - K(t_k)H(x(t_k),t_k)P(t_{k|k-1}) \tag{4.27}$$

$$= (I - K(t_k)H(x(t_k),t_k))P(t_{k|k-1})(I - K(t_k)H(x(t_k),t_k))^T + K(t_k)R_kK(t_k)^T, \tag{4.28}$$

with Kalman gain $K(t_k) = P(t_{k|k-1})H^T(x(t_k),t_k)(H(x(t_k),t_k)P(t_{k|k-1})H^T(x(t_k),t_k) + R_k)^{-1}$.

4.3.2 The Second-Order CD-EKF

Substituting the first- and second-order expansion terms from (4.22) and (4.23) into (4.10) and (4.11) give the second-order CD-EKF moment equations

$$\frac{d\bar{x}(t)}{dt} = f(\bar{x}(t), t) + \frac{1}{2}P(x(t), t)f''(\bar{x}(t), t), \tag{4.29}$$

$$\frac{dP(x(t), t)}{dt} = P(x(t), t)J^T(x(t), t) + J(x(t), t)P(x(t), t) + q(t)Q(t)q^T(t)\}. \tag{4.30}$$

Note that the above formulation omits the second-order term in (4.23) following the convention in [7]. As with the first-order CD-EKF, the observation equations are the same as those given in (4.26) to (4.28).

4.3.3 The CD-UKF

The UKF, formulated in [8], is based on a set of 2n+1 function evaluations (the UTs) that allow the approximation of the first two moments in (4.10) and (4.11) according to a set of sigma points. These sigma points do a better job than the Jacobians in the first-order CD-EKF in approximating the MDEs because they represent more than simply just the first Taylor expanded term in (4.22) and (4.23). The matrix form of the moment differential equations associated with the CD-UKF presented in [19] is as follows:

$$\frac{dm(t)}{dt} = f(\chi(t))w_m, \tag{4.31}$$

$$\frac{dP(x(t), t)}{dt} = f(\chi(t))W\chi^T(t) + \chi(t)Wf^T(\chi(t)) + Q_c(t), \tag{4.32}$$

where $m(t) = \bar{x}(t)$ is simpler notation for the mean state vector of the UKF and

$$\begin{aligned}
&W = (I - [w_m, \ldots, w_m])\mathrm{diag}([W_c^0, \ldots, W_c^{2n}])(I - [w_m, \ldots, w_m])^T,\\
&w_m = [W_m^0, \ldots, W_m^{2n}]^T,\\
&W_m^0 = \frac{\lambda}{n+\lambda}, \quad W_c^0 = \frac{\lambda}{n+\lambda} + (1 - a^2 + \beta),\\
&W_m^i = W_c^i = \frac{1}{2(n+\lambda)}, \quad i = 1, \ldots, 2n,\\
&\lambda = a^2(n + \kappa) - n,\\
&\chi(t) = [m(t), \ldots, m(t)] + \sqrt{n+\lambda}\left[0, \sqrt{P(t)}, -\sqrt{P(t)}\ \right],
\end{aligned} \tag{4.33}$$

where a, β and κ are parameters of the UTs and λ is a resulting scaling factor. For Gaussian state distributions, β is typically set to 2, and generally $\kappa = 0$. The $\sqrt{P(t)}$ matrix is obtained using a Cholesky factorisation as in [19]. The square root of the state covariance can also be obtained through an eigen decomposition, referred to as a $\mathrm{UDU}^{\mathrm{T}}$ decomposition [40], also discussed in [39], and is found to be more efficient in certain cases only when compared to the Cholesky decomposition. Since the Cholesky decomposition can be used as a way to

check that the state covariance matrices are in fact positive-definite, it is the chosen decomposition method in the filter. The moment equations in (4.31) and (4.32) can then be integrated to obtain the future state mean and its error covariance. The measurement update equations of the CD-UKF are

$$K(t_k) = P_{xy}P_{yy}^{-1}, \tag{4.34}$$

$$P(t_{k|k}) = P(t_{k|k-1}) - K(t_k)P_{xy}^T, \tag{4.35}$$

$$m(t_{k|k}) = m(t_{k|k-1}) + K(t_k)(y(t_k) - z_k)), \tag{4.36}$$

where $P_{xy} = \chi(t_k)W\mathcal{Y}^T(t_k)$ is the cross-covariance between the *a priori* state estimate and the output, $P_{yy} = \mathcal{Y}(t_k)W\mathcal{Y}^T(t_k)$ is the output covariance and $\mathcal{Y}(t_k) = \Upsilon(\chi(t_k))$ is the UTs associated with the measurement equation. The predicted output obtained via the measurement UTs is $z_k = \mathcal{Y}(t_k)w_m$.

4.3.4 Monte Carlo Propagation

The propagation equation in (4.32) is, however, not utilised in the filter algorithms designed in this thesis for propagating the state covariance due to the amount of effort required to solve it when compared to a simpler approach detailed in Section 4.4. Another approach to covariance propagation can be adopted from Monte Carlo type filter techniques that simultaneously propagate all individual sigma points in (4.33) across the integration interval δ and calculates the future covariance according to

$$P(x(t+\delta), t+\delta) = \chi(t+\delta)W\chi^T(t+\delta) + \int_t^{t+\delta} Q_c(\tau)d\tau. \tag{4.37}$$

This is a much simpler solution to the task of propagating the state covariance since all future sigma points will in any case have to be calculated. A down side to this approach is that the filter algorithm would need to explicitly calculate and store the $\chi(t+\delta)$ matrix with dimensions 2n+1 $\times$ n.

4.4 State Propagation of the CD-HCKF

A CD-KF comprises two main steps: (1) time update (state propagation), and (2) measurement update (state correction). For a non-linear system, these processes are given above for the CD-EKF and CD-UKF. The new CD-KF uses a combination of these filters, referred to as a hybrid type filter. In addition to using the CD-EKF and CD-UKF, CD Monte Carlo methods are also employed. The reason for this amalgam will be conveyed as the filter development progresses in the remainder of the chapter.

Begin by employing the second-order accurate, linearly implicit TH scheme in (3.24) to solve the MDE in (4.24) or the more general case of $\frac{d(x(t))}{dt} = f(x(t))$. For an integration interval of δ the TH scheme is

$$x(t_{i+1}) = x(t_i) + \left(I - J(x(t_i))\frac{\delta}{2}\right)^{-1} f(x(t_i))\delta. \tag{4.38}$$

Although the new CD-KF proceeds with the second-order accurate TH scheme, it can be replaced by the third-order accurate linearly implicit scheme in (3.6). Both of these methods satisfy the necessary requirements for use in the new CD-KF, i.e., the method must be A-stable, have minimal inversion matrices, and use only 1 Jacobian evaluation at $x(t_i)$. In general, the TH scheme can be replaced by any linearly implicit (or SIRK) method in (3.4) that satisfies $\sum_{l=1}^{j-1} \mathbb{C}_{jl} = 0$.

The semi-implicit MGL scheme in [31] can be used to solve the MDE in (4.25). This scheme is second-order accurate, A-stable and produces PD covariances provided that $P(t_0)$ and $Q_c(t_0)$ are PD. This is ideal since $P(t)$ (by definition of a covariance) is always required to be PD. The scheme is given as

$$P(t_{i+1}) = M(t_\iota)P(t_i)M(t_\iota)^T + \varphi(t_\iota)Q_c(t_i)\varphi(t_\iota)^T\delta, \tag{4.39}$$

where $\iota = i + \frac{1}{2}$, $\varphi(t_\iota) = \left(I - J(x(t_\iota))\frac{\delta}{2}\right)^{-1}$ and $M(t_\iota) = \varphi(t_\iota)\left(I + J(x(t_\iota))\frac{\delta}{2}\right)$. Alternatively, the Monte Carlo based scheme in (4.37) can be employed to propagate the state covariance, where the process covariance term is integrated as done in (4.39) such that

$$P(t_{i+1}) = \chi(t_{i+1})W\chi^T(t_{i+1}) + \varphi(t_\iota)Q_c(t_i)\varphi(t_\iota)^T\delta. \tag{4.40}$$

Like (4.39), this scheme also produces symmetric state covariances and has the same stability and accuracy properties as the scheme used to solve the drift MDE, which in this case is the TH scheme in (4.38). This is on condition that the process covariance, $Q_c(t)$, is integrated with the same order of accuracy as the drift MDEs, which is in fact the case in (4.40). Here the $\varphi(t_\iota)Q_c(t_i)\varphi(t_\iota)^T\delta$ term maintains a symmetric covariance result and is second-order accurate.

The problem with having complex system dynamics is that the drift Jacobian needed in (4.38) and (4.39) is not easily computed, especially in large-scale problems, adding a level of difficultly to the implementation of linearly implicit methods. In order to solve this problem, the UTs are used to extract an equivalent of the state Jacobian, called the drift HOLM, and used as a substitute in (4.38), (4.39) and other linearly implicit methods such as (3.6).

4.4.1 The Drift HOLM

The matrix form of the UKF moment equations in (4.31) and (4.32) looks striking similar to those of the EKF in (4.24) and (4.25). This is naturally the case, since they are different ways to compute the same quantities. The theorem below formalises their relationship.

Theorem 4.4.1. *For a Lipschitz continuous function $f(x(t)) \in \mathbb{R}^n$ it follows that*

$$J(x(t)) + \Upsilon(x(t)) = f(\chi(t))W\chi^T(t)P(t)^{-1}, \tag{4.41}$$

where the drift Jacobian is that evaluated at the mean state trajectory obtained from the UTs, i.e., $J(x(t)) = \left.\frac{\partial f(x)}{\partial x}\right|_{x=m(t)}$,

$\Upsilon(x(t))$ *accounts for non-linear influences introduced by the UTs and is a combination of higher-order state derivatives,* $L^T P(t)L > 0$ *for all* $L \in \mathbb{R}^n$ *such that* $P(t)^{-1}$ *exists and* $\mathcal{J}(x(t)) \triangleq J(x(t)) + \Upsilon(x(t))$ *is the drift HOLM.*

(The i, j, k and l indices used in the following proof do not have the same meaning as elsewhere in the chapter.)

Proof. Dropping time dependence to simplify notation, consider a Taylor expansion of the transformed state vector under the drift function f such that

$$f(x) = f(m) + f'(m)(x-m) + \left[\frac{1}{2}(x-m)^T f^{i\prime\prime}(m)(x-m)\right]^i + \left[\frac{1}{3!}\sum_{j,k}\left\{((x-m)(x-m)^T)_{ijk} f^{i\,\prime\prime\prime}_{kjl}(m)\right\}(x-m)\right]^i + HOT, \tag{4.42}$$

where HOT stands for higher-order terms, m is the state mean in the UKF sense as in (4.31) and $[\cdot]^i$ is notation from [68] used to denote the i-th element of the column formed using the matrices $f^{i\prime\prime}(m) \in \mathbb{R}^{n\times n}$ and $f^{i\prime\prime\prime}(m) \in \mathbb{R}^{n\times n\times n}$. The matrices $f^{i\prime\prime}(m)$ and $f^{i\prime\prime\prime}(m)$ correspond to those formed using the i-th component of $f(m)$. Since P is PD it follows that $P = USU^T = \sum_{i=1}^n \sigma_i^2 u_i u_i^T$, where u_i is the i-th column vector of the unitary matrix U. Using the singular value decomposition of P above and setting $\kappa = 0$ according to convention, the sigma points from (4.33) are:

$$x^0 = m, \quad x^{\pm i} = m \pm \sqrt{a^2 n}\sigma_i u_i, \tag{4.43}$$

where $\chi^T(t) = [x^0, x^{+i}, x^{-i}]$ for $i = 1, \ldots, n$. Substitute (4.43) into (4.42) to obtain the transformed sigma points, simplified to

$$f(x^{\pm i}) = f(m) \pm f'(m)\sqrt{a^2 n}\sigma_i u_i + \frac{1}{2}\left[(\sqrt{a^2 n}\sigma_i u_i)^T f^{i\prime\prime}(m)\sqrt{a^2 n}\sigma_i u_i\right]^i \pm \frac{1}{3!}\left[\sum_{j,k}\left\{((\sqrt{a^2 n}\sigma_i u_i)(\sqrt{a^2 n}\sigma_i u_i)^T)_{ijk} f^{i\,\prime\prime\prime}_{kjl}(m)\right\}\sqrt{a^2 n}\sigma_i u_i\right]^i + HOT. \tag{4.44}$$

Expand part of the right-hand side of (4.41), where (4.45) below is due to [19], as follows:

$$\begin{aligned} & f(\chi(t))W\chi^T(t) \\ &= \sum_{i=0}^{2n} W_c^i (f(x^i) - m_f)(x^i - m)^T \qquad (4.45) \\ &= W_c^i \sum_{i=1}^{2n} (f(x^i) - m_f)(x^i - m)^T + W_c^0 (f(x^0) - m_f)(x^0 - m)^T \\ &= W_c^i \sum_{i=1}^{n} (f(x^i) - m_f)(x^i - m)^T + W_c^i \sum_{i=1}^{n} (f(x^{-i}) - m_f)(x^{-i} - m)^T, \qquad (4.46) \end{aligned}$$

where m_f is the mean of the drift function. Substitute the sigma points from (4.43) and the function evaluations from (4.44) into the above expression to give:

$$
\begin{aligned}
f(\chi(t))W\chi^T(t) &= W_c^i \sum_{i=1}^{n} \Bigg((f(m) + f'(m)\sqrt{a^2 n}\sigma_i u_i + \frac{1}{2}\Big[\sqrt{a^2 n}\sigma_i u_i)^T f^{i\prime\prime}(m)\sqrt{a^2 n}\sigma_i u_i\Big]^i \\
&\quad + \frac{1}{3!}\Big[\sum_{j,k}\Big\{((\sqrt{a^2 n}\sigma_i u_i)(\sqrt{a^2 n}\sigma_i u_i)^T)_{ijk} f^{i}_{kjl}{}^{\prime\prime\prime}(m)\Big\}(\sqrt{a^2 n}\sigma_i u_i)\Big]^i + HOT - m_f)(m + \sqrt{a^2 n}\sigma_i u_i - m)^T\Bigg) \\
&\quad + W_c^i \sum_{i=1}^{n} \Bigg((f(m) - f'(m)\sqrt{a^2 n}\sigma_i u_i + \frac{1}{2}\Big[\sqrt{a^2 n}\sigma_i u_i)^T f^{i\prime\prime}(m)\sqrt{a^2 n}\sigma_i u_i\Big]^i \\
&\quad - \frac{1}{3!}\Big[\sum_{j,k}\Big\{((\sqrt{a^2 n}\sigma_i u_i)(\sqrt{a^2 n}\sigma_i u_i)^T)_{ijk} f^{i}_{kjl}{}^{\prime\prime\prime}(m)\Big\}(\sqrt{a^2 n}\sigma_i u_i)\Big]^i + HOT - m_f)(m - \sqrt{a^2 n}\sigma_i u_i - m)^T\Bigg) \\
&= W_c^i \sum_{i=1}^{n} \Bigg(2f'(m)(\sqrt{a^2 n}\sigma_i u_i)(\sqrt{a^2 n}\sigma_i u_i)^T + \frac{2}{3!}\Big[\sum_{j,k}\Big\{((\sqrt{a^2 n}\sigma_i u_i)(\sqrt{a^2 n}\sigma_i u_i)^T)_{ijk} f^{i}_{kjl}{}^{\prime\prime\prime}(m)\Big\}(\sqrt{a^2 n}\sigma_i u_i)\Big]^i \\
&\quad (\sqrt{a^2 n}\sigma_i u_i)^T\Bigg) + HOT. \qquad (4.47)
\end{aligned}
$$

Note the cancellation of all even order state derivative terms in the analysis above. With this, right multiply by $P(t)^{-1} = US^{-1}U^T = \sum_{i=1}^{n} \frac{1}{\sigma_i^2} u_i u_i^T$, such that

$$
\begin{aligned}
f(\chi(t))W\chi^T(t)P(t)^{-1} &= W_c^i \sum_{i=1}^{n} 2f'(m)a^2 n\sigma_i^2 u_i u_i^T \frac{1}{\sigma_i^2} u_i u_i^T \\
&\quad + \frac{2}{3!}a^4 n^2 W_c^i \sum_{i=1}^{n}\Big[\sigma_i^4 \sum_{j,k}\Big\{(u_i u_i^T)_{ijk} f^{i}_{kjl}{}^{\prime\prime\prime}(m)\Big\}u_i\Big]^i u_i^T \frac{1}{\sigma_i^2} u_i u_i^T) + HOT \\
&= f'(m)\sum_{i=1}^{n} u_i u_i^T + \frac{1}{3!}a^2 \sum_{i=1}^{n}\Big[\sigma_i^2 \sum_{j,k}\Big\{(u_i u_i^T)_{ijk} f^{i}_{kjl}{}^{\prime\prime\prime}(m)\Big\}u_i u_i^T\Big]^i + HOT, \qquad (4.48)
\end{aligned}
$$

where $W_c^i = \frac{1}{2a^2 n}$ for $\kappa = 0$, $u_i^T u_i = 1$ and $\sum_{i=1}^{n} u_i u_i^T = 1$. Note that (4.48) is the same form as $\mathcal{J}(x)$ and is a linear combination of all odd order state derivative matrices such as the Jacobian. □

Remark 1. *According to Theorem 4.4.1 it is possible to encapsulate state dynamics via the UTs without the need to analytically or numerically evaluate state partial derivatives.*

Corollary 4.4.1.1. *For linear systems, the covariance propagation equations in (4.25) and (4.32) are equivalent so that*

$$f(\chi(t))W\chi^T(t) = J(x(t))P(t).$$

To see this, set all higher-order derivatives in (4.48) to zero.

4.4.2 State Propagation of the CD-HCKF Algorithm

For measurements arriving at times t_k it follows that $\delta N = \Delta$, where N is the number of integration time steps between consecutive measurements arriving at intervals of Δ, such that $t_{k,i} = t_k + i\delta$. For simplicity N is chosen as a constant but of course this can vary as a function of k. The time update process of the HCKF is outlined in Algorithm 1A. In step (1) and (2) of the algorithm the mean state trajectory is propagated according to the CD-UKF in (4.31), which is equivalent to (4.29) [68]. Step (3) of the algorithm makes use of Theorem 4.4.1. The result in step (5) of the algorithm can be shown by substituting the drift HOLM evaluated at the mean state trajectory in place of the drift Jacobian in (4.38) to obtain the sigma points at the next integration time step, $\chi(t_{k,i+1|k}) = \chi(t_{k,i|k}) + \left(I - \mathcal{J}(t_{k,i|k})\frac{\delta}{2}\right)^{-1} f(\chi)(t_{k,i|k})\delta$. Multiplying through by w_m gives $\chi(t_{k,i+1|k})w_m = \chi(t_{k,i|k})w_m + \left(I - \mathcal{J}(t_{k,i|k})\frac{\delta}{2}\right)^{-1} f(\chi)(t_{k,i|k})w_m\delta$. Using (4.31) and $m(t) = \chi(t)w_m$, the equation in step (5) follows. Provided that the sigma points in (4.43) are chosen at a reasonable spread, then $\mathcal{J}(t_{k,i|k})$ is a fair approximation to the true drift HOLM of individual sigma points, implying that the $\left(I - \mathcal{J}(t_{k,i|k})\frac{\delta}{2}\right)^{-1}$ matrix is still able to provide A-stable solutions. The value of a must therefore span a good portion of the state uncertainty, σ_i, so that substantial system detail is captured and no numerical problems are encountered. The requirements on a are confirmed when an example is visited in Section 4.6. It is easy to check that replacing the drift Jacobian with the drift HOLM in the TH scheme maintains second-order accuracy so it is omitted here. Note that the TH scheme in step (5) can easily be replace by the third-order accurate linearly implicit scheme in (3.6). In step (6) the state covariance is propagated via methods associated with the EKF in order to take advantage of the properties of the MGL scheme in (4.39) - bearing in mind that the φ matrix is re-used in this step. Unlike the original MGL scheme, however, the drift Jacobian is evaluated at time t_i instead of t_l. This has been chosen to avoid the additional effort required to obtain the Jacobian at time t_l and simply use expressions which have already been calculated. Consequently, the solution of the state covariance equation loses an order of accuracy but remains A-stable (see Appendix B.2). Knowledge of noise covariances in practical applications is, however, less accurate than the system model, implying that a second-order scheme for state covariance propagation could be overly optimistic about the covariance accuracy in many cases. Employing the state covariance propagation scheme in (4.40) instead of that in (4.39) incurs an additional step and would make use of much larger matrices in the algorithm (see Algorithm 1A alternative). In the interest of efficiency, it is therefore not used as the primary state covariance propagation scheme in this thesis.

4.5 State Correction of the CD-HCKF

The state correction stage of the CD-HCKF takes the *a priori* state estimates at the output of the state propagation stage, $m(t_{k+1,0|k}) = m_{k+1|k}$ and $P(t_{k+1,0|k}) = P_{k+1|k}$ (refer to Algorithm 1A), and calculates the *a posteriori* state estimates, $\hat{m}_{k+1} = m_{k+1|k+1}$ and $\hat{P}_{k+1} = P_{k+1|k+1}$, given measurements from the output of the system.

It is not uncommon for accumulated numerical error associated with limited computational precision to cause

Algorithm 1A: State propagation of the CD-HCKF

Initialise with $P(t_{k,o|k})$ and $m(t_{k,o|k})$. At each step $t_{k,i|k}$; for $i = 1, .., N$; calculate:

1. $\chi(t_{k,i|k}) = [m(t_{k,i|k}), \ldots, m(t_{k,i|k})] + \sqrt{n+\lambda}\ [0,\ \sqrt{P(t_{k,i|k})},\ -\sqrt{P(t_{k,i|k})}]$.
2. $\dot{\chi}(t_{k,i|k}) = f(\chi(t_{k,i|k}))$ and $\dot{m}(t_{k,i|k}) = \dot{\chi}(t_{k,i|k})w_m$.
3. $\mathcal{J}(t_{k,i|k}) = \dot{\chi}(t_{k,i|k})W\chi^T(t_{k,i|k})P(t_{k,i|k})^{-1}$.
4. $\varphi(t_{k,i|k}) = \left(I - \mathcal{J}(t_{k,i|k})\frac{\delta}{2}\right)^{-1}$ and $M(t_{k,i|k}) = \varphi(t_{k,i|k})\left(I + \mathcal{J}(t_{k,i|k})\frac{\delta}{2}\right)$.
5. $m(t_{k,i+1|k}) = m(t_{k,i|k}) + \varphi(t_{k,i|k})\dot{m}(t_{k,i|k})\delta$.
6. $P(t_{k,i+1|k}) = M(t_{k,i|k})P(t_{k,i|k})M^T(t_{k,i|k}) + \varphi(t_{k,i|k})Q_c(t_{k,i})\varphi^T(t_{k,i|k})\delta$.

At time $t_{k,N|k}$, where $t_{k,N|k} = t_{k+1,o|k}$, the *a priori* state vector and its associated covariance matrix is then set to $m(t_{k+1,o|k})$ and $P(t_{k+1,o|k})$. Calculate sigma points $\chi(t_{k+1,o|k})$ as done in step (1) for use in the state correction stage of the filter.

Algorithm 1A alternative

Replace steps (4), (5) and (6) in **Algorithm 1A** with the following calculations:

4. $\varphi(t_{k,i|k}) = \left(I - \mathcal{J}(t_{k,i|k})\frac{\delta}{2}\right)^{-1}$
5. $\chi(t_{k,i+1|k}) = \chi(t_{k,i|k}) + \varphi(t_{k,i|k})\dot{\chi}(t_{k,i|k})\delta$
6. $m(t_{k,i+1|k}) = \chi(t_{k,i+1|k})w_m$
7. $P(t_{k,i+1|k}) = \chi(t_{k,i+1|k})W\chi^T(t_{k,i+1|k}) + \varphi(t_{k,i|k})Q_c(t_{k,i})\varphi^T(t_{k,i|k})\delta$

state covariance matrices in (4.35) to lose positive-definiteness. The state covariance measurement update equation of the EKF in (4.21), however, is robust, stable and ensures a PD covariances [69] provided that $P_{k+1|k}$ from the output of the state propagation stage in Algorithm 1A is PD. Direct calculation of the measurement Jacobian is, however, as with the drift Jacobian, not always trivial. The same reasoning as in Section 4.4 can thus be applied to obtain the measurement HOLM, $\mathcal{H}(x(t_k))$, according to the following lemma in order to replace $H(x(t_k))$ in (4.26) to (4.28).

Lemma 4.5.1. *For a Lipschitz continuous function $\Upsilon(x(t)) \in \mathbb{R}^m$ it follows analogously from Theorem 4.4.1 that*

$$H(x(t_k)) + \Lambda(x(t_k)) = \Upsilon(\chi(t_k))W\chi^T(t_k)P(t_k)^{-1}, \tag{4.49}$$

where $H(x(t_k)) = \frac{\partial \Upsilon(x)}{\partial x}\Big|_{x=m(t_k)}$ is the measurement Jacobian evaluated at the a priori mean vector, $\Lambda(x(t_k))$ is a linear combination of higher-order derivative terms, $L^T P(t_k)L > 0$ for all $L \in \mathbb{R}^n$ such that $P(t_k)^{-1}$ exists and $\mathcal{H}(x(t_k)) \triangleq H(x(t_k)) + \Lambda(x(t_k))$ is the measurement HOLM.

With this, the method for the measurement update process is outlined in Algorithm 1B.

Algorithm 1B: State correction of the CD-HCKF

Given $\chi_{k+1|k}$ as well as $m_{k+1|k}$ and its associated covariance $P_{k+1|k}$ from the output of the propagation stage in Algorithm 1A, calculate:

1. $\mathcal{H}_{k+1|k} = \Upsilon(\chi_{k+1|k}) W \chi_{k+1|k}^T P_{k+1|k}^{-1}$, where $\chi_{k+1|k} = \chi(x)|_{x=m_{k+1|k}}$.
2. $K_{k+1} = P_{k+1|k}\mathcal{H}_{k+1|k}^T (\mathcal{H}_{k+1|k} P_{k+1|k} \mathcal{H}_{k+1|k}^T + R_{k+1})^{-1}$.
3. $P_{k+1|k+1} = (I - K_{k+1}\mathcal{H}_{k+1|k}) P_{k+1|k} (I - K_{k+1}\mathcal{H}_{k+1|k})^T + K_{k+1} R_{k+1} K_{k+1}^T$.
4. $m_{k+1|k+1} = m_{k+1|k} + K_{k+1}(y_{k+1} - z_{k+1})$.

4.6 Case Study

Consider example 1, the Van der Pol oscillator from Section 3.3.1. It is described by a stiff non-linear differential equation with characteristic parameter ε to control the level of damping and thus the amount of stiffness exhibited by the model. Its state vector comprises its position, x_1 [m], and velocity, x_2 [ms^{-1}]. Dropping explicit time dependence, the system model is given by

$$d\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ \varepsilon\left(1 - x_1^2\right)x_2 - x_1 \end{pmatrix} dt + \underbrace{\begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}}_{q(t)} dw(t),$$

with observation equation

$$y(t_k) = x_1(t_k) + u_k. \tag{4.50}$$

4.6.1 Filter Set-Up

Measurements, $y(t_k)$, are simulated using sampled position values from a model with relatively small integration time step sizes for $\varepsilon = 1.4$. Noise is added to this by random sampling from a Gaussian distribution with a standard deviation of 0.12 m. The measurement covariance is therefore chosen as $R_k = 0.13^2$ m^2 assuming that the exact noise distribution is unknown. The diffusion matrix is chosen as $Q_c = \text{diag}\{[\sigma_1^2 \text{ m}^2\text{s}^{-1}, \sigma_2^2 \text{ m}^2\text{s}^{-3}]\}$ with $\sigma_1 = 0$, $\sigma_2 = 0.22$ ms$^{-1.5}$, the initial state mean as $\hat{m}_0 = [0.5 \text{ m}, 0.5 \text{ ms}^{-1}]^T$ and the initial state error covariance as $\hat{P}_0 = \text{diag}\{[0.01 \text{ m}^2, 0.1 \text{ m}^2\text{s}^{-2}]\}$. Here diag$\{\cdot\}$ forms a diagonal matrix with the containing vector. The efficiency and accuracy of different filters are compared by observing the total execution time and average mean squared error (MSE) in velocity of 30 successive simulations.

4.6.2 Results and Discussion

It is noticed in this example that the difference between the drift HOLM and the drift Jacobian, $\Upsilon(x(t)) = \mathcal{J}(x(t)) - J(x(t))$, becomes smaller as a^2 is made smaller in the filter. This agrees with the expression on the

right hand side of (4.48) which implies that $\Upsilon(x(t)) \to 0$ as $a^2 \to 0$.

Five methods are singled out for comparison. (1) The CD-UKF-EKF_HOLM - a hybrid filter (with state covariance propagation according to the CD-EKF, and state correction and state mean propagation according to the CD-UKF) that uses the drift HOLM for model integration and state covariance propagation, (2) the CD-UKF-EKF_J - the same as the CD-UKF-EKF_HOLM but with model integration and state covariance propagation performed using the drift Jacobian, (3) the CD-HCKF - presented in Algorithms 1A and 1B, (4) the CD-EKF - a first-order CD-EKF that uses analytic Jacobians, and (5) the CD-UKF - where the state propagation equations in (4.31) and (4.32) are solved using the second-order implicit trapezoidal method (via Newton iterations and finite-difference Jacobian approximations). According to Fig. 4.6.1a the CD-UKF-EKF_HOLM is identical to the HCKF.

Filter covariances that are more prone to becoming poorly conditioned are forced by changing the corresponding inputs in Section 4.6.1 to $\hat{P}_0 = \text{diag}\{[0.001\ \text{m}^2, 0.1\ \text{m}^2\text{s}^{-2}]\}$ and $R_k = (3.3 \times 10^{-6})^2\ \text{m}^2$. The result is shown in Fig. 4.6.1b. Here the CD-UKF-EKF_HOLM fails execution (due to a failed Cholesky decomposition) as the model stiffness reaches $\varepsilon = 1.6$. The difference between the two filters is in their measurement update stages. The CD-UKF-EKF_HOLM updates according to the CD-UKF, whereas the CD-HCKF updates according to the symmetric CD-EKF equations. The plots in Fig. 4.6.1b. indicate that the CD-HCKF exhibits improved numerical stability over the CD-UKF-EKF_HOLM and approximately the same accuracy, supporting the paper's theoretical contribution.

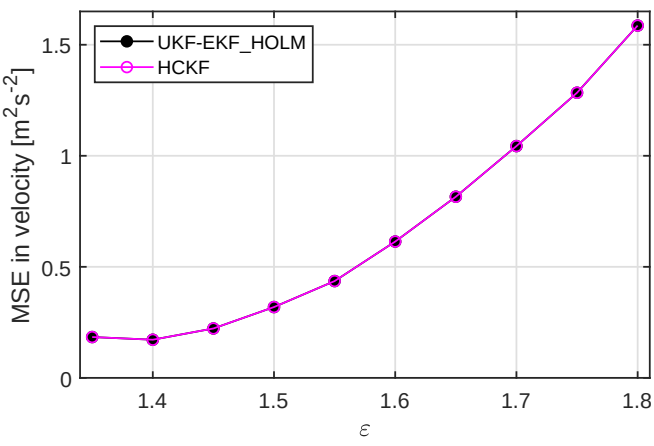
Consider again the standard filter set-up in Section 4.6.1. It can be seen in Figs. 4.6.1c and 4.6.1d that for an integration step size of $\delta = 0.25$ s, the CD-HCKF achieves similar accuracy to the CD-UKF and the CD-UKF-EKF_J on condition that $a^2 < 1$. Here the CD-HCKF appears to not decrease much in accuracy when a smaller fraction of the uncertainty is spanned. This makes sense theoretically since as $a^2 \to 0$ the Jacobian alone is extracted in (4.48). The lower bound on a^2 can thus be set according to the word length of the computer. As a^2 increases beyond 1, the sigma points span too wide a range, making the numerical process less reliable, and because it is coupled to the integration scheme, affects the estimates more drastically. It must also be considered that by increasing a^2, a larger fraction of the non-linear derivative matrices are included in the HOLM as can be seen in (4.48). The value of a^2 can thus be chosen based on these factors, and should ideally be less than 1. At $a^2 = 4$ and $\varepsilon = 1.45$ the CD-UKF again fails execution. This is attributed to the CD-UKF's state covariance propagation scheme (found in none of the other four methods) that causes covariance matrices to lose positive-definiteness. Similar results are observed for $\delta = 0.33$ s (Figs. 4.6.1e and 4.6.1f) where the CD-UKF again fails execution, however, this time at all values of ε and a^2. The CD-HCKF, on the other hand, manages to remain numerically stable. Note that the CD-UKF-EKF_J is a fair alternative to the CD-HCKF in terms of both accuracy and numerical stability but comes with the downside of requiring an analytic or numerical Jacobian calculated via usual means.

The discussion is ended with a comparison of algorithmic complexity between the filters. The CD-EKF calculates no UTs and executes the fastest at 1.80 s. The CD-UKF-EKF_HOLM and CD-UKF-EKF_J calculates

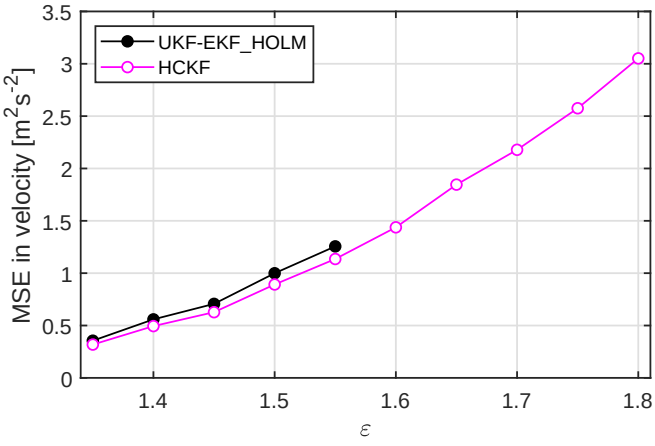
the same number of UTs, differing solely by whether the Jacobian or HOLM is formed, and executes in 14.56 s and 14.78 s respectively. The only significant complexity difference between the CD-HCKF, with an execution time of 14.45 s, and the CD-UKF-EKF_HOLM is in the measurement update multiplication process, which for n=2 is evidently negligible. The CD-UKF, with an execution time of 22.57 s, is the most inefficient due to the laborious Newton iterations.

4.7 Conclusion

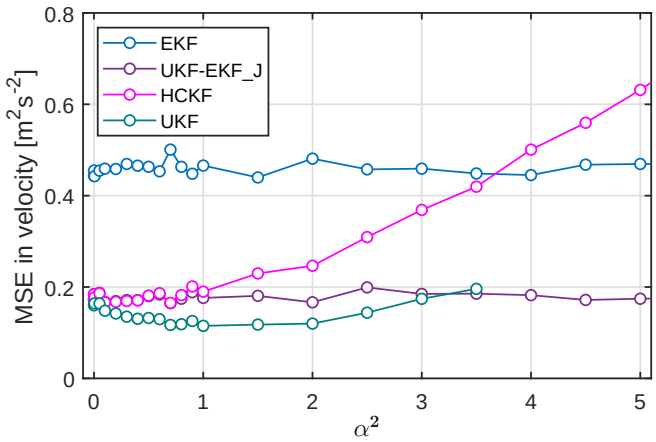
In this chapter the novel CD-HCKF was developed that was shown to be almost equivalent to the CD-UKF where accuracy is concerned, on condition that $a^2 < 1$, but with added benefits over the CD-UKF such as improved numerical stability and faster execution. Furthermore, no analytic or numerical Jacobians are necessary in the CD-HCKF which makes its implementation much easier in comparison to standard CD-EKF formulations. The core application in Part 2 of this thesis, involving a power boiler, includes a parallelised application of the UTs which should see a reduction in execution time for the CD-HCKF. Such a large-scale problem with an increased number of state variables and non-trivial Jacobian matrices will furthermore see an increase in the CD-EKF execution time, making the CD-HCKF an even more viable filter option.



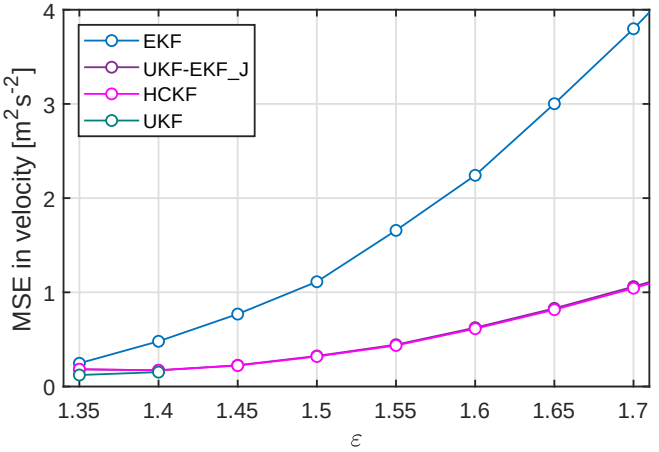
(a) $\delta = 0.25$ s, $\Delta = 0.5$ s, $a = \sqrt{0.05}$.



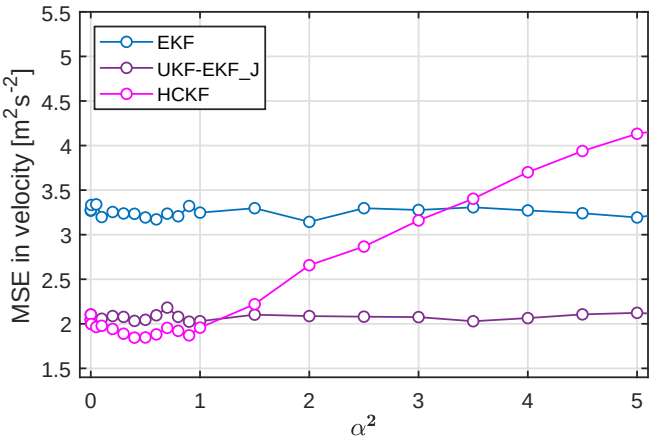
(b) $\delta = 0.25$ s, $\Delta = 0.5$ s, $a = \sqrt{0.05}$.



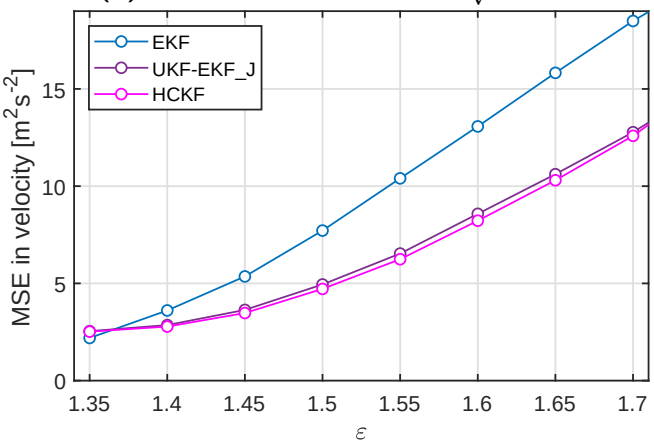
(c) $\delta = 0.25$ s, $\Delta = 0.5$ s, $\varepsilon = 1.4$.



(d) $\delta = 0.25$ s, $\Delta = 0.5$ s, $a = \sqrt{0.05}$.



(e) $\delta = 0.33$ s, $\Delta = 1$ s, $\varepsilon = 1.4$.



(f) $\delta = 0.33$ s, $\Delta = 1$ s, $a = \sqrt{0.05}$.

Figure 4.6.1: Example 1: Average MSE of the 30 simulations across a^2 and ε parameters for the respective CD filters.

5

Higher-Order Kalman Filters

In the previous chapter, approximate non-linear filters were discussed and a more suitable approach to CD Kalman filtering for stiff large-scale systems, termed the CD-HCKF, was developed. As with this filter and other classic approximate filters, they are based on the Euler-Maruyama (EM) truncation of the *Itô*-Taylor expansion. In this chapter, both an EM and a Milstein (higher-order) approximation of an amended SDE is embedded in the filter formulation to obtain variations of the CD-HCKF - termed the CD-HCKF_EM and the CD-HCKF_Milstein. Higher-order filters are not a new concept and have been introduced in [45]. Here a KF based on the *Itô*-1.5 expansion is developed and shown to produce better estimates in comparison to its lower order variants. An application of this filter to radar tracking in [71] highlights its superior performance. In a similar sense, Fourier-Hermite Kalman filters [72], also a class of Gaussian filters, are based on a truncation of the Fourier-Hermite series (an alternative to the Taylor series truncation in classic filters). In [72] filters based on truncations of the Fourier-Hermite series of up to third-order are demonstrated and are referred to as statistical higher-order filters. The difference between the work in this chapter and those already in the literature is the way in which the *Itô* SDE is defined, which includes a state-dependent process noise term. With respect to this SDE, a higher-order truncation of the *Itô*-Taylor series expansion (or other expansions in fact) results in a filter formulation that includes derivatives of the diffusion coefficient.

The covariances in classic Gaussian Kalman filters are defined as a constant or simply time-varying Gaussian field. This typically constitutes modelling error, process uncertainty and other statistical noise. Of course, the more accurate the noise models and the more comprehensive the list of uncertainty sources, the closer to optimal the filter will be, resulting in better filter performance [27]. It stands to reason that the uncertainty of the

numerical scheme should therefore also be included in the error budget as this could potentially form a significant part of system uncertainty when inadequate temporal discretisation steps are used in the numerical ODE solver. A state-varying process noise term is therefore included in the filter formulation to represent the uncertainty associated with the numerical solver, found to be proportional to the drift term, and hence a non-linear function of the state vector.

A statistical representation of the numerical error of this form is motivated in [73]. Here a Gaussian field is introduced into the state differential equation to reflect the uncertainty in the numerical scheme. This is referred to as a probabilistic solver and its convergence is formally shown in [74]. This motivates the assumption of a Gaussian based numerical error term in the amended SDE used in this chapter. Unlike [73], which models the stochastic error term using a constant diffusion coefficient, the formulation for the stochastic error model associated with the ODE solver in this chapter is based on the state-dependent error function in Chapter 3. In Chapter 3, expressions for both the local and global error estimates of a linearly implicit second-order accurate numerical method were formulated in (3.41) and (3.34), and were found to accurately represent the true errors associated with the numerical solution. Using this expression for the local error, an accurate model of the stochastic simulation error term can be formulated. Since the diffusion coefficient of this noise term is a function of the state, an assumption of a purely additive noise term reduces the filter's order of strong convergence from $\lambda = 1$ to $\lambda = 0.5$. A filter based on the Milstein disretisation scheme is therefore proposed in order to increase the filter's order of convergence to $\lambda = 1$ so that in the least, the convergence accuracy of the filter is not decreased. The Milstein scheme is based on a higher-order truncation of the *Itô*-Taylor expansion in comparison to the Euler-Maruyama approximation used in conventional KFs.

This chapter continues with the theme of CD filtering where the state vector is propagated in time between measurements. In order to achieve a certain level of accuracy during state propagation, one would need to make relatively small time steps or use highly sophisticated integration schemes in every unscented function. This leads to time consuming simulation and poses a problem for on-line estimation. The contribution of this chapter is therefore to facilitate faster estimation by employing coarser than usual integration steps in the second-order accurate linearly implicit simulation, but to systematically account for accumulated simulation error via the filtering process so that performance is not degraded. This is achieved by adding a state-dependent stochastic simulation error term alongside the conventional additive noise term in the standard SDE. The filter is then developed according to the EM discretisation of the SDE as well as a statistically higher-order discretisation of the SDE - the Milstein scheme. A Milstein based filter is motivated on the grounds that a false assumption of a purely constant diffusion coefficient in a SDE whose stochastic error term holds explicit state dependence, theoretically lowers the convergence accuracy of the filter.

This chapter proceeds as follows. Section 5.1 justifies the use of an amended SDE in the new CD filter approach. Section 5.2 begins with the development of an EM based representation of the filter MDEs according to the amended SDE and is followed by the development of a Milstein based representation of the filter MDEs according to the amended SDE, from which the novel CD-HCKF_Milstein is then formalised. The implementation

aspects of the CD-HCKF_Milstein are discussed in Section 5.3 and an outline of the filter algorithms given in Section 5.3.3. The performance of the filters are then evaluated in Section 5.4.

5.1 Amended Non-Linear State-Space Model

The conventional approximate non-linear CD-KF begins with the state SDE, typically already under the assumption of additive white noise,

$$dx(t) = f(x(t), t)dt + q(t)dw(t), \tag{5.1}$$

where $x(t) \in \mathbb{R}^n$ is a Markov process representing the system states; $f(x(t), t)$ is the drift function of the state; $dw(t) \in \mathbb{R}^{d_1 \times 1}$ is an independent increment function belonging to the Brownian motion $w(t)$; $q(t) \in \mathbb{R}^{n \times d_1}$ is the diffusion coefficient or intensity factor that scales the noise and $E(dw(t)dw^T(t)) = Q(t)dt$. Here $Q(t)$ $[s^{-1}]$ is the noise spectral density taken as the identity matrix I_{d_1}. The measurement equation is given by

$$y(t_k) = \Upsilon(x(t_k), t_k) + u(t_k), \tag{5.2}$$

where $y(t_k)$ is a vector of m outputs and $u(t_k) \in \mathbb{R}^m$ is the white measurement noise with $E(u_k u_k^T) = R(t_k) \in \mathbb{R}^{m \times m}$. It is assumed here that $dw(t)$, $u(t_k)$ and $x(t)$ are independent. Following an EM based discretisation of the process in (5.1), the standard CD (or continuous-time) EKF [7] and CD-UKF [75] can be derived.

In [73] a Gaussian field is introduced in the state differential equation to reflect the uncertainty in the numerical scheme. This solver takes the same form as the *Ito*-based SDE in (5.1) and is given as

$$dx(t) = f(x(t), t)dt + \delta^{p+1/2}\sqrt{\tilde{Q}}dw(t), \tag{5.3}$$

where $\tilde{Q} \in \mathbb{R}^{n \times n}$ is a covariance matrix found by calibrating against classical error estimation techniques. This is referred to as a probabilistic solver and justifies the assumption of a Gaussian based numerical error term.

In this chapter, a different form of the Gaussian numerical error term is proposed which involves a state-dependent function. This gives a more accurate bound on the covariance of the simulation uncertainty. The state-dependent noise term is added to the SDE in (5.1) alongside the additive noise term as follows:

$$dx(t) = f(x(t), t)dt + q(t)dw(t) + b(x(t), t)dv(t), \tag{5.4}$$

where $w(t)$ and $v(t)$ are independent standard Brownian motions and $b(x(t), t) \in \mathbb{R}^{n \times d_2}$ is a diffusion coefficient associated with the model integration error. Here $d_2 = 1$ since the state-dependent noise term models only one source of error - numerical integration error (or simulation error). The two noise terms are modelled separately to easily distinguish the two and also because they will be calculated separately (one on-line and one off-line). The $q(t)dw(t)$ noise term can be modelled in the conventional sense as an off-line task and comprises model error, process disturbances and other system uncertainties (as done in the case study in Section 4.6).

The state-dependent noise term, however, is calculated at each integration time step and is mainly an on-line process.

5.2 CD Kalman Filtering Using The Amended SDE

In this section, the CD-KF for a system with a state-dependent noise term is formulated for two respective SDE discretisation approximations based on truncations of the *Itô*-Taylor expansion in (A.31) - the EM and Milstein approximations. The *Itô* 1.5 based filter, a more accurate approximation than both the EM or Milstein approximation, is developed in [45] for a constant or simply time-varying noise term, $q(t)dw(t)$. The resultant filter therefore has no derivative diffusion terms irrespective of the approximation scheme. For the SDE in (5.4) the noise term is not simply time-varying or constant. As a result, a slightly different filter representation emerges when embedded in higher-order approximations of the SDE.

5.2.1 Approximations of the SDE

From the *Itô*-Taylor expansion in (A.31), various strong order discrete-time approximations can be constructed. Consider the generic SDE

$$dx(t) = f(x(t), t)dt + b(x(t), t)dv(t). \tag{5.5}$$

The simplest discrete approximation of (5.5) is the explicit Euler method, referred to as the explicit Euler-Maruyama (EM) method, and has a strong order of convergence of $\lambda = 0.5$. It is given over time $(t, t+\delta)$ as

$$x(t+\delta) = x(t) + \delta f(x(t), t) + b(x(t), t)v(t), \tag{5.6}$$

where $v(t)$ is a random Gaussian variable redefined as $v(t) \sim \mathcal{N}(0, \delta I_{d_2})$, equivalent to the Wiener increment, $\Delta W = W_{i+1} - W_i$. It is chosen that $d_2 = 1$. Adding more terms from the *Itô*-Taylor expansion in (A.31) gives the Milstein scheme [50] over time $(t, t+\delta)$ as

$$x(t+\delta) = x(t) + \delta \underline{f}(x(t), t) + b(x(t), t)v(t) + \frac{1}{2}b'(x(t), t)b(x(t), t)v^2(t), \tag{5.7}$$

where $\underline{f}(x(t), t) = f(x(t), t) - \frac{1}{2}b'(x(t), t)b(x(t), t)$. The Milstein scheme has a strong order of convergence of $\lambda = 1$. It is easily seen that when the noise term is purely time-varying or a constant, i.e., $b(x(t), t) = b(t)$, the approximation reduces to that of the EM scheme but maintains its 1.0 order of convergence. In the standard derivation of non-linear approximate filters, the diffusion coefficient in the noise term is constant or purely time-varying as in (5.1). Under this assumption, the moment equations of various CD non-linear approximate filters can be derived by applying an EM discretisation scheme to (5.1), and exhibits a $\lambda = 1$ order of strong convergence. The differential equations in this case are given by (4.10) and (4.11) and lead to standard approximate filters in Section 4.3. It is possible to go a step further and write down the order 1.5 *Itô* strong Tay-

lor scheme used in [45] (which includes even more terms from the *Itô*-Taylor expansion). The work in [45] does not, however, include state-dependent noise terms and thus any discretisation scheme does not result in derivative diffusion coefficient terms in the filter. The work in this chapter is limited to a scheme with a 1.0 order of strong convergence, so that in the least, convergence accuracy of the filter is maintained. It follows now to derive the EM and Milstein based filters associated with the amended SDE in (5.4).

5.2.2 EM Based Moment Differential Equations

Theorem 5.2.1. *Let $f(x(t),t) \in \mathbb{R}^{n\times 1}$ and $b(x(t),t) \in \mathbb{R}^{n\times 1}$ be Lipschitz continuous functions, then the dynamic moment equations of the amended SDE in (5.4), discretised according to the EM approximation in (5.6), are*

$$\frac{d}{dt}E\{x(t)\} = E\{f(x(t),t)\} \tag{5.8}$$

$$\frac{d}{dt}var\{x(t)\} = cov\{x(t),f(x(t),t)\} + cov\{f(x(t),t),x(t)\} + q(t)q^T(t) + E\{b(x(t),t)b^T(x(t),t)\}. \tag{5.9}$$

Proof. Consider the amended state SDE in (5.4) and apply the EM approximation in (5.6) over time $(t, t+\delta)$ to obtain:

$$x(t+\delta) \approx x(t) + \delta f(x(t),t) + q(t)w(t) + b(x(t),t)v(t), \tag{5.10}$$

where $w(t) \sim \mathcal{N}(0, \delta I_{d_1})$ and $v(t) \sim \mathcal{N}(0, \delta I_{d_2})$. The mean of (5.10) and its difference equation is as follows:

$$\begin{aligned} \mathrm{E}\{x(t+\delta)\} &= \mathrm{E}\{x(t)\} + \delta\mathrm{E}\{f(x(t),t)\} + \mathrm{E}\{q(t)w(t)\} + \mathrm{E}\{b(x(t),t)v(t)\} \\ &= \mathrm{E}\{x(t)\} + \delta\mathrm{E}\{f(x(t),t)\}. \\ \therefore \frac{\mathrm{E}\{x(t+\delta)\} - \mathrm{E}\{x(t)\}}{\delta} &= \mathrm{E}\{f(x(t),t)\}. \end{aligned} \tag{5.11}$$

With no correlation between state-dependent terms and the Brownian processes, the covariance of (5.10) and

its difference equation becomes:

$$
\begin{aligned}
\operatorname{var}\{x(t+\delta)\} &= \operatorname{var}\{x(t) + \delta f(x(t),t) + q(t)w(t) + b(x(t),t)v(t)\} \\
&= \operatorname{var}\{x(t)\} + \delta^2 \operatorname{var}\{f(x(t),t)\} + \operatorname{var}\{q(t)w(t)\} + \operatorname{var}\{b(x(t),t)v(t)\} \\
&\quad + \delta \operatorname{cov}\{x(t), f(x(t),t)\} + \delta \operatorname{cov}\{f(x(t),t), x(t)\} \\
&= \operatorname{var}\{x(t)\} + \delta^2 \operatorname{var}\{f(x(t),t)\} + \mathrm{E}\{(q(t)w(t))(q(t)w(t))^T\} \\
&\quad - \mathrm{E}\{q(t)w(t)\}\mathrm{E}\{q(t)w(t)\}^T + \mathrm{E}\{(b(x(t),t)v(t))(b(x(t),t)v(t))^T\} \\
&\quad - \mathrm{E}\{b(x(t),t)v(t)\}\mathrm{E}\{b(x(t),t)v(t)\}^T + \delta \operatorname{cov}\{x(t), f(x(t),t)\} \\
&\quad + \delta \operatorname{cov}\{f(x(t),t), x(t)\} \\
&= \operatorname{var}\{x(t)\} + \delta q(t) I_{\mathrm{d}_1} q(t)^T + \delta \mathrm{E}\{b(x(t),t) I_{\mathrm{d}_2} b(x(t),t)^T\} \\
&\quad + \delta \operatorname{cov}\{x(t), f(x(t),t)\} + \delta \operatorname{cov}\{f(x(t),t), x(t)\}. \\
\therefore \frac{\operatorname{var}\{x(t+\delta)\} - \operatorname{var}\{x(t)\}}{\delta} &= q(t)q(t)^T + \mathrm{E}\{b(x(t),t)b(x(t),t)^T\} + \operatorname{cov}\{x(t), f(x(t),t)\} \\
&\quad + \operatorname{cov}\{f(x(t),t), x(t)\}. \qquad (5.12)
\end{aligned}
$$

Here the δ^2 terms are set to zero. Taking the limit of (5.11) and (5.12) as $\delta \to 0$ gives the corresponding dynamic moment differential equations in (5.8) and (5.9).

□

Remark 2. *For an EM approximation, the amended system model in (5.4) produces similar moment dynamics to the standard model in (5.1). The expected values of the respective noise covariances simply add. See (4.10) and (4.11). Unlike the standard formulation in (4.10) and (4.11) that exhibits a strong order of convergence of $\lambda = 1$, the EM approximation applied to the amended system model in the above theorem results in an order of strong convergence of $\lambda = 0.5$. From (5.8) and (5.9), the standard CD-EKF and CD-UKF can be implemented. It should be noted that evaluation of the expectation and covariances in equation (5.8) and (5.9) will differ depending on whether a CD-EKF, CD-UKF, CD-CKF, etc, is employed.*

Remark 3. *In Section 4.1, classic filters were derived using Itô's Lemma. It was mentioned that it can be derived via other methods such as Kolmogorov's forward equation in (4.6) or via a truncation of the Itô-Taylor expansion and then taking the limit for the continuous-time case. This truncation method is used in Theorem 5.2.1. The reason that Itô's Lemma and Kolmogorov's forward equation give the same result as an EM approximation of the Itô-Taylor expansion is because they too are based on first-order Taylor approximations.*

5.2.3 Milstein Based Moment Differential Equations

Taking a similar approach as in Section 5.2.2, the MDEs based on the Milstein discretisation approximation, which exhibits a $\lambda = 1$ order of strong convergence, are given by the following theorem.

Theorem 5.2.2. *Let $f(x(t),t) \in \mathbb{R}^{n\times 1}$ and $b(x(t),t) \in \mathbb{R}^{n\times 1}$ be Lipschitz continuous functions, then the dynamic moment equations of the amended SDE in (5.4), discretised according to the Milstein approximation in (5.7), are*

$$\frac{d}{dt}E\{x(t)\} = E\{\underline{f}(x(t),t)\} + \frac{1}{2}E\{b'(x(t),t)b(x(t),t)\}, \tag{5.13}$$

$$\begin{aligned}\frac{d}{dt}var\{x(t)\} &= q(t)q(t)^T + E\{b(x(t),t)b^T(x(t),t)\} + cov\{x(t),\underline{f}(x(t),t)\} + cov\{\underline{f}(x(t),t),x(t)\}\\ &+ \frac{3}{4}\int_t^{t+dt} E\{(b'(x(t),t)b(x(t),t))(b'(x(t),t)b(x(t),t))^T\}dt\\ &- \frac{1}{4}\int_t^{t+dt} E\{b'(x(t),t)b(x(t),t)\}E\{b'(x(t),t)b(x(t),t)\}^T dt,\end{aligned} \tag{5.14}$$

where $\underline{f}(x(t),t) = f(x(t),t) - \frac{1}{2}b'(x(t),t)b(x(t),t)$ *and* $b'(x(t),t) \in \mathbb{R}^{n\times n}$ *is the derivative of the diffusion coefficient with respect to the state, or the Jacobian of the diffusion coefficient.*

Proof. Apply the Milstein approximation in (5.7) to (5.4) such that

$$x(t+\delta) = x(t) + \delta f(x(t),t) + q(t)w(t) + b(x(t),t)v(t) + \frac{1}{2}b'(x(t),t)b(x(t),t)[v^2(t) - \delta]. \tag{5.15}$$

The mean of (5.15) and its difference equation is:

$$\begin{aligned}\mathrm{E}\{x(t+\delta)\} &= \mathrm{E}\{x(t)\} + \delta\mathrm{E}\{\underline{f}(x(t),t)\} + \mathrm{E}\{\frac{1}{2}b'(x(t),t)b(x(t),t)v^2(t)\}\\ &= \mathrm{E}\{x(t)\} + \delta\mathrm{E}\{\underline{f}(x(t),t)\} + \frac{1}{2}\mathrm{E}\{b'(x(t),t)b(x(t),t)\}\delta I_{d_z}.\\ \therefore \frac{\mathrm{E}\{x(t+\delta)\} - \mathrm{E}\{x(t)\}}{\delta} &= \mathrm{E}\{\underline{f}(x(t),t)\} + \frac{1}{2}\mathrm{E}\{b'(x(t),t)b(x(t),t)\}.\end{aligned} \tag{5.16}$$

The covariance of (5.15) follows previous arguments in Theorem 5.2.1 about the state and noise terms such that

$$\begin{aligned}\mathrm{var}\{x(t+\delta)\} &= \mathrm{var}\{x(t)\} + \delta^2\mathrm{var}\{\underline{f}(x(t),t)\} + \mathrm{var}\{q(t)w(t)\} + \mathrm{var}\{b(x(t),t)v(t)\}\\ &+ \mathrm{var}\{\frac{1}{2}((b'(x(t),t)b(x(t),t)v^2(t)\} + \delta\mathrm{cov}\{x(t),\underline{f}(x(t),t)\} + \delta\mathrm{cov}\{\underline{f}(x(t),t),x(t)\}\\ &+ \mathrm{cov}\{\frac{1}{2}((b'(x(t),t)b(x(t),t)v^2(t),b(x(t),t)v(t)\} + \mathrm{cov}\{b(x(t),t)v(t),\frac{1}{2}((b'(x(t),t)b(x(t),t)v^2(t)\}.\end{aligned} \tag{5.17}$$

Noting that $\mathrm{cov}\{v(t),v^2(t)\} = 0$, it follows that

$$\begin{aligned}\mathrm{cov}\{b(x(t),t)v(t),\frac{1}{2}(b(x(t),t)(b'(x(t),t)v^2(t)\} &= \frac{1}{2}\mathrm{E}\{(b(x(t),t)(b'(x(t),t)b(x(t),t))^T\}\mathrm{cov}\{v(t),v^2(t)\}\\ &= 0.\end{aligned}$$

Similarly $\mathrm{cov}\{\frac{1}{2}b'(x(t),t)b(x(t),t)v^2(t),b(x(t),t)v(t)\} = 0$. Expanding and applying property 13 in Appendix

A.1.4 gives

$$\begin{aligned}\mathrm{var}\{\tfrac{1}{2}b'(x(t),t)b(x(t),t)v^2(t)\} &= \tfrac{1}{4}\mathrm{E}\{(b'(x(t),t)b(x(t),t)v^2(t))(b'(x(t),t)b(x(t),t)v^2(t))^T\}\\ &\quad-\tfrac{1}{4}\mathrm{E}\{b'(x(t),t)b(x(t),t)v^2(t)\}\mathrm{E}\{b'(x(t),t)b(x(t),t)v^2(t)\}^T\\ &= \tfrac{3}{4}\mathrm{E}\{(b'(x(t),t)b(x(t),t))(b'(x(t),t)b(x(t),t))^T\}(\delta I_{\mathrm{d}_2})^2\\ &\quad-\tfrac{1}{4}\mathrm{E}\{b'(x(t),t)b(x(t),t)\}\mathrm{E}\{b'(x(t),t)b(x(t),t)\}^T(\delta I_{\mathrm{d}_2})^2. \qquad (5.18)\end{aligned}$$

The variance in (5.17) and its difference equation then becomes

$$\begin{aligned}\mathrm{var}\{x(t+\delta)\} &= \mathrm{var}\{x(t)\} + \delta q(t)q^T(t) + \delta\mathrm{E}\{b(x(t),t)b^T(x(t),t)\}\\ &\quad+\delta\mathrm{cov}\{x(t),\underline{f}(x(t),t)\}+\delta\mathrm{cov}\{\underline{f}(x(t),t),x(t)\}\\ &\quad+\frac{3\delta^2}{4}\mathrm{E}\{(b'(x(t),t)b(x(t),t))(b'(x(t),t)b(x(t),t))^T\}\\ &\quad-\frac{\delta^2}{4}\mathrm{E}\{b'(x(t),t)b(x(t),t)\}\mathrm{E}\{b'(x(t),t)b(x(t),t\})^T.\\ \therefore\ \frac{\mathrm{var}\{x(t+\delta)\}-\mathrm{var}\{x(t)\}}{\delta} &= q(t)q^T(t)+\mathrm{E}\{b(x(t),t)b^T(x(t),t)\}\\ &\quad+\mathrm{cov}\{x(t),\underline{f}(x(t),t)\}+\mathrm{cov}\{\underline{f}(x(t),t),x(t)\}\\ &\quad+\frac{3\delta}{4}\mathrm{E}\{(b'(x(t),t)b(x(t),t))(b'(x(t),t)b(x(t),t))^T\}\\ &\quad-\frac{\delta}{4}\mathrm{E}\{b'(x(t),t)b(x(t),t)\}\mathrm{E}\{b'(x(t),t)b(x(t),t\})^T. \qquad (5.19)\end{aligned}$$

Taking the limits of (5.16) and (5.19) as $\delta\to 0$ gives the result in (5.13) and (5.14). □

5.2.4 The CD-HCKF_Milstein

State Propagation

In the state propagation stage of the filter, the continuous-time moment equations in (5.13) and (5.14) need to be solved between successive measurements. The moments are propagated between measurement updates at times t_k and t_{k+1} in intervals of δ such that x_k^i represents $x(t)$ at time $t = k\Delta + i\delta$, where $N\delta = \Delta$ and N is the number of integration steps between measurements. Integrate both sides of (5.13) to obtain the solution

$$\Delta\mathrm{E}\{x(t)\} = \int_t^{t+\delta}\mathrm{E}\{\underline{f}(x(t),t)\}dt + \int_t^{t+\delta}\frac{1}{2}\mathrm{E}\{b'(x(t),t)b(x(t),t)\}dt. \qquad (5.20)$$

The first integral term in (5.20) is the solution associated with the drift term and the second integral is the solution associated with the diffusion term. The integral associated with the diffusion term can be evaluated using

the first-order accurate Euler scheme. The solution of the mean state estimate in this case is

$$\begin{aligned}\bar{x}_{k|k}^{i+1} &= \mathrm{E}\{x_{k|k}^{i+1}\} \\ &= \mathrm{E}\{\underline{f_d}(x_{k|k}^i)\} + \frac{\delta}{2}\mathrm{E}\{b'(x_{k|k}^i)b(x_{k|k}^i)\} \\ &= \int_{\mathbb{R}^n} \underline{f_d}(x_{k|k}^i)\mathcal{N}(x_k)dx_k + \frac{\delta}{2}\int_{\mathbb{R}^n} b'(x_{k|k}^i)b(x_{k|k}^i)\mathcal{N}(x_k)dx_k, \end{aligned} \tag{5.21}$$

where $\underline{f_d}(t) = x(t_i) + \delta\underline{\Phi}(t_i)$ is the solution of the drift function and $\underline{\Phi}(t_i)$ is the increment function associated with $\underline{f}(t)$ for the TH scheme. The respective terms in (5.21) are solved with different levels of accuracy - the drift term being second-order accurate and the diffusion term being first-order accurate. This will not be a problem because it will later be revealed in Section 5.3.2 that $b(x(t),t) \propto \delta^{p+1/2}$, while $b'(x(t),t) \propto \delta^{-1/2}$. It can thus be said that $b'(x(t),t)b(x(t),t) \propto \delta^p$. For a p-th order solution of $\underline{f_d}(t)$ in (5.21) it follows that

$$\bar{x}_{k|k}^{i+1} = \int_{\mathbb{R}^n} \underline{f_d}(x_{k|k}^i)\mathcal{N}(x_k)dx_k + \mathcal{O}(\delta^{p+1}) + \frac{1}{2}\mathcal{O}(\delta^{p+1}). \tag{5.22}$$

In the limit as $\delta \to 0$ of (5.22), higher-order terms vanish faster, resulting in an asymptotically correct mean of

$$\bar{x}_{k|k}^{i+1} = \int_{\mathbb{R}^n} f_d(x_{k|k}^i)\mathcal{N}(x_k)dx_k. \tag{5.23}$$

Similarly with the mean state estimate, the solution of the state covariance estimate from (5.14) is

$$\begin{aligned} P_{k|k}^{i+1} &= \mathrm{E}\{(x_{k|k}^{i+1} - \bar{x}_{k|k}^{i+1})(x_{k|k}^{i+1} - \bar{x}_{k|k}^{i+1})^T\} \\ &= \int_{\mathbb{R}^n} \underline{f_d}(x_{k|k}^i)\underline{f_d}^T(x_{k|k}^i)\mathcal{N}(x_k)dx_k + \delta\int_{\mathbb{R}^n} b(x_{k|k}^i)b^T(x_{k|k}^i)\mathcal{N}(x_k)dx_k + \delta qq^T \\ &+ \frac{3\delta^2}{4}\int_{\mathbb{R}^n} b'(x_{k|k}^i)b(x_{k|k}^i)(b'(x_{k|k}^i)b(x_{k|k}^i))^T\mathcal{N}(x_k)dx_k \\ &- \frac{\delta^2}{4}\Big(\int_{\mathbb{R}^n} b'(x_{k|k}^i)b(x_{k|k}^i)\mathcal{N}(x_k)dx_k\Big)\Big(\int_{\mathbb{R}^n} b'(x_{k|k}^i)b(x_{k|k}^i)\mathcal{N}(x_k)dx_k\Big)^T - \bar{x}_{k|k}^{i+1}(\bar{x}_{k|k}^{i+1})^T, \end{aligned} \tag{5.24}$$

where the terms that are made up of the diffusion coefficient are solved according to a first-order accurate Euler scheme. Using a first-order accurate solution of the terms containing the diffusion coefficient, but a second-order accurate solution of the drift terms, can be motivated in the same way as was done for the mean state solution in (5.23) and exhibits the correct convergence properties as $\delta \to 0$. The first integral in (5.21) is evaluated as

$$\int_{\mathbb{R}^n} \underline{f_d}(x_{k|k}^i)\mathcal{N}(x_k)dx_k = \underline{f_d}(\chi_{k|k}^i)w_m \tag{5.25}$$

according to the CD-UKF in Section 4.3.3 and

$$\int_{\mathbb{R}^n} \underline{f_d}(x^i_{k|k})\mathcal{N}(x_k)dx_k = \underline{f_d}(\bar{x}^i_{k|k}) \tag{5.26}$$

according to the first-order CD-EKF. Other approaches, such as those based on the second-order CD-EKF (see Section 4.3.2) or the CD-CKF are also possible [9]. A first-order Euler solution of the first and last term in (5.24) is:

$$\begin{aligned}\int_{\mathbb{R}^n} \underline{f_d}(x^i_{k|k})\underline{f_d}^T(x^i_{k|k})\mathcal{N}(x_k)dx_k - \bar{x}^{i+1}_{k|k}(\bar{x}^{i+1}_{k|k})^T &= \text{var}\{x^i_{k|k}\} + \delta\text{cov}\{x^i_{k|k}, \underline{f}(x^i_{k|k})\} + \delta\text{cov}\{\underline{f}(x^i_{k|k}), x^i_{k|k}\} \\ &= P^i_{k|k} + \delta\Big(\text{cov}\{x^i_{k|k}, \underline{f}(x^i_{k|k})\} + \text{cov}\{\underline{f}(x^i_{k|k}), x^i_{k|k}\}\Big).\end{aligned} \tag{5.27}$$

Similarly with the state mean, the above can be expressed according to various filter formulations. Using the UTs from the CD-UKF, (5.27) is:

$$\int_{\mathbb{R}^n} \underline{f_d}(x^i_{k|k})\underline{f_d}^T(x^i_{k|k})\mathcal{N}(x_k)dx_k - \bar{x}^{i+1}_{k|k}(\bar{x}^{i+1}_{k|k})^T = P^i_{k|k} + \delta\Big(\chi^i_{k|k} W \underline{f}^T(\chi^i_{k|k}) + \underline{f}(\chi^i_{k|k}) W(\chi^i_{k|k})^T\Big). \tag{5.28}$$

Similarly, a first-order CD-EKF equivalent is:

$$\int_{\mathbb{R}^n} \underline{f_d}(x^i_{k|k})\underline{f_d}^T(x^i_{k|k})\mathcal{N}(x_k)dx_k - \bar{x}^{i+1}_{k|k}(\bar{x}^{i+1}_{k|k})^T = P^i_{k|k} + \delta\Big(J^i_{k|k}P^i_{k|k} + P^i_{k|k}(J_{k|k^i})^T\Big), \tag{5.29}$$

where $J^i_{k|k} = \frac{d}{dx}\underline{f}(x)|_{x=\bar{x}^i_{k|k}}$ is the first-order Jacobian of the mean drift term. Of course better integration methods for (5.27) or (5.28) can be used instead of a first-order Euler scheme, and can be found in, for example, [31] or [35].

The integrals in (5.21) and (5.24) associated with the non-linear diffusion coefficient are evaluated according to a simplifying assumption, similar to that in [45]: if the non-linearity in $b(x(t), t)$ is not too severe then $\int_{\mathbb{R}^n} b'(x^i_{k|k})b(x^i_{k|k})\mathcal{N}(x_k)dx_k \approx b'(\bar{x}^i_{k|k})b(\bar{x}^i_{k|k})$, and similarly for the integrals in (5.24) such that (5.21) and (5.24) simplify to

$$\bar{x}^{i+1}_{k|k} = \int_{\mathbb{R}^n} \underline{f_d}(x^i_{k|k})\mathcal{N}(x_k)dx_k + \frac{\delta}{2}b'(\bar{x}^i_{k|k})b(\bar{x}^i_{k|k}), \tag{5.30}$$

$$\begin{aligned}P^{i+1}_{k|k} = &\int_{\mathbb{R}^n} \underline{f_d}(x^i_{k|k})\underline{f_d}^T(x^i_{k|k})\mathcal{N}(x_k)dx_k + \delta b(\bar{x}^i_{k|k})b^T(\bar{x}^i_{k|k}) + \delta qq^T + \frac{\delta^2 3}{4}\Big(b'(\bar{x}^i_{k|k})b(\bar{x}^i_{k|k})\Big)\Big((b'(\bar{x}^i_{k|k})b(\bar{x}^i_{k|k}))^T\Big) \\ &- \frac{\delta^2}{4}\Big(b'(\bar{x}^i_{k|k})b(\bar{x}^i_{k|k})\Big)\Big(b'(\bar{x}^i_{k|k})b(\bar{x}^i_{k|k})\Big)^T - \bar{x}^{i+1}_{k|k}(\bar{x}^{i+1}_{k|k})^T.\end{aligned} \tag{5.31}$$

To re-iterate, the integrals in (5.30) and (5.31) can be evaluated according to conventional approximate KF methods.

State Correction

State correction by measurement follows that in Section 4.5.

5.3 Filter Implementation

5.3.1 Evaluating the Diffusion coefficient

The proposed filter relies on a model of the Jacobian of the diffusion coefficient, $b'(x(t_i), t_i)$. Calculating this expression analytically is often too complex a task, while numerical approximations can be rather crude. The UTs from the CD-UKF are therefore used to obtain the higher-order linearised matrix (HOLM) associated with the diffusion coefficient, in the same manner as it has been done for the drift and measurement functions in Theorem 4.4.1 and Lemma 4.5.1.

Lemma 5.3.1. *For a Lipschitz continuous function $b(x(t)) \in \mathbb{R}^{n\times 1}$ it follows that*

$$b'(x(t)) + \Omega(x(t)) = b(\chi(t))W\chi^T(t)P(t)^{-1}, \tag{5.32}$$

where $\mathcal{B}(x(t)) = b'(x(t)) + \Omega(x(t))$ is the diffusion coefficient HOLM. The Jacobian of the diffusion coefficient is evaluated at the mean state vector, i.e., $b'(x(t)) = \frac{\partial b(x)}{\partial x}\Big|_{x=\bar{x}(t)}$. The $\Omega(x(t))$ term accounts for non-linear influences introduced by the UTs and is a combination of higher-order state derivatives. Furthermore, $L^TP(t)L > 0$ for $L \in \mathbb{R}^n$ such that $P(t)^{-1}$ exists.

5.3.2 Evaluating the State-Dependent Noise Term

It is shown in [73] that the posterior uncertainty density is correctly propagated along its trajectory by choosing an appropriate value of $\tilde{Q}$ in (5.3) to represent the variance of the local error. Since the SDE used to develop the moment equations in Section 5.2 also makes use of a stochastic local error model it can thus be said that any uncertainties associated with the numerical method will be propagated correctly at each integration time instance according to (5.31). It remains now to determine the exact form of the diffusion coefficient, $b(x(t))$, which is chosen as a function of the local error estimate calculated in Section 3.2 as

$$l_{i+1} = \delta^2 J_i \mathcal{C}_i f(x_i). \tag{5.33}$$

This error estimate provides a more accurate representation of the local error compared to the purely time-varying estimate used in(5.3). It is in the interest of efficiency, however, to avoid the task of calculating $J_i\mathcal{C}_i$

at each integration step. The on-line process can be simplified by computing a scaled equivalent of $J_i\mathcal{C}_i$, for $i = 1, 2, \ldots$, in a case specific manner. The principle is best illustrated using a few examples.

Example 1: The Van der Pol Oscillator

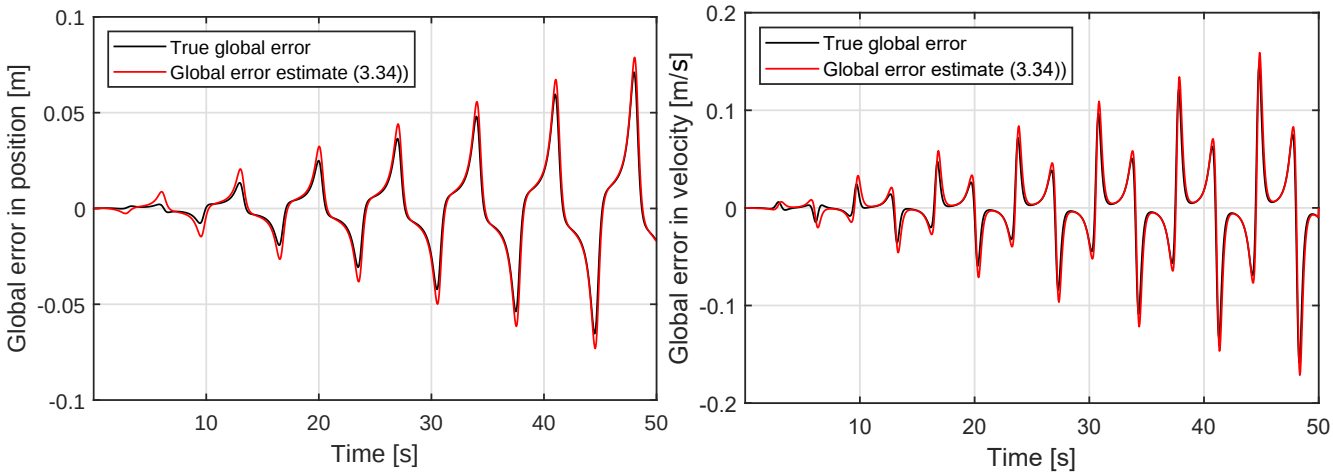
Consider the Van der Pol oscillator in Section 3.3.1. Here its local error estimate, l_i, for the second-order accurate TH scheme was calculated according to (3.41). In order to avoid the numerous operations associated with (3.41), a simpler expression can be approximated. This is done by first calculating the local error estimate in (5.33) as well as the drift function, $f(x_i)$, at various integration step sizes, chosen as $\delta = 0.05$ s, $\delta = 0.1$ s and $\delta = 0.2$ s. The best fit (least-squares) matrix of the $J_i\mathcal{C}_i$ trajectory for each of these integration steps according to the equation $l_i = \delta^2 J_i\mathcal{C}_i f(x_i)$ can then be calculated. This will produce 3 matrices: $(J\mathcal{C})_1$, $(J\mathcal{C})_2$ and $(J\mathcal{C})_3$. A curve can be fit to this set of matrices in Excel® and is found to be of the form $\delta^{1.2}\mathcal{D}$, where

$$\mathcal{D} = \begin{pmatrix} 1.3590 & 0.0216 \\ -0.3372 & 0.5862 \end{pmatrix} \tag{5.34}$$

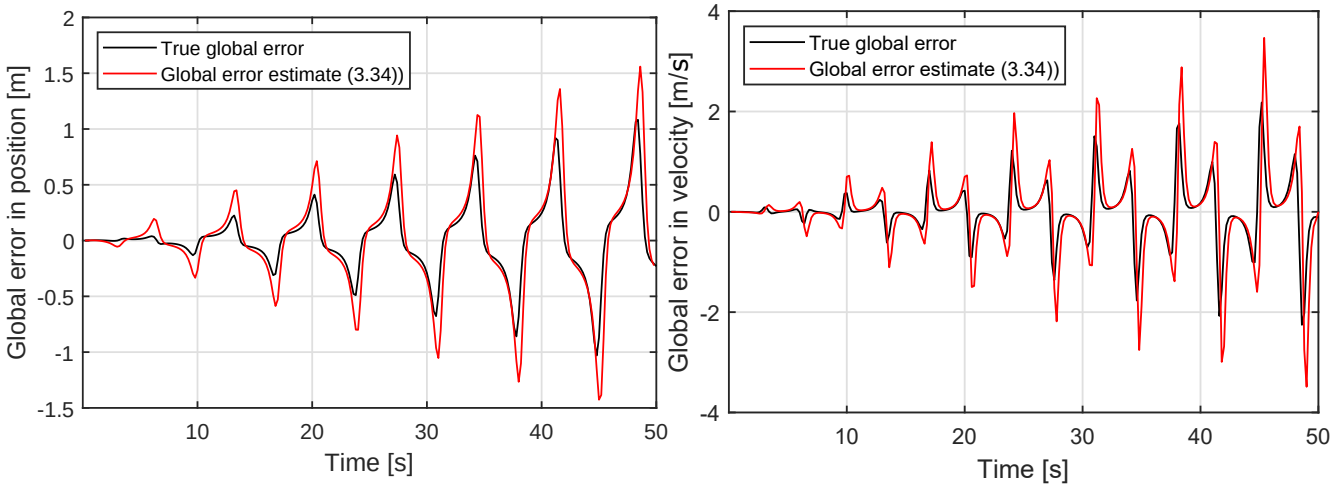
The local error for this example is therefore

$$l_{i+1} = \delta^{3.2}\mathcal{D}f(x_i). \tag{5.35}$$

Using this simplified estimate of the local error, the global error estimates can be calculated using (3.34) and are shown in Fig. 5.3.1.



(a) $\delta = 0.05$ s. **(b)** $\delta = 0.05$ s.



(c) $\delta = 0.2$ s. **(d)** $\delta = 0.2$ s.

Figure 5.3.1: Example 1: Global error in position and velocity for various values of δ using the simplified expression for the local error estimates.

Example 2: The Vertically Falling Body Problem

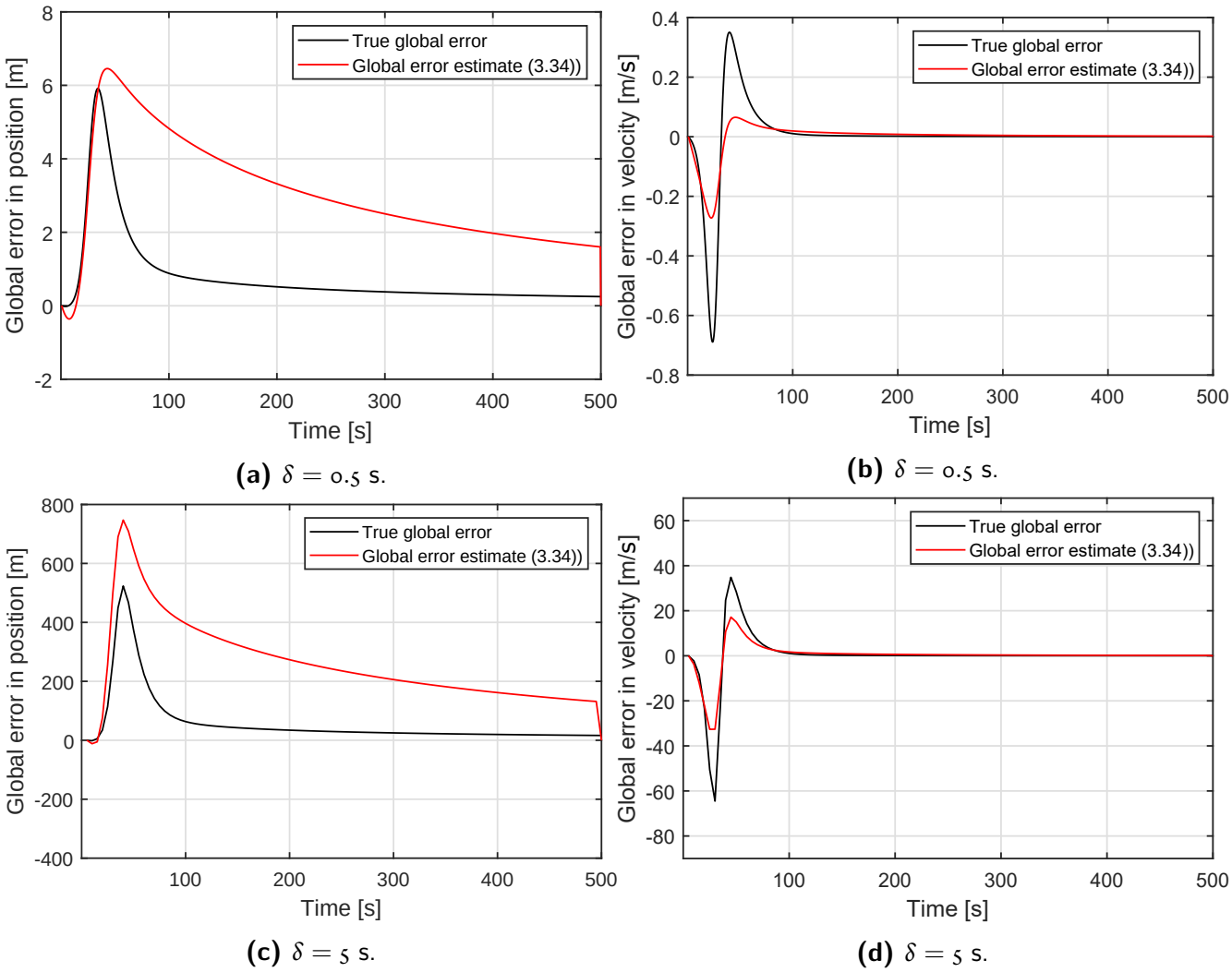
Refer to Section 3.3.2 for the case of the vertically falling body problem. The local error is found to be of the form

$$l_{i+1} = \delta^3 \mathcal{D} f(x_i), \tag{5.36}$$

with

$$\mathcal{D} = \begin{pmatrix} 9.0 \times 10^{-5} & -5.6 \times 10^{-3} \\ 1.4 \times 10^{-5} & 6 \times 10^{-4} \end{pmatrix}$$

The global error estimates are shown in Fig. 5.3.2.



(a) $\delta = 0.5$ s. **(b)** $\delta = 0.5$ s.

(c) $\delta = 5$ s. **(d)** $\delta = 5$ s.

Figure 5.3.2: Example 2: Global error in position and velocity for various values of δ using the simplified expression for the local error estimates.

For the vertically falling body problem, the local error in (5.36) results in slightly less accurate convergence behaviour with respect to position error compared to the behaviour when using the local error estimate in (5.33) (See Fig. 3.4.1). For the remainder of the variables in the two examples, it is evident that the simplified method for calculating the local error estimates results in reasonable global error estimates. Based on the two examples above, the general form of the local error associated with the second-order accurate TH scheme in (5.33) can be approximated as

$$l_{i+1} = \delta^3 \mathcal{D} f(x_i). \tag{5.37}$$

Another example is given below. Although this example does not form part of the class of stiff problems, it is included to demonstrate some features associated with local and global error estimation for a broader range of systems.

Example 3: The Mackey-Glass Time Series

The Mackey-Glass time series is another common non-linear benchmark problem displaying chaotic behaviour for $n > 7$. Its system model is given by

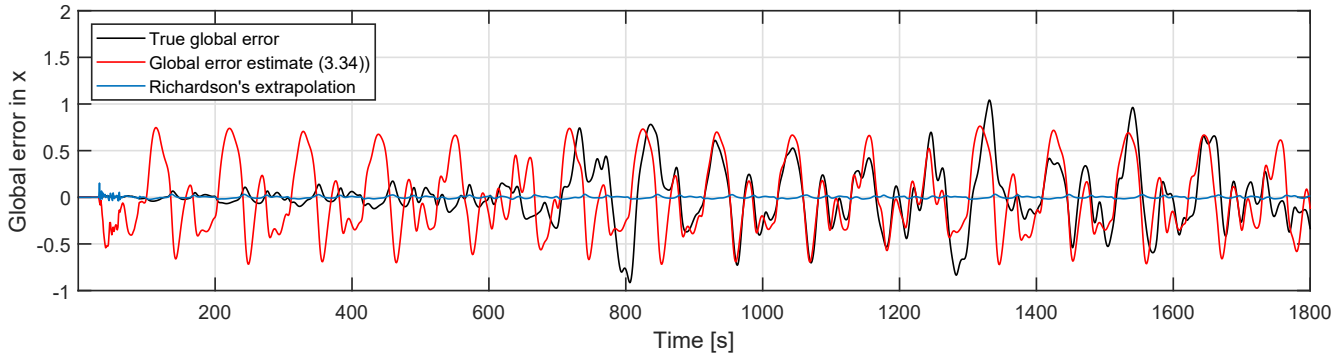
$$dx(t) = \left(\frac{\kappa_1 x(t-\tau)}{1 + x^n(t-\tau)} - \kappa_2 x(t) \right) dt + qw(t), \tag{5.38}$$

$$y(t_k) = x(t_k) + u_k. \tag{5.39}$$

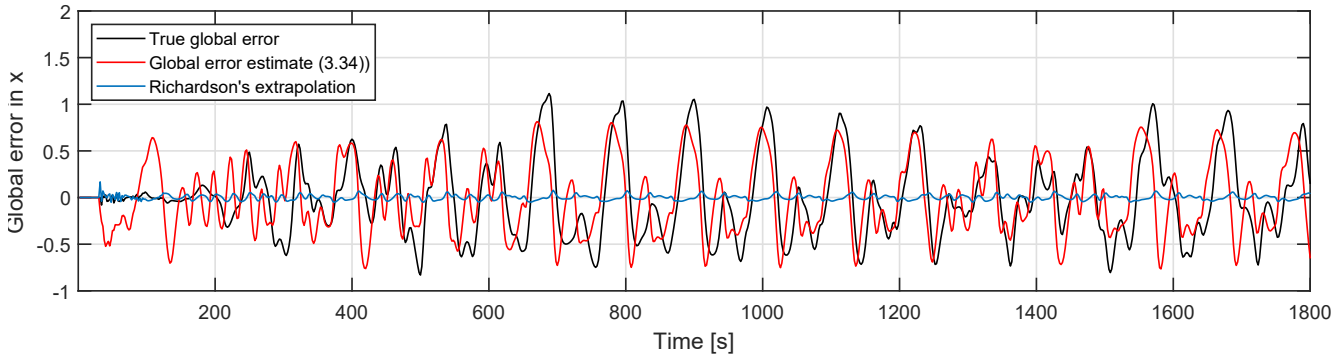
Here $\kappa_1 = 0.22$, $\kappa_2 = 0.1$ and $n = 10$ are positive constants and $\tau = 30$ s is a time delay. Thirty random numbers in the open interval $(0, 1)$ are generated, followed by the calculation of the $J_i C_i$ trajectory for integration step sizes of $\delta = 1$ s, $\delta = 2$ s and $\delta = 6$ s. Following the procedure in Example 1, the local error is found to be of the form

$$l_{i+1} = \delta \mathcal{D} f(x_i). \tag{5.40}$$

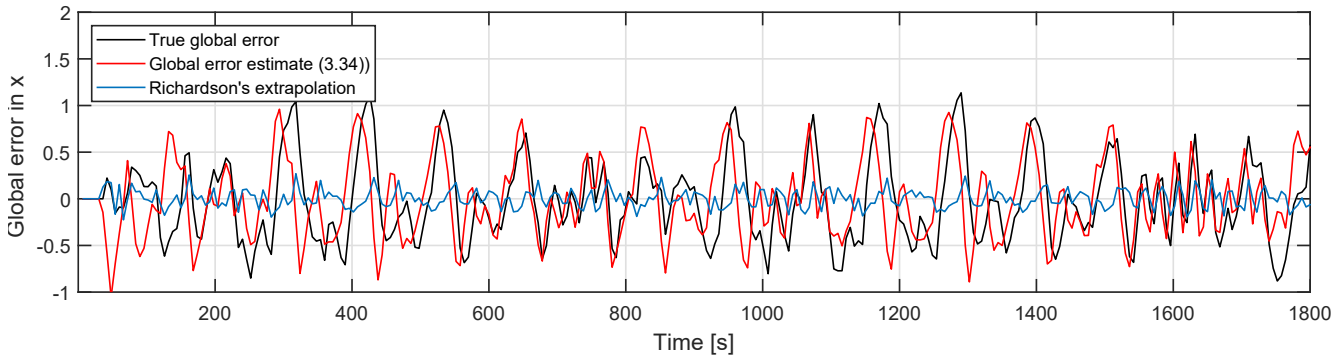
The behaviour of the local error, being proportional to δ and not δ^3, is likely due to the chaotic nature of the problem which results in a drop in the order of accuracy of the method. For $\mathcal{D} = -1.4$ the global error estimates are shown in Fig. 5.3.3 alongside the global error estimates associated with Richardson's extrapolation in (3.39). Due to the chaotic nature of the problem, the global error estimates would also behave chaotically and hence display a global error estimate that does not match the true global error exactly. The order of the global error estimate, however, is correct. Compare this to the poor estimate calculated using Richardson's extrapolation.



(a) $\delta = 1$ s



(b) $\delta = 2$ s



(c) $\delta = 6$ s

Figure 5.3.3: Example 3: Global error for various values of δ using the simplified expression for the local error estimates.

It was shown that the local error estimates for the second-order accurate TH scheme, in the standard non-linear case, is of the form in (5.37). This can be expressed more generally for a scheme of order p as

$$l_{i+1} = \delta^{p+1} \mathcal{D} f(x_i), \tag{5.41}$$

Note that the order of the error is correctly related to the accuracy of the method, i.e., $\mathcal{O}(\delta^{p+1})$, and establishes further confidence in the expression for the local error estimate in (5.33). Assuming that $\mathcal{D}$ forms the bounds of a uniform distribution, its variance is equal to $\frac{1}{\sqrt{3}}\mathcal{D}$. The stochastic numerical error term associated with the

amended SDE in (5.4) is therefore chosen as

$$\frac{\delta^{p+1}}{\delta^{0.5}}\left(\frac{1}{\sqrt{3}}\mathcal{D}\right)f(x_i)dv(t) = \underbrace{\frac{\delta^{p+1/2}}{\sqrt{3}}\mathcal{D}f(x_i)}_{b(x_{i+1})}\,dv(t). \tag{5.42}$$

The only factor in the above expression that has not yet been accounted for is $\delta^{-0.5}$. Since it was chosen that $Q(t) = I_{d_v}\,[s^{-1}]$ for $E(dv(t)dv(t)) = Q(t)dt$, both the dimensions and any scaling associated with $\sqrt{Q(t)}$ are, without a loss in generality, absorbed into $b(x(t))$.

5.3.3 The CD-HCKF_Milstein and CD-HCKF_EM Algorithms

The algorithm for the CD-HCKF_Milstein is outlined in Algorithm 2A and extends from that of the CD-HCKF in Section 4.4.2. Note that explicit state dependence is dropped for notational coherence. Dependence is implied from the context of the algorithm. Steps (1), (2), (3) and (4) of Algorithm 2A are similar to those in Algorithm 1A. In calculating $f(\chi(t_i))$ in step (2), the diffusion coefficient and its HOLM from the previous step, $b(t_{i-1})$ and $\mathcal{B}(t_{i-1})$, are used, in a predictor-corrector type of manner. In step (5), the UT of $b(t_{i+1})$ in (5.42) is calculated as a function of the sigma points for the second-order accurate TH scheme such that

$$\mathcal{L}(t_{i+1}) = \frac{\delta^{2.5}}{\sqrt{3}}\mathcal{D}f(\chi(t_i)). \tag{5.43}$$

The diffusion coefficient is calculated in step (6) as a weighted sum of (5.43) such that

$$b(t_{i+1}) = \mathcal{L}(t_{i+1})w_m. \tag{5.44}$$

In step (7) the diffusion coefficient HOLM is calculated according to (5.32). The state error covariance and state mean at the next time instance is calculated in steps (8) and (9) according to (5.31) and (5.30). The algorithm for the CD-HCKF_EM is easily obtained from the CD-HCKF_Milstein by dropping the terms associated with $b'(x(t))$ and is outlined in Algorithm 3A.

Algorithm 2A: State propagation of the CD-HCKF_Milstein

Initialise with $P(t_{k,o|k})$, $m(t_{k,o|k})$ and $\Xi(t_{k,o|k}) = [\mathbf{o}, \ldots, \mathbf{o}]$. At each step $t_{k,i|k}$; for $i = 1, .., N$; calculate:

1. $\chi(t_{k,i|k}) = [m(t_{k,i|k}), \ldots, m(t_{k,i|k})] + \sqrt{n+\lambda}\,[\mathbf{o},\ \sqrt{P(t_{k,i|k})},\ -\sqrt{P(t_{k,i|k})}]$.
2. $\dot{\chi}(t_{k,i|k}) = f(\chi(t_{k,i|k}))$ and $\dot{m}(t_{k,i|k}) = \dot{\chi}(t_{k,i|k})w_m$.
3. $\mathcal{J}(t_{k,i|k}) = \dot{\chi}(t_{k,i|k})W\chi^T(t_{k,i|k})P(t_{k,i|k})^{-1}$.
4. $\varphi(t_{k,i|k}) = \left(I - \mathcal{J}(t_{k,i|k})\frac{\delta t}{2}\right)^{-1}$ and $M(t_{k,i|k}) = \varphi(t_{k,i|k})\left(I + \mathcal{J}(t_{k,i|k})\frac{\delta t}{2}\right)$.
5. $\mathcal{L}(t_{k,i+1|k}) = \frac{\delta^{2.5}}{\sqrt{3}}\mathcal{D}\dot{\chi}(t_{k,i|k})$.
6. $b(t_{k,i+1|k}) = \mathcal{L}(t_{k,i+1|k})w_m$.
7. $\mathcal{B}(t_{k,i+1|k}) = \mathcal{L}(t_{k,i+1|k})W\chi^T(t_{k,i+1|k})P(t_{k,i|k})^{-1}$.
8. $P(t_{k,i+1|k}) = M(t_{k,i|k})P(t_{k,i|k})M^T(t_{k,i|k}) + \varphi(t_{k,i|k})Q_c(t_{k,i})\varphi^T(t_{k,i|k})\delta t + \delta b(t_{k,i+1|k})b^T(t_{k,i+1|k}) + \frac{\delta^2}{2}\mathcal{B}(t_{k,i+1|k})b(t_{k,i+1|k})\left(\mathcal{B}(t_{k,i+1|k})b(t_{k,i+1|k})\right)^T$.
9. $m(t_{k,i+1|k}) = m(t_{k,i|k}) + \varphi(t_{k,i|k})\dot{m}(t_{k,i|k})\delta + \frac{\delta}{2}\mathcal{B}(t_{k,i+1|k})b^T(t_{k,i+1|k})$.

At time $t_{k,N|k}$, where $t_{k,N|k} = t_{k+1,o|k}$, the *a priori* state vector and its associated covariance matrix are then set to $m(t_{k+1,o|k})$ and $P(t_{k+1,o|k})$. Calculate sigma points $\chi(t_{k+1,o|k})$ as done in step (1) for use in the correction stage of the filter which is the same as that in Algorithm 1B.

Algorithm 3A: State propagation of the CD-HCKF_EM

Initialise with $P(t_{k,o|k})$, $m(t_{k,o|k})$ and $\Xi(t_{k,o|k}) = [\mathbf{o}, \ldots, \mathbf{o}]$. At each step $t_{k,i|k}$; for $i = 1, .., N$; calculate:

1. $\chi(t_{k,i|k}) = [m(t_{k,i|k}), \ldots, m(t_{k,i|k})] + \sqrt{n+\lambda}\,[\mathbf{o},\ \sqrt{P(t_{k,i|k})},\ -\sqrt{P(t_{k,i|k})}]$.
2. $\dot{\chi}(t_{k,i|k}) = f(\chi(t_{k,i|k}))$ and $\dot{m}(t_{k,i|k}) = \dot{\chi}(t_{k,i|k})w_m$.
3. $\mathcal{J}(t_{k,i|k}) = \dot{\chi}(t_{k,i|k})W\chi^T(t_{k,i|k})P(t_{k,i|k})^{-1}$.
4. $\varphi(t_{k,i|k}) = \left(I - \mathcal{J}(t_{k,i|k})\frac{\delta t}{2}\right)^{-1}$ and $M(t_{k,i|k}) = \varphi(t_{k,i|k})\left(I + \mathcal{J}(t_{k,i|k})\frac{\delta t}{2}\right)$.
5. $\mathcal{L}(t_{k,i+1|k}) = \frac{\delta^{2.5}}{\sqrt{3}}\mathcal{D}\dot{\chi}(t_{k,i|k})$.
6. $b(t_{k,i+1|k}) = \mathcal{L}(t_{k,i+1|k})w_m$.
7. $P(t_{k,i+1|k}) = M(t_{k,i|k})P(t_{k,i|k})M^T(t_{k,i|k}) + \varphi(t_{k,i|k})Q_c(t_{k,i})\varphi^T(t_{k,i|k})\delta t + \delta b(t_{k,i+1|k})b^T(t_{k,i+1|k})$.
8. $m(t_{k,i+1|k}) = m(t_{k,i|k}) + \varphi(t_{k,i|k})\dot{m}(t_{k,i|k})\delta$.

At time $t_{k,N|k}$, where $t_{k,N|k} = t_{k+1,o|k}$, the *a priori* state vector and its associated covariance matrix are then set to $m(t_{k+1,o|k})$ and $P(t_{k+1,o|k})$. Calculate sigma points $\chi(t_{k+1,o|k})$ as done in step (1) for use in the correction stage of the filter which is the same as that in Algorithm 1B.

5.4 Case Study

The amended system model for the Mackey-Glass time series from Section 5.3.2 is given by

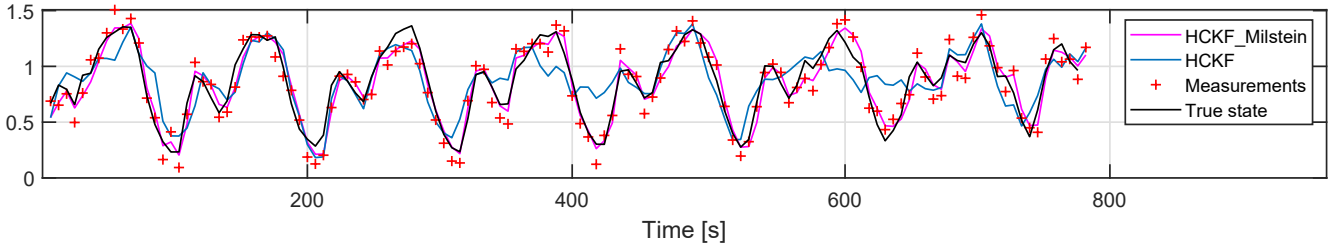
$$dx(t) = \left(\frac{\kappa_1 x(t-\tau)}{1+x^n(t-\tau)} - \kappa_2 x(t)\right)dt + q(t)dw(t) + b(x)dv(t), \tag{5.45}$$

$$y(t_k) = x(t_k) + u_k, \tag{5.46}$$

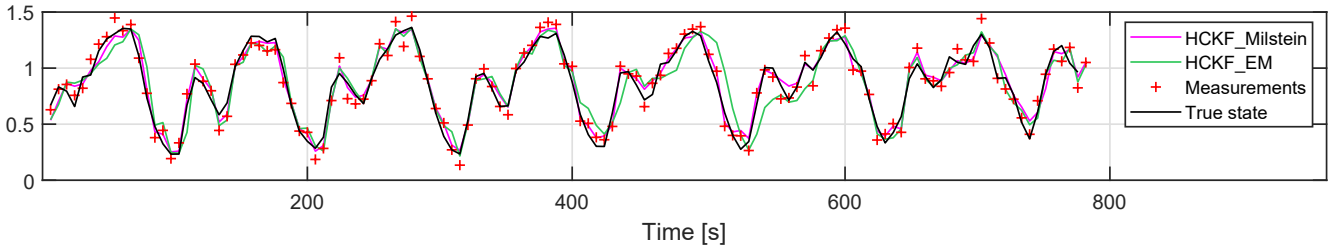
where $w(t) \in \mathbb{R}, v(t) \in \mathbb{R}, b(x) \in \mathbb{R}, \kappa_1 = 0.2, \kappa_2 = 0.1, n = 10, \tau = 30$ s and $q(t) = 0.03\ \text{s}^{-0.5}$ such that $Q_c(t) = q(t)^2 = 0.03^2\ \text{s}^{-1}$. The experiment begins by generating 30 random numbers in the open interval $(0,1)$. True state variables are calculated using a first-order accurate Euler scheme. Filter measurements are formed by sampling the true state variable from a zero mean Gaussian distribution with a standard deviation of 0.08. The measurement covariance is therefore chosen as $R_k = 0.12^2$, assuming that the exact measurement noise distribution is unknown. The state mean for the first 30 time instances of the filter are taken as the initial 30 random numbers that were generated. The initial state error covariance is chosen as $\hat{P}_0 = 0.1^2$. The resultant plots in Figs. 5.4.1a, 5.4.1b, 5.4.1c and 5.4.1d are obtained using the same initial random states in each respective filter according to Algorithms 1A and 1B (the CD-HCKF), Algorithms 2A and 1B (the CD-HCKF_Milstein) and Algorithms 3A and 1B (the CD-HCKF_EM). All three algorithms make use of the second-order accurate TH scheme. In this example the diffusion coefficient associated with the CD-HCKF_Milstein and CD-HCKF_EM must be calculated according to the local error estimate in (5.40) and hence step (5) in Algorithm 2A and 3A is replaced by $\mathcal{L}(t_{k,i+1|k}) = \frac{\delta^{0.5}}{\sqrt{3}}\mathcal{D}\dot{\chi}(t_{k,i|k})$. The plots in Fig. 5.4.1a and 5.4.1b compare the CD-HCKF_Milstein to the CD-HCKF and CD-HCKF_EM, using an integration step size of $\delta = 6$ s. The comparative plots between the CD-HCKF_Milstein and CD-HCKF in Figure 5.4.1c require additional explaining. Here the CD-HCKF includes a diffusion term of the form $b(t)dw(t)$ that accounts for local simulation error, similar to the SDE in (5.3). The constant diffusion coefficient is chosen as $b(t) = 0.013\ \text{s}^{-0.5}$, equivalent to the power in a local simulation error trajectory. Since $b(t)$ is a constant, it can simply be added to $q(t)$ such that $Q_c(t) = (q(t) + b(t))^2$. Returning to a CD-HCKF with no $b(t)$ related term, the plot in Fig. 5.4.1d compares the CD-HCKF_Milstein and the CD-HCKF using a more refined simulation with an integration step size of $\delta = 1$ s. In each of the respective cases an observation is made every 6 s. The average execution time for the respective filters are provided in Table 5.4.1.

	HCKF_Milstein	HCKF_EM	HCKF
$\delta = 1$ s	0.37 s	0.35 s	0.33 s
$\delta = 6$ s	0.12 s	0.12 s	0.10 s

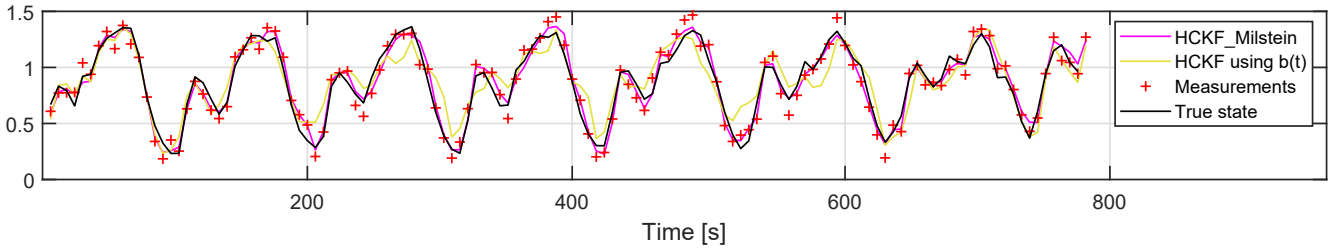
Table 5.4.1: Execution time for the respective CD filters.



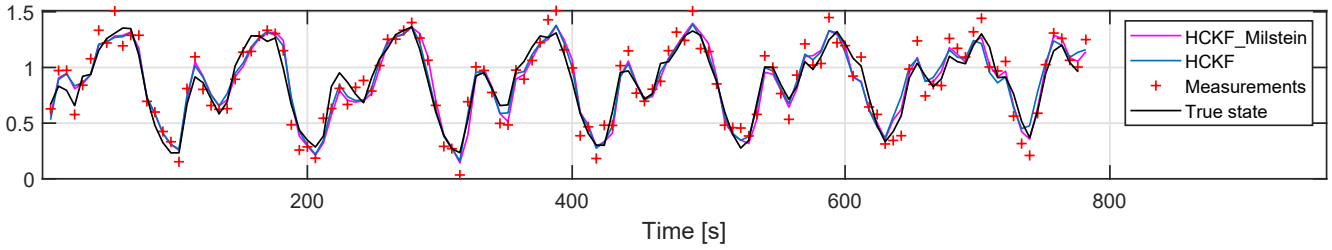
(a) $\Delta = 6$ s, $\delta = 6$ s.



(b) $\Delta = 6$ s, $\delta = 6$ s.



(c) $\Delta = 6$ s, $\delta = 6$ s.



(d) $\Delta = 6$ s, $\delta = 1$ s.

Figure 5.4.1: State trajectories for the respective CD filters.

5.4.1 Discussion

With reference to Figs. 5.4.1a and 5.4.1b, the CD-HCKF_Milstein and CD-HCKF_EM perform better than the CD-HCKF. Since the CD-HCKF_EM factors in the numerical uncertainty via the $b(x)$ related term, it performs better compared to the CD-HCKF that disregards numerical uncertainty. In order to compare the use of a state-dependent numerical error term, $b(x)$, with a constant one, $b(t)$, the CD-HCKF in Fig. 5.4.1c includes a component for numerical error in its covariance, $Q_c(t)$. It can be seen here that the CD-HCKF_Milstein, with a state-dependent error term, outperforms the CD-HCKF that uses a constant numerical error term. With reference to all of the aforementioned plots in this example, it can be said that having a more accurate estimate of the numerical error in the filter, i.e., a state-dependent error estimate $b(x(t))$, improves the filter's

performance. Although the CD-HCKF_EM, deviates slightly from the true state trajectory in Fig. 5.4.1b, it is very similar to the CD-HCKF_Milstein, and is a much better filter approach than the CD-HCKF or the CD-HCKF that uses a constant numerical error estimate. When the integration step size is relatively small, as is the case in Fig. 5.4.1d, and hence the numerical error is relatively small, the CD-HCKF_Milstein reduces to the CD-HCKF as expected. What is important to note in this example is that the CD-HCKF_Milstein and CD-HCKF_EM in Fig. 5.4.1b, which use a relatively large step size, construct estimates almost as well as the filters in Fig. 5.4.1d that employ smaller simulation step sizes. The benefit of taking such huge leaps is that it saves the CD-HCKF_Milstein and CD-HCKF_EM in Fig. 5.4.1b approximately a third of the computation time when compared to the filters in Fig. 5.4.1d (see Table 5.4.1).

5.5 Conclusion

In this chapter, the CD-HCKF_Milstein and the CD-HCKF_EM were developed based on an amended SDE that includes a state-dependent stochastic simulation error term. This error term is modelled based on a cheap method of local error estimation and was found to be proportional to the drift function. The CD-HCKF_Milstein and the CD-HCKF_EM were shown to perform better than the CD-HCKF when significant integration step sizes are employed. This makes sense since a larger step size implies greater numerical uncertainty. Leaving this uncertainty unknown to the filter (in the case of the HCKF) or feeding primitive uncertainty estimates to the filter (in the case of the constant diffusion coefficient) will result in poorer filter performance. The CD-HCKF_Milstein and the CD-HCKF_EM are able to take larger integration steps without losing accuracy, thereby saving computation time. The CD-HCKF_Milstein was shown to be marginally more robust when compared to the CD-HCKF_EM owing to its higher-order of convergence.

Part II

6

State Estimation in Power Plant Boilers

Economic growth, industrialisation and the massive electrification roll-out in 2001 [76] have led to an increase in South Africa's energy consumption over the recent years. The negative environmental impact that comes with meeting these demands has prompted the need for cleaner energy production and improved efficiency in existing plants. Although this has led to proposed plans for more energy to be supplied by the renewable sector [77], majority of South Africa's electricity is still currently being produced by coal-fired plants [78], and recent forecasts, substantiated by the ongoing construction of the combined 9564 MW Medupi and Kusile power plants, implies that this will be the case for (at least) the next 15 years [77]. Efforts must therefore be directed at improving the efficiency and reliability of coal-fired plants. Contrary to such endeavours is the lack of reliable real-time information with respect to state variables that offer key insight into plant behaviour; necessary for monitoring, optimisation, fault detection and control [79]. Important process variables include tube wall temperatures, furnace temperature, heat uptake, steam mass flow rates, etc. Most of these state variables, however, cannot be measured directly and the measured variables that are available, are corrupted with both random and systematic error. High-fidelity models are often employed to obtain such immeasurable state variables. However, the existence of disturbances and a variety of time-dependent unknowns associated with physical plants make their exact behaviour hard to capture in simulation alone. Kalman filtering is a soft sensing approach that has seen many successful applications associated with thermal power plants that began with works in [80], [81], and [82], and extended to more recent applications in fouling and tube leakage detection in [83] and [84] (more on this later). Already, Kalman filtering has become prominent in modern power systems and serve as a means to monitor the current operating state of the network, vital for security analysis functions and energy market valuations [85], [86]. It has consistently shown to be a practical and cost effective approach to construct

internal variable estimates, detect bad data and estimate parameters in complex large-scale systems - all upon which many other functions rely [87].

6.1 Coal-Fired Plants

A typical once-through pulverised coal-fired steam generation unit is, as with other steam-powered plants, based on the Rankine cycle. A single unit consists of several mills into which coal is conveyed before drying and pre-processing. The coal is then pulverised and mixed with primary air and fed into the burners for furnace combustion. Heat from this process is transferred to compressed water from the feedwater pump to produce superheated steam at the exit of the superheater. In this process, thermal energy in the steam is converted to mechanical and then electrical energy in the turbines and generators respectively. From the turbines, the steam goes to a condenser and deaerator and is cycled back to the feedwater inlet. Exhaust gases exiting the furnace are treated and released through the stack and back into the atmosphere. The furnace is supplied combustion air from the forced draught fans while the induced draught fans ensure a negative pressure within the furnace so that combustible vapour does not leak out. The forced and induced draught fans transport flue gas through the convection pass of the boiler and out into the atmosphere.

6.2 Motivation

In 2014 Duvha's unit 3 exploded due to what was reported as an over-pressurisation of the furnace [88]. This was allegedly caused by out of design coal being fed to the boiler that resulted in a residue build up along the tube walls and an oxygen deficit in the furnace. This was further aggravated by a build up of unburnt fuel, the condition of the boiler at the time as well as its then operating practices [88]. An extensive study into reliability centred maintenance at Eskom in [89] has listed creep, long-term overheating, corrosion under deposit, pitting, fly ash erosion, thermal-mechanical fatigue and stress corrosion as a few of the major problems associated with the superheaters in Eskom's fleet. Deposit/scale formation (fouling) is considered the most probable cause of creep, overheating, thermal fatigue and under deposit corrosion/pitting; while poor coal quality is the leading cause of fly ash erosion. It is a well known fact that the calorific value of coal used by the utility has decreased over the years [89], [90], [91]. Low coal quality means that the boiler will have to operate at close to maximum throughput and this leads to increased wear and tear associated with the components of the boiler [90]. In light of these issues, it is recommended that the wear rates of boiler tubes be monitored more frequently, which currently involves a physical inspection on a time scheduled basis. It is thus suggested in [89] that Eskom shift towards condition based monitoring, in place of the time- and failure-based methods currently employed, in order to improve plant reliability. This is echoed in [90] and [91] where fuel mass flow rates and pulverised coal particle size are estimated in order to facilitate a conditioned based monitoring system that can improve combustion efficiency, and in the case of the latter report, indicate the wear levels of certain mill components.

To re-iterate, it has become increasingly important to improve the overall performance, management, safety and reliability practices of the plant in the current climate where coal-fired plants are expected to operate under

stressful conditions, i.e., low grade coal quality and backlogged maintenance plans [89]. The need to (1) detect faults, and (2) monitor, optimise and control plant processes has thus become even more important for safe and cost effective plant operations [79].

6.2.1 Fault Detection

The inability to recognise a fault in time can lead to long term consequences such as hindered performance, reduced reliability, costly damage to the plant and its surrounding, outages and even the loss of lives [92], [84]. Fault detection is categorised into model-based techniques and purely data-driven methods [93]. Typically, model based techniques are synonymous with state estimators such as Kalman Filters and other Bayesian estimators. In [94] caution is suggested when using the standard EKF in severely non-linear boiler environments; however, the EKF is generally preferred for real-time fault detection applications such as reheater tube leakage [84] and fouling detection [95].

Purely data-driven methods (devoid of any physics) are based on colossal amounts of historic plant data and include principle component analysis (PCA), artificial neural networks (ANNs), fuzzy sets, genetic algorithms and k-means clustering, to mention but a few [96]. The downside to such methods are that they typically only consider a fault at a single operating point [84].

6.2.2 Process Monitoring, Optimisation and Control

Increased variation in the energy supply due to disruptive technologies means that coal-fired plants must be made more flexible in order to facilitate an integration of the total energy mix. It is also necessary from a financial and environmental point of view that plants be made more efficient. This is all brought about by optimising plant performance and operations such that it becomes viable to generate electricity, not only during standard operations, but during load variation, low loads and frequent start-up and shut-down [97]. This has led energy industry professionals to turn to advanced analytics and software-based solutions in order to understand their plant better, i.e., computational intelligence (CI), Kalman Filters, etc. Purely data-driven techniques (or CI), however, hold very little physical significance and is thus not able to handle plant disturbances [98]. A model based platform for on-line performance monitoring is documented in [98]. It determines parameters such as the lower heating value of coal as well as heat absorption rates that can be used for plant optimisation. Other variables such as stress [99], creep and fatigue factors [100] can be estimated to ensure that they do not exceed specified thresholds. This further assists with identifying dangerous loads and overall plant optimisation. In the furnace, estimation of the flue gas oxygen content can help optimise boiler combustion [101].

A state observer or estimator is usually designed together with a state feedback controller [102]. Such is the case in [103] where it is shown that fuel flow estimates via Kalman filtering leads to better control. This is reflected in more recent work that uses Kalman filtering to achieve more accurate control (less overshoot and a faster response) [87]. More accurate control compared to conventional PID control is also reported in [104] where Kalman filtered estimates are used as input to the controller. Improved control algorithms such as the

aforementioned help attain maximum plant efficiency and reduces energy wastage [87].

6.2.3 Measured Data

Measured process variables are sent at 10 s intervals to the EtaPro® monitoring database via the object linking and embedding for process control (OPC) communication channel. These measurements include pressure, temperature and mass flow rates at select locations along the boiler. This list also includes, but is not limited to, valve position measurements and excess O_2 and CO_2 levels in the furnace. These measurements are not, however, a comprehensive list that completely identifies the state of the plant. This is because not all critical performance variables can be made available in real-time via this platform due to:

- Physical constraints - Some local process variables are situated in either unfavourable conditions, for example, amidst hot flue gases, or reside in awkward locations, and therefore cannot be measured directly.
- Resource constraints - Certain procedures are costly and do not allow measurements to be made in real-time, such as moisture analysis in coal [91].

To make matters worse, the limited selection of available measurements are subject to drift error, calibration error, telemetry faults, distortion (or random noise) and fluctuation (associated with flow rate measurements) [105] which can make them, as mentioned before, less reliable.

6.2.4 Supervisory Control and Plant Management Systems

The above ties together for use in what is referred to as plant management systems. These exist at the supervisory level where plant optimisation is done. It comprises a single user interface that has access to all technical and commercial data relating to the plant that is used for overall decision making and control instructions. It functions as a support system to the plant operator, pre-processing and interpreting information that will be used to make the final call on desired actions for an optimal outcome. It is also used to determine set-points and inputs of control algorithms. The supervisory layer sometimes incorporates analysis of this information such as expert systems that provide suggested actions. It is necessary for this layer to have openness [106] so that informed decisions can be made using as much information as possible. An on-line state estimator acts as a filter, constructing the most reliable estimate of the state of a system and can form a core constituent of any supervisory program. See [107] where this has been implemented.

6.3 Existing Plant State Estimators

Related endeavours in state estimation of a thermal power plant include the following: In [108] an application of a constant gain EKF is made to an Advanced Gas-cooled Reactor (AGR) once-through boiler. The constant gain EKF consists of 34 temperature and specific enthalpy states across its length (economiser, evaporators and superheaters) and 4 output measurements. Here a full EKF is avoided due to its lengthy computational process.

An interesting result in this study shows that adding fictitious covariance results in better performance than augmenting the state vector with a bias term (creating a bias filter). Furthermore, 'input-ignorant' filtering does not produce good state estimates (due to the high non-linearity of the distributed boiler models). Correct boiler inputs are thus necessary for accurate estimates. In [109] the temperature, total water volume, steam quality and steam volume of the drum of a drum-type boiler is estimated using a UKF. The UKF is chosen to preserve some of the non-linearity in the model and to avoid difficult Jacobian calculations. It uses temperature and drum level as measurements in the filter. The second investigation in the paper estimates the pressure and entropy of the primary superheater (2 state variables in total) using temperature and pressure measurements from the superheater. Since the superheater is described by a set of PDEs (not compatible with KFs) the study proposes the use of a correction factor in the estimates. In both investigations most of the estimates are shown to converge to their true values, i.e., the true state variables fell within 2 standard deviations, $2\sqrt{\hat{P}}$, of the estimates. Estimates such as the drum temperature and total water volume, however, displayed a slight bias, likely due to model error. In [80] an EKF which includes a bias state (to estimate unknown parameters or disturbances) is designed for use with the current superheater temperature controller of an oil-fired plant. The filter comprises 5 specific enthalpy states associated with the superheater. Specific enthalpies are chosen as the state variables (instead of temperatures) because of the highly non-linear relationship between temperature and pressure. This biased KF based temperature controller shows improved disturbance rejection compared to the current operational PID controller. In [110] a UKF is applied to a coal-fired drum boiler. The boiler's measurable variables, drum pressure and drum water level, are used as filter observation data to estimate unmeasurable variables - the quality of vapour at the riser outlet and the volume of steam in the drum. The estimator is tested in simulation and is found to track the true state variables well and successfully helps meet the objectives of the proposed controller. In [111] an EKF is designed to estimate 3 specific enthalpy states (associated with the evaporator in a once-through Benson boiler) and a heat disturbance which is then used in a feedwater supply controller. Specific enthalpy is chosen as the state variable in order to facilitate more accurate steam control. The control algorithm using the EKF was shown to display improved performance over conventional control methods used in the plant. More computationally demanding state estimators have been used in boiler applications such as that in [112], where 400 ensemble points were found to estimate drum level and drum vapour temperature with good accuracy. In [113] a moving horizon estimator (an optimisation-based estimator) is applied to the final superheater of a once-through power boiler. Twenty states of the superheater (comprising steam temperatures and tube wall temperatures of its 10 segments) as well as an unmeasured disturbance in the flue gas are estimated successfully. However, the optimisation problem greatly increased the computation time when compared to other non-linear filters. Optimal estimation of the moisture content in coal for a coal-fired boiler using the EKF is presented in [114]. The estimator uses models of the furnace of the order found in [115]. Three sets of data taken from the mill are fed into the filter, where the third set is associated with a mill blockage. The estimator successfully estimates the moisture content in all three cases and is hence able to reveal the fault in the third data set. A model of a coal supply system is developed in [116] and used in an EKF to estimate coal flow in the furnace using oxygen concentration and total air flow measurements. When used as input to a furnace simulation, the coal flow estimates are reported to be more accurate when compared to those from an open-loop furnace

simulation. Coal flow is again estimated using an EKF in [117] but the filter uses measurements of the fluid temperature in the furnace water walls instead of oxygen concentration measurements in the furnace. Here a detailed real-time simulator of the coal-fired plant is used as a test bench to verify the estimates. A comparison between estimates of the density and enthalpy of furnace gas for a linear and extended KF of a tangentially-fired coal furnace is shown in [79]. The temperature and pressure of the furnace gas are used as measurements. The EKF is shown to produce superior estimates compared to those from the LKF (which uses a linearised state-space model of the furnace). Details of an operational mill estimator is given in [103].

In the above mentioned applications, the devised LKF, EKF and UKF estimators, limited to a few states (a maximum of 4 in the UKF applications), were deemed suitable for future on-line implementation. They were all tested using measurements from the actual plant and a sufficiently detailed plant simulation. In some cases, estimates were verified indirectly through improved controller performance.

6.4 Problem Statement

The lack of transparency surrounding certain processes in South African power plants signals the need for more sophisticated (and commercially viable) methods to produce a set of accurate variable estimates across the boiler length such as tube wall temperature and fluid specific enthalpy. These variables, although they do not form a comprehensive list, are critical to monitoring, fault detection and control of the boiler. Monitoring of the plant is done on a time- and failure-finding-bases - a rather outdated approach. Furthermore, no pre-processing of measurements or variables are considered before being fed into other functions such as feedback controllers and commercial forecasting software. Large-scale non-linear on-line state estimation of over a hundred state variables is, however, acknowledged as a difficult task.

6.5 Hypothesis

Using the Kalman filter algorithms in Part 1 of the thesis, centralised on-line state estimation of an excess of a hundred state variables is technically (and commercially) feasible.

6.6 Objectives

The objectives of Part 2 are to:

- Develop suitable MATLAB® and Flownex® models of the reference once-though utility power boiler (detailed in Chapter 7).
- Apply the techniques in Part 1 to help design a state estimator for the boiler and then demonstrate its feasibility to on-line implementation (the core of Chapter 8).

6.7 Methodology

Two separate models of the boiler are developed, one as a mathematical MATLAB® model with state differential equations formulated from first principles, the other, a Flownex® model comprising an assembly of base components with built-in numerical solvers to obtain the solution of the dynamic network represented by a more detailed set of partial differential equations (PDEs). The models are based on technical plant information that has been redacted to hide confidential details. The set of CD-KFs formulated in Part 1 are developed around the simpler mathematical MATLAB® model. Performance of the MATLAB® model is validated against the PDE-based Flownex® model as well as historic plant data. The Flownex® model, subject to faults and disturbances, is made to represent the 'real' plant from which observation data can be constructed and fed to the estimator. Internal variables associated with the 'real' plant, not available in actuality, can then be compared to estimates generated by the filter and used to evaluate its performance. Once the estimator fulfils the necessary specifications, simulated observation data is replaced with historic plant data to illustrate its utility.

6.8 Conclusion

Plant operators alone cannot detect faults or inconsistencies in a complex, large-scale system. It is thus recommended that all input parameters have verified anticipated effects. This means that all measurements should be checked for consistency against internal plant processes and other measurements. This can be done systematically, using available measurements and underlying plant dynamics to construct the best possible estimate of the state of the boiler that measured data or simulation alone is not able to capture, i.e., through Kalman filtering. Kalman filters can easily be integrated into current plant systems and existing infrastructure. Furthermore, they are cost-effective and offer a simple and convenient means to obtain state estimates that can be used at the supervisory level for condition-based monitoring, plant optimisation and control.

7

Once-Through Boiler Modelling

This chapter provides a brief description of the boiler unit of interest and details two separate models for it. The first is a simple mathematical model in MATLAB® intended for use in the state estimator and the second a more sophisticated model built in Flownex®. The Flownex® model initially serves as validation for the mathematical model and is used in later chapters to construct filter observation data. The model boundary is taken at the burner where coal enters the furnace to the last stage superheater exiting to the turbine. The model uses a set of energy and mass balance equations and a set of input boundary conditions from plant data to determine the dynamics of both the water- and dry-side of the boiler. It is intended that a better understanding is gained regarding the variables involved in this process through state estimation. In order to do this, the mathematical model is required to have sufficient spatially discretised segments to correctly capture the distributed behaviour along the main boiler tube. Since each segment is associated with more than one variable of interest (tube wall temperature and fluid specific enthalpy), the estimator will have twice as many state variables as discrete model segments. To simplify the modelling process and reduce model stiffness, it is assumed that the time derivative of pressure is zero, while the spatial gradient of pressure is calculated based on load.

7.1 Boiler Unit 1 of Plant A

For the sake of confidentiality, exact values of variables and parameters of the boiler are withheld or normalised where necessary. The normalised quantity will typically be expressed as $\frac{\text{actual variable value}}{\text{base variable value}}$. Boiler unit 1 of Plant A is one of four 600 MW wet-cooled once-through boiler units. Once-through boilers are capable of operating at either sub-critical or supercritical pressure. Above the critical pressure of water, no boiling occurs and the

critical fluid is converted directly to superheated steam. Older Benson® type boilers, such as those of Plant A, are typically sub-critical and are less efficient than supercritical steam generators.

7.1.1 Furnace and Pulverising Mill

The vertical spindle pulverising mill consists of the primary air fan (heated via the air heater and extracted via the air duct) which releases warm primary air that mixes with ground coal from the coal pulveriser. See Fig. 7.1.1. The temperature inside the mill is controlled such that moisture in the coal is evaporated as the pulverised coal and primary air mix makes its way to the burners. The evaporated moisture leaves the burners together with the primary air and pulverised coal. At standard outlet temperature and pressure, the mills are designed for coal with a worst-case moisture content of 1.16 times higher than the nominal. Anything higher than this will result in excess water vapour in the furnace which would consume lots of energy and change combustion dynamics. The flow rate of the output coal and air stream is dependent on several factors. A few of them being the moisture content of coal and primary (and seal) air. The boiler consists of multiple burners situated in the side walls of the furnace, forced- and induced draft fans and primary air fans. Combustion takes place inside the furnace with the help of secondary air flow, producing a sizeable flame. Radiation produced from combustion is the dominant source of heat for the furnace water walls and contributes lesser to heat absorption in the rest of the main pressure components the higher up in the boiler they are situated.

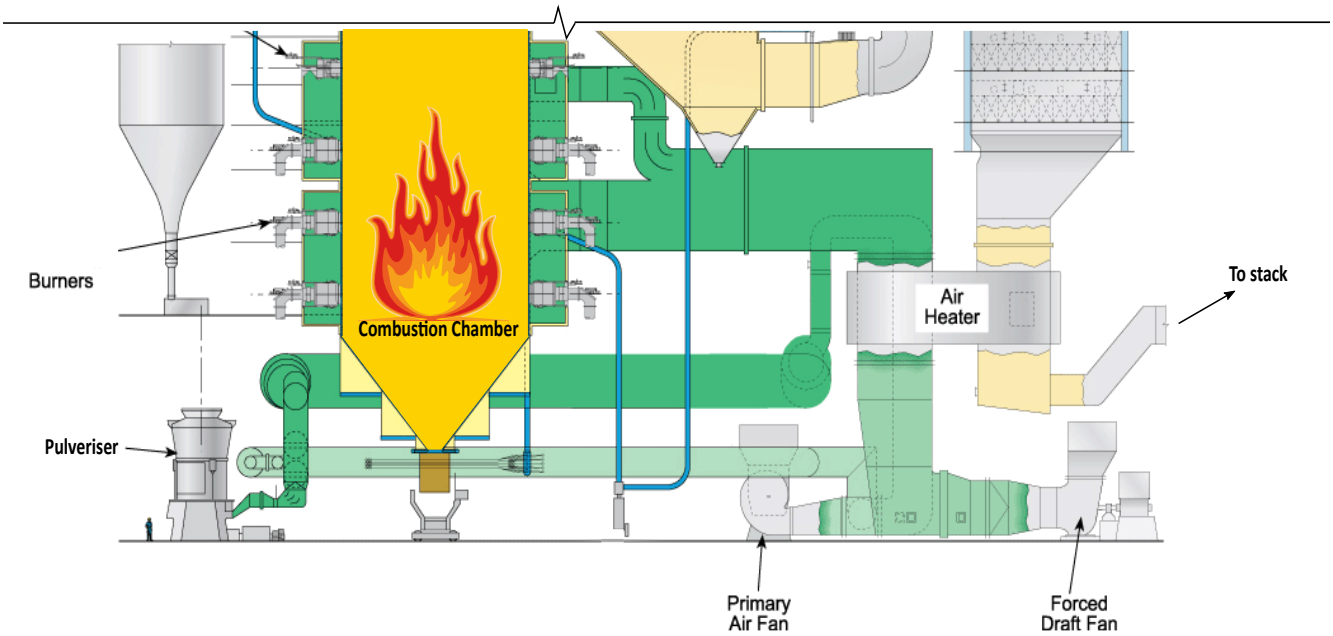
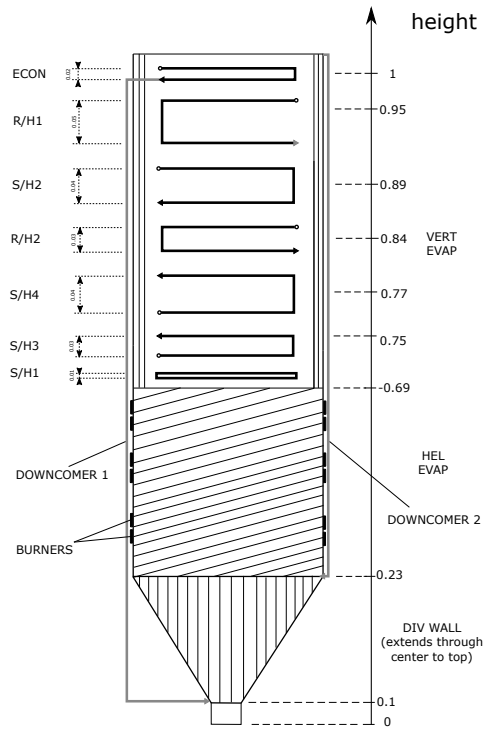
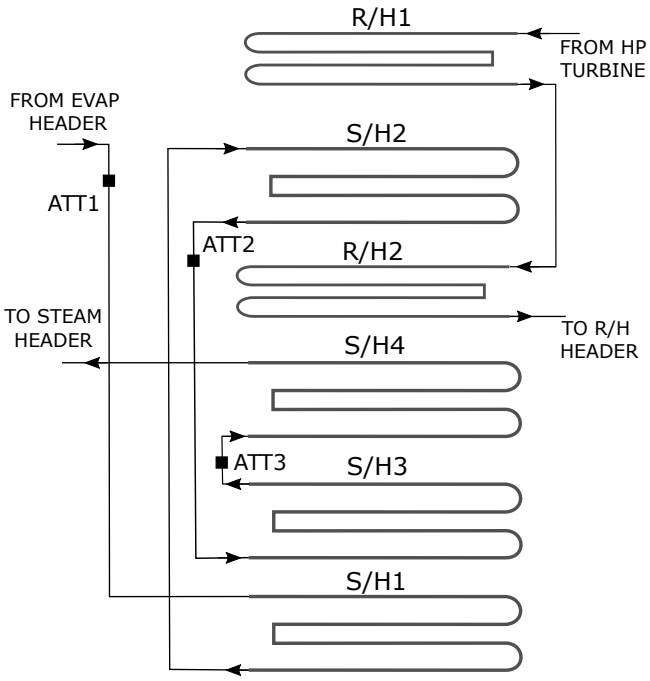


Figure 7.1.1: A diagram showing the pulveriser, furnace, air heater, and fans. Adapted from [1].



(a) Rear view of the combustion chamber showing the position of the burners. Height measurements have been normalised according to the maximum boiler height.



(b) A diagram of the convection pass showing the configuration of the superheaters, reheaters and attemperators.

Figure 7.1.2

7.1.2 Convection Pass

With reference to Fig.7.1.2a, the convection pass consists of 4 superheaters, 2 reheaters and an economiser, each made up of several parallel tubes that form a series of cross-flow heat exchangers. These parallel tubes are fed from, and terminate into, headers that balance mild temperature and pressure variability amidst the individual tubes of the various convection elements (or boiler elements in general). A cross-flow configuration implies that flue gasses from the furnace pass perpendicularly through the fluid-carrying tubes and has the advantage over a parallel flow arrangement of better heat transfer due to its large surface area per unit volume. It includes the class of heat exchangers referred to as compact heat exchangers.

The first-stage superheater converts saturated (wet) steam into superheated (dry) steam, while the remainder of the superheaters increase the temperature of the dry superheated steam. The boiler has 4 horizontal superheaters in an unmixed cross-flow configuration with the flue gas. S/H3 (the third superheater in the series) and S/H4 are in a parallel-flow arrangement, and S/H1 and S/H2 a counterflow arrangement with the flue gas (see Fig. 7.1.2b). The benefit of a counterflow arrangement is a more uniform heat transfer rate across the heat exchanger which maximises its heat transfer ability. This is particularly important in the first-stage superheater which is subject to variations in radiation originating from the furnace. Amidst the superheaters are 2 horizontal reheaters, responsible for heating the expanded superheated steam from the high-pressure turbine outlet in

order to reduce moisture content detrimental to the turbines.

Three injection attemperators (ATT3, ATT2 and ATT1) are configured in a cascaded control loop at the exit of S/H3, S/H2 and the vertical evaporator, and are responsible for regulating the steam temperature out of the superheater. Each attemperator comprises multiple individual sub-attemperators positioned at the respective sub-headers. At the outlet of the vertical evaporator are a total of 2 sub-attemperators whereas the remainder of the attemporators have 4 sub-attemperators respectively. The attemperators are fed using feedwater from the economiser.

The gilled economiser is fed demineralised water (forced by the feed pump) from the high pressure heater which is then heated to saturation temperature (to become saturated water) before entering the downcomer. The downcomer then takes the saturated water to the entrance of the division-wall.

7.1.3 Division Wall

The division wall is responsible for additional heat uptake in the boiler and vertically splits the entire boiler into two equal parts, ending in two headers above the economiser. The inside of vertical riser tubes are typically riffled internally to avoid the onset of film boiling and departure from nucleate boiling. Fluid from the division wall then goes to the second downcomer which takes the fluid to the entrance of the helical evaporators.

7.1.4 Evaporators

The evaporators constitute both helical (furnace waterwalls) and vertical tubes. The fluid at the exit of downcomer 2 goes to the main bottom ring header and into the smooth and bare helical tubes which then terminate into 4 headers. The main source of heat transfer in the helical evaporator is via radiation originating from the furnace combustion process. The slight inclination in the helical tubes facilitates the movement of fluid through the tubes and increases the overall heat transfer area. From the headers, the fluid goes to the vertical evaporator.

7.2 Model 1: MATLAB® Boiler model

This section details the 1-dimensional lumped-parameter mathematical model of the boiler that results in a characteristic set of initial value problems. Its boundary conditions include the economiser inlet fluid mass flow rate, pressure and temperature; the respective reheaters inlet temperature, pressure and mass flow rates; the vertical and helical evaporator heat rate; the coal mass flow rate; the combustion supply air mass flow rate and temperature; the flue gas pressure and mass flow rate; and the attemperators' spray water mass flow rates. The model is built from knowledge of heat flow from the inner tube wall to the fluid (internal convection), heat flow from the flue gas to the outer tube wall (external forced convection), the effective exchange between the internal and external fluid (heat exchangers), furnace behaviour (combustion) and the flow of heat and mass through the length of the tubes (tube and fluid dynamics). These are detailed below.

7.2.1 Internal Convection

Understanding the transfer of heat and mass through the water-side of the boiler (a collection of parallel fluid streams) begins with calculating the convection heat transfer from one of the many circular inner tube walls to the fluid. Its various regions and corresponding heat transfer regimes are shown in Fig. 7.2.1. A mathematical formulation of each region is provided below.

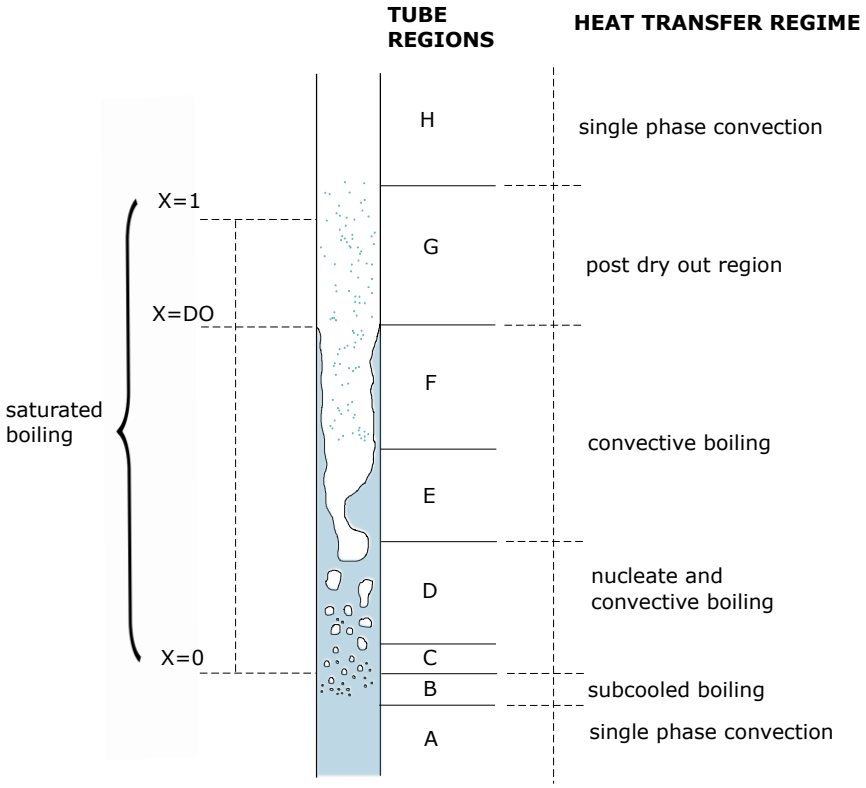


Figure 7.2.1: A graphical representation of the various flow regions inside the boiler tubes, following [2].

Single-Phase Convection: Region A and H

During single-phase convection, associated with either pure water or steam, the rate of heat transfer into or out of the fluid is given by the general expression

$$\dot{q}_f = \mathbb{h}_f(T_m - T_f), \tag{7.1}$$

where T_m is the temperature of the inner tube wall, T_f is the temperature of the fluid and $\mathbb{h}_f$ is the convection heat transfer coefficient of the fluid [118]. The ratio of the convection to conduction heat transfer is referred to as the Nusselt number and is given by $Nu_f = \frac{\mathbb{h}_f d_i}{\lambda_f}$, where d_i is the inner tube wall diameter and λ_f is the thermal

conductivity of the fluid. This gives

$$\dot{q}_f = \frac{\lambda_f}{d_i} Nu_f (T_m - T_f), \tag{7.2}$$

with

$$\mathbb{h}_f = \frac{Nu_f \lambda_f}{d_i}. \tag{7.3}$$

An important fluid property is the Prandtl number, defined as

$$Pr_f = \frac{\eta_f c_{p,f}}{\lambda_f}, \tag{7.4}$$

where $c_{p,f}$ is the specific heat capacity of the fluid and η_f is the viscosity of the fluid. Another property of a fluid is the Reynolds number, defined as the ratio of inertial to viscous forces. For internal flow within a circular tube, the Reynolds number is

$$Re_f = \frac{\rho_f u_{m,f} d_i}{\eta_f}, \tag{7.5}$$

where $u_{m,f}$ is the mean fluid velocity over its cross section and ρ_f is the fluid density. For fully developed flow the onset of turbulence occurs at $Re_f \approx 2300$. Boilers operate in the transitional turbulence region between $2300 < Re_f < 10000$. Substituting the fluid mass flow rate $\dot{m}_f = \rho_f u_{m,f} A_x$ and tube cross sectional area $A_x = \frac{\pi d_i^2}{4}$ into (7.5) gives

$$Re_f = \frac{4\dot{m}_f}{\pi d_i \eta_f}. \tag{7.6}$$

For turbulent or transition regions over the ranges $0.5 < Pr_f < 2000$ and $3000 < Re_f < 5 \times 10^6$ the Gnielinski correlation can be used to calculate the Nusselt number[118]:

$$Nu_f = \frac{\frac{\xi}{8} Pr_f (Re_f - 1000)}{1 + 12.7\sqrt{\frac{\xi}{8}}(Pr_f^{2/3} - 1)}, \tag{7.7}$$

with friction factor

$$\xi = (2log\left(\frac{d_i}{k}\right) + 1.14)^{-2}, \tag{7.8}$$

where $k = 40$ µm is the inner tube surface roughness coefficient [119].

Subcooled and Saturated Nuclear Boiling: Region B, C, D, E and F

For fully developed subcooled boiling the heat transfer rate can be calculated using the Jens and Lottes correlation (1951) [120],[2]:

$$\Delta T_{sat} = 25(\dot{q}_f)^{0.25} \exp\left(\frac{-p_f}{62}\right), \tag{7.9}$$

where ΔT_{Sat} is the degree of superheat defined as $\Delta T_{Sat} = T_m - T_{sat}$, T_{sat} is the saturation temperature of the fluid and p_f is the fluid pressure in bar. This equation is valid for 7 bar $< p_f <$ 172 bar, 115° C $< T_f <$ 340 °C, 11 kg/m²s $< \dot{m}_f/A_x <$ 1.05 $\times$ 10⁴ kg/m²s and $\dot{q}_f/A_x <$ 12.5 MW/m². The heat transfer mechanism in the saturated nucleate boiling region is the same as that in the subcooled region. As a result, the correlation in (7.9) remains valid subject to the same aforementioned conditions and is given by

$$\Delta T_f = 25(\dot{q}_f)^{0.25} \exp\left(\frac{-p_f}{62}\right), \tag{7.10}$$

where $\Delta T_f = T_m - T_f$. The heat transfer rate in W with p_f in MPa is therefore $\dot{q}_f = 2.56\Delta T_f^4 \exp(p_f/1.55)$.

Post Dryout: Region G

The heat transfer rate in the post dryout region is given by the Groeneveld Correlation (1973) [121], [2]:

$$\dot{q}_f = \left(\frac{\lambda_f Nu_f}{d_i(T_m - T_f)}\right)^{1/0.888}, \tag{7.11}$$

where

$$Nu_f = a\left[Re_f\left(x + \frac{\rho_{st}}{\rho_l}(1-x)\right)\right]^b Pr_{f,w}^c Y^d \tag{7.12}$$

and

$$Y = 1 - 0.1\left(\frac{\rho_l}{\rho_{st}} - 1\right)^{0.4}(1-x)^{0.4}, \tag{7.13}$$

where ρ_l is the density of liquid in the fluid, ρ_{st} is the density of steam in the fluid, x is the fluid quality and $Pr_{f,w}$ is the Prandtl number at wall temperature. The constants in (7.12) are given in Table 7.2.1.

Table 7.2.1: Groenewald correlation constants

a	b	c	d
1.09 $\times 10^{-3}$	0.99	1.41	-1.15

Critical Heat Flux

When the heat transfer rate is equal to the critical heat flux (CHF), either departure from nucleate boiling (DNB) or dryout occurs, depending on whether the fluid is in a subcooled or saturated state. The various heat

transfer regimes in relation to the critical heat flux is summarised in Fig. 7.2.2. Film boiling is an anomaly that rarely occurs in standard boiler operations. Its heat transfer regime is therefore not explained further. Empirical lists of CHF data will be used is the model to determine whether the fluid has undergone dryout.

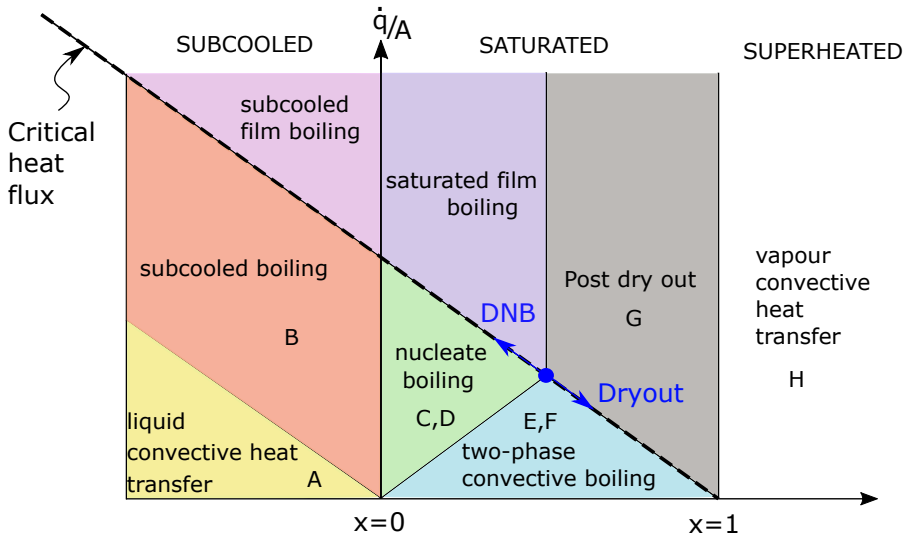


Figure 7.2.2: A graphical representation of the various heat transfer regimes inside a circular tube, following [2]. The vertical axis represents heat flux and the horizontal axis represents steam quality.

7.2.2 External Forced Convection Over Tube Bundles

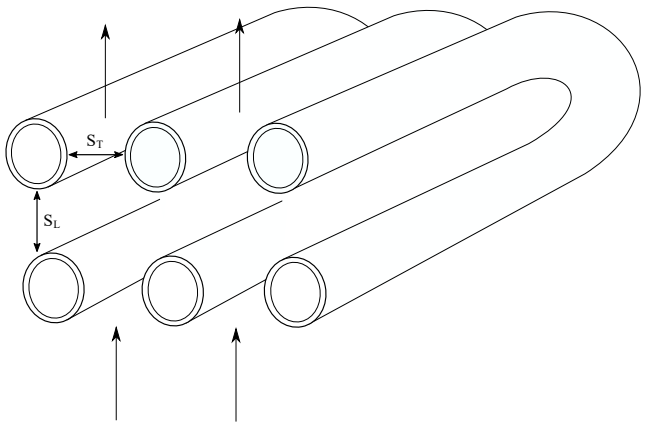


Figure 7.2.3: Parallel tubes in a cross-flow configuration. The arrows represent the flow of flue gas.

It follows now to compute the heat transfer behaviour of a set of parallel circular tubes in cross-flow with the flue gas. Here the flue gas flows perpendicular to the direction of fluid as shown in Fig. 7.2.3. The Reynolds

number of the flue gas is given by

$$Re_g = \frac{\rho_g v_g D_H}{\mu_g}, \tag{7.14}$$

where ρ_g, v_g and μ_g is the density, velocity and viscosity of the flue gas, while D_H is the hydraulic diameter given as

$$D_H = \frac{4 L_g A_{ff}}{A_s}, \tag{7.15}$$

where A_s is the collective surface area of the tube bundle, L_g is the distance travelled by the flue gas and A_{ff} is the free-flow area of the bundle. The maximum velocity for aligned tubes is given as

$$v_{g,max} = \frac{S_T - d_o}{S_T} v_{g,ave}, \tag{7.16}$$

where $v_{g,ave} = \frac{\dot{m}_g}{\rho_g A_{ff}}$ is the incoming average velocity of the flue gas and S_T is the transverse pitch associated with the tube bundles (see Fig. 7.2.3). Various correlations for the Nusselt number have been formulated such as those by Churchill and Bernstein or Zukauskas. In this case, the Hilpert correlation is used and has Nusselt number [122] for $Pr_g \gtrsim 0.7$:

$$Nu_g = C Re_g^m Pr_g^{1/3}. \tag{7.17}$$

The constants in the model are chosen as $C = 0.193$ and $m = 0.618$ for a flue gas Reynolds number which is usually in the range $4000 < Re < 40000$ according to [118]. The Prandtl number is

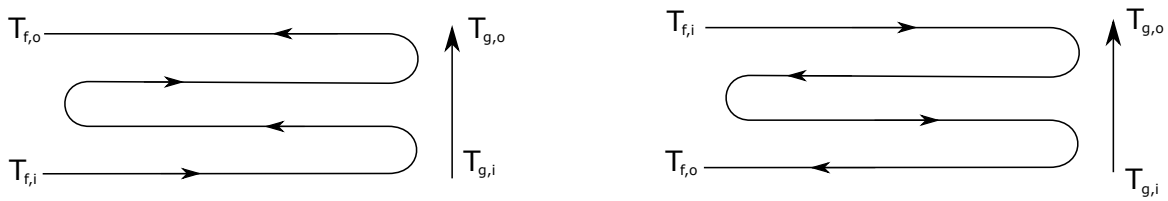
$$Pr_g = \frac{\mu_g c_{p,g}}{\lambda_g}, \tag{7.18}$$

where $c_{p,g}$ is the specific heat capacity and λ_g is the thermal conductivity of flue gas. Finally, the Stanton number is calculated as $St = \frac{Nu_g}{Re_g Pr_g}$ and the Colburn factor due to convection as $j_{H,con} = St\ Pr_g^{2/3}$. The effects of radiation on the heat transfer coefficient is added to the Colburn factor due to convection such that the total Colburn factor is $j_H = j_{H,conv} + j_{H,rad}$ where $j_{H,rad}$ is based on empirical data for similar boiler units [123]. The heat transfer coefficient for external flow associated with the flue gas is therefore

$$\mathbb{h}_g = \frac{j_H\ \rho_g v_g c_{p,g}}{Pr_g^{2/3}}. \tag{7.19}$$

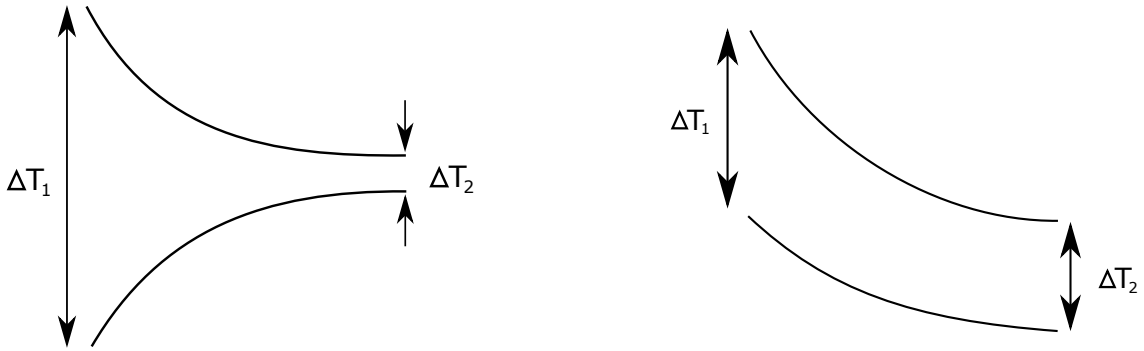
7.2.3 Heat Exchangers

The total heat available to the tubes can now be calculated according to the theory of heat exchangers. Consider the two distinct cross-flow arrangements in Figs. 7.2.4a and 7.2.4b. In either case, the overall heat transfer rate



(a) Parallel-flow. **(b)** Counter-flow.

Figure 7.2.4: Heat exchangers in a cross-flow configuration.



(a) Parallel-flow. **(b)** Counter-flow.

Figure 7.2.5: Temperature profiles of heat exchangers in a cross-flow configuration.

can be calculated in multiple ways:

$$\dot{q} = \dot{m}_g c_{p,g}(T_{g,i} - T_{g,o}), \tag{7.20}$$

$$\dot{q} = \dot{m}_f c_{p,f}(T_{f,i} - T_{f,o}), \tag{7.21}$$

$$\dot{q} = UA\Delta T, \tag{7.22}$$

where all temperatures are expressed in the mean sense over an area A, U is the overall heat transfer coefficient and ΔT is the difference in temperature between the hot and cold fluid. The temperature difference for the parallel-flow heat exchanger in Fig. 7.2.4a with temperature profile in Fig. 7.2.5a is

$$\Delta T_1 = T_{g,i} - T_{f,i} \tag{7.23}$$

$$\Delta T_2 = T_{g,o} - T_{f,o}. \tag{7.24}$$

The temperature difference for the counter-flow heat exchanger in Fig. 7.2.4b with temperature profile in Fig. 7.2.5b is

$$\Delta T_1 = T_{g,i} - T_{f,o} \tag{7.25}$$

$$\Delta T_2 = T_{g,o} - T_{f,i}. \tag{7.26}$$

Effectiveness NTU Method

According to the effectiveness NTU method the flue gas temperatures as it passes through the various stages of convection pass can be calculated using the current fluid temperatures and inlet flue gas temperature (to the first superheater). The maximum heat rate across a heat exchanger is

$$\dot{q}_{max} = C_{min}(T_{g,i} - T_{f,i}),$$

where $C_{min} = \min[\dot{m}_g c_{p,g}, \dot{m}_f c_{p,f}]$. The effectiveness of a heat exchanger, $\varepsilon = \frac{\dot{q}}{\dot{q}_{max}}$, is the ratio of its actual to maximum heat rate. The effective heat rate across the heat-exchanger can therefore be calculated as

$$\dot{q} = \varepsilon C_{min}(T_{g,i} - T_{f,i}). \tag{7.27}$$

For a bundle of concentric tube heat exchangers in cross-flow its effectiveness for either configuration is:

$$\text{parallel-flow:} \qquad \varepsilon = \frac{1 - \exp(-\text{NTU}(1 + C_r))}{1 + C_r} \tag{7.28}$$

$$\text{counter-flow:} \qquad \varepsilon = \frac{1 - \exp(-\text{NTU}(1 - C_r))}{1 - C_r \exp(-\text{NTU}(1 - C_r))}, \tag{7.29}$$

where $C_r = \frac{C_{min}}{C_{max}}$ is the ratio of the minimum to maximum heat capacity and NTU is the number of transfer units given as

$$\text{NTU} = \frac{UA}{C_{\text{min}}}. \tag{7.30}$$

The effectiveness can be used to calculate the flue gas and fluid temperatures at the outlet of each convective element by combining (7.20) or (7.21) with (7.27) to give

$$\varepsilon = \frac{C_g(T_{g,i} - T_{g,o})}{C_{min}(T_{g,i} - T_{f,i})} \qquad \text{OR} \qquad \varepsilon = \frac{C_f(T_{f,o} - T_{f,i})}{C_{min}(T_{g,i} - T_{f,i})} \tag{7.31}$$

7.2.4 Overall Heat Transfer Coefficient

The overall heat transfer coefficient of the heat exchanger needed in (7.30) considers the heat transferred from the flue gas to the outer pipe wall, through the wall to the inner tube wall and to the steam. For tubular heat

exchangers, the overall heat transfer coefficient U is calculated according to

$$\frac{1}{UA_o} = \frac{1}{\mathbb{h}_f A_i} + \frac{\ln\left(\frac{d_o}{d_i}\right)}{\lambda_m 2\pi L} + \frac{1}{\mathbb{h}_g A_o}, \tag{7.32}$$

where $A_i = \pi d_i L$ and $A_o = \pi d_o L$ is the inner and outer tube wall surface area, L is the length of the tube and λ_m is the thermal conductivity of the tube. The heat transfer coefficients, $\mathbb{h}_f$ and $\mathbb{h}_g$, are calculated according to (7.3) (or (7.1)) and (7.19).

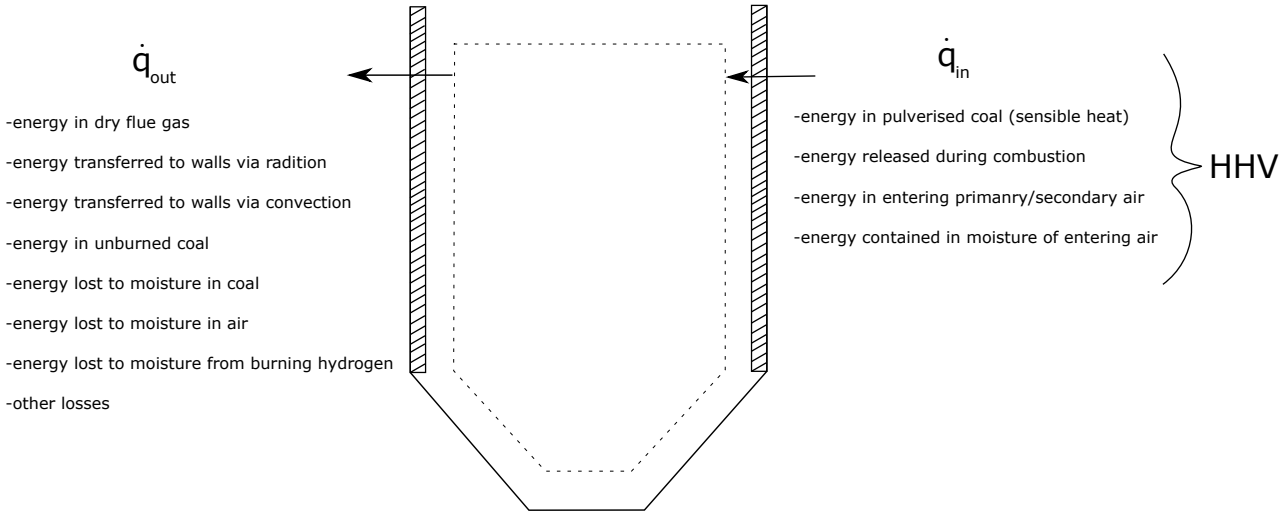


Figure 7.2.6: Energy balance inside the furnace. The boundary is depicted by the dotted line.

7.2.5 Combustion

The temperature of the flue gas entering the superheater series in (7.27) and (7.31) is calculated according to a combustion process. With reference to Fig. 7.2.6 a boundary can be drawn inside the furnace corresponding to the following energy balance

$$m_g c_{p,g} \frac{dT_g}{dt} = \dot{q}_{in} - \dot{q}_{out}, \tag{7.33}$$

where m_g is the mass of the flue gas and $c_{p,g}$ is its specific heat capacity. The factors contributing to heat transferred across the boundaries are summarised in Fig. 7.2.6. The total energy into the system can be taken as the higher heating value (HHV) - the net or ideal energy released during combustion. Given an ultimate analysis of coal, the HHV is given by Dulong's formula [124] in brackets and multiplied by the mass flow rate of pulverised coal

$$\text{HHV} = \left(33960\text{C} + 141890(\text{H} - \frac{\text{O}}{8}) + 9420\text{S}\right) \dot{m}_{pf} \quad [\text{kJ/s}], \tag{7.34}$$

where C,H,O and S are the mass fractions of carbon, hydrogen, oxygen and sulphur in coal and $\dot{m}_{pf}$ is the mass flow rate of pulverised coal. It is assumed here that the temperature of the pulverised coal is the same as that

of the combustion air supply temperature. The energy lost through the boundary, summarised in Fig. 7.2.6, is quantified in [124] and [115] and presented below.

Energy Transfer to Waterwalls

The heat transferred to the waterwalls of the furnace due to radiation can be expressed as the mass flow rate of fluid multiplied by the change in enthalpy of the fluid in the furnace waterwalls and the feedwater

$$\dot{q}_{rad} = \dot{m}_{f,1}(h_{f,1} - h_{f,2}) \quad [\text{kJ/s}], \tag{7.35}$$

where $\dot{m}_{f,1}$ is the mass flow rate of fluid, $h_{f,1}$ is the average fluid specific enthalpy in the furnace waterfalls and $h_{f,2}$ is the feedwater specific enthalpy.

Energy Lost to Dry Flue Gas

The energy lost due to high flue gas temperature in the stack as a consequence of higher than standard burner supply temperature is given as

$$\dot{q}_{dfg} = \dot{m}_{pf} M_g c_{p,dfg}(T_{f,1} - T_{c,2}) \quad [\text{kJ/s}], \tag{7.36}$$

where $T_{f,1}$ is the flue gas temperature at the final superheater outlet, $T_{c,2}$ is the combustion supply air temperature (primary and secondary air), $M_g = \frac{\text{mass of dry flue gas}}{\text{mass of pulverised coal}}$ is the mass ratio of flue gas to pulverised coal and $c_{p,dfg}$ is the specific heat capacity of dry flue gas.

Energy Lost to Moisture Formed From Burning Hydrogen

The energy used to vaporise moisture in the exhaust gases as a result of burning hydrogen in coal is given by

$$\dot{q}_{hf} = 9\text{H}\dot{m}_{pf}(h_{f,3} - h_{f,4}) \quad [\text{kJ/s}], \tag{7.37}$$

where $h_{f,3}$ is the specific enthalpy of superheated steam at a low pressure of 10 kPa (0.01 MPa) and $h_{f,4}$ is the specific enthalpy of saturated liquid at coal supply temperature. In this case the supply temperature of coal corresponds roughly to a water saturation (vapour) pressure of 8.6 kPa, from which $h_{f,4}$ can be calculated.

Energy Lost to Moisture in Fuel

In a similar fashion as above, the energy used to vaporise inherent and surface moisture in coal is given by

$$\dot{q}_{mf} = \frac{\mathcal{M}}{1 - \mathcal{M}}\dot{m}_{pf}(h_{f,3} - h_{f,4}) \quad [\text{kJ/s}], \tag{7.38}$$

where $\mathcal{M}$ is the mass fraction of moisture in coal.

Energy Lost to Vaporisation of Moisture in the Combustion Air Supply

Again, the energy used to vaporise moisture in the combustion air supply is

$$\dot{q}_{ma} = M_a M_v \dot{m}_{pf}(h_{f,3} - h_{f,5}) \quad [\text{kJ/s}], \tag{7.39}$$

where M_a is the mass fraction of air to pulverised coal, M_v is the mass fraction of moisture in dry air and $h_{f,5}$ is the specific enthalpy of superheated steam at air supply temperature and water vapour partial pressure. In order to calculate $h_{f,5}$ the water vapour partial pressure at air supply temperature is determined, from which the vapour saturation enthalpy at this pressure can be obtained using the IAPWS formulation.

The mass fraction of air to coal can be calculated using the mass fractions of particular elements. With the fraction of oxygen O_2 in the combustion chamber known and the fraction of O_2 in standard air being 20.95 % then

$$M_a = \frac{\text{O}_2\text{in supply air} \times \text{C in coal} \times (1/\text{O}_2 \text{ in standard air})}{\text{C in supply air}}, \tag{7.40}$$

where C in supply air $= \frac{12}{44}\text{CO}_2 + \frac{12}{28}\text{CO}$. Here CO_2 and CO are the mass fractions of carbon dioxide and carbon monoxide in supply air.

Energy Lost to Unburned Coal

The energy lost due to incomplete combustion is

$$\dot{q}_{uc} = \dot{m}_{pf}\text{HV}_{\text{CO}}\frac{28\text{C}}{12}\frac{\text{CO}}{\text{CO} + \text{CO}_2} \quad [\text{kJ/s}], \tag{7.41}$$

where HV_{CO} is the heating value that results from burning CO to CO_2 and is given roughly as $\text{HV}_{\text{CO}} = 10114$ kJ/kg.

Other Losses

Other losses include heat that is lost to the surroundings due to radiation and convection heat transfer as well as the energy lost to the ash hopper. Since these losses are relatively small in comparison to other losses, they will not be included in the energy balance calculations.

Energy Balance

The heat rate out of the boundary can now be calculated as:

$$\dot{q}_{out} = \dot{q}_{rad} + \dot{q}_{dfg} + \dot{q}_{hf} + \dot{q}_{mf} + \dot{q}_{ma} + \dot{q}_{uc}, \tag{7.42}$$

and the heat rate into the boundary as:

$$\dot{q}_{in} = \dot{q}_{pa} + \dot{q}_{sa} + \dot{q}_{mo} + HHV, \tag{7.43}$$

where the input energy rate due to primary air, secondary air and moisture in the air is given by

$$\dot{q}_{pa} = c_{p,pa}\dot{m}_{pa}T_{pa}$$
$$\dot{q}_{sa} = c_{p,sa}\dot{m}_{sa}T_{sa}$$
$$\dot{q}_{mo} = c_{p,mo}\mathcal{M}\dot{m}_{pf}T_{pa},$$

where $c_{p,pa}$, $c_{p,sa}$, $c_{p,mo}$, T_{pa}, T_{sa}, $\dot{m}_{pa}$ and $\dot{m}_{sa}$ are the specific heat capacity of primary air, the specific heat capacity of secondary air, the specific heat capacity of moisture, the temperature of primary air, the temperature of secondary air, the mass flow rate of primary air and the mass flow rate of secondary air, respectively. From (7.42) and (7.43), the flue gas temperature dynamics in (7.33) can be solved. A first-order Euler approximation was found to suffice in the integration step size range of $\delta < 5$ s.

7.2.6 Attemperators

The fluid at the exit of the evaporator and superheaters are controlled via attemperation as depicted in Fig. 7.2.7a for ATT2 (and similarly for ATT1 and ATT3). Here feedwater is injected into the fluid to reduce its temperature. The mass flow of the spray water is controlled using cascaded PI controllers. To simplify the mathematical modelling process, the spray water temperature and its mass flow rate are taken as inputs to the model according to a basic energy balance:

$$h_{f,out}(\dot{m}_f + \dot{m}_{sw}) = h_{f,in}\dot{m}_f + h_{sw}\dot{m}_{sw}, \tag{7.44}$$

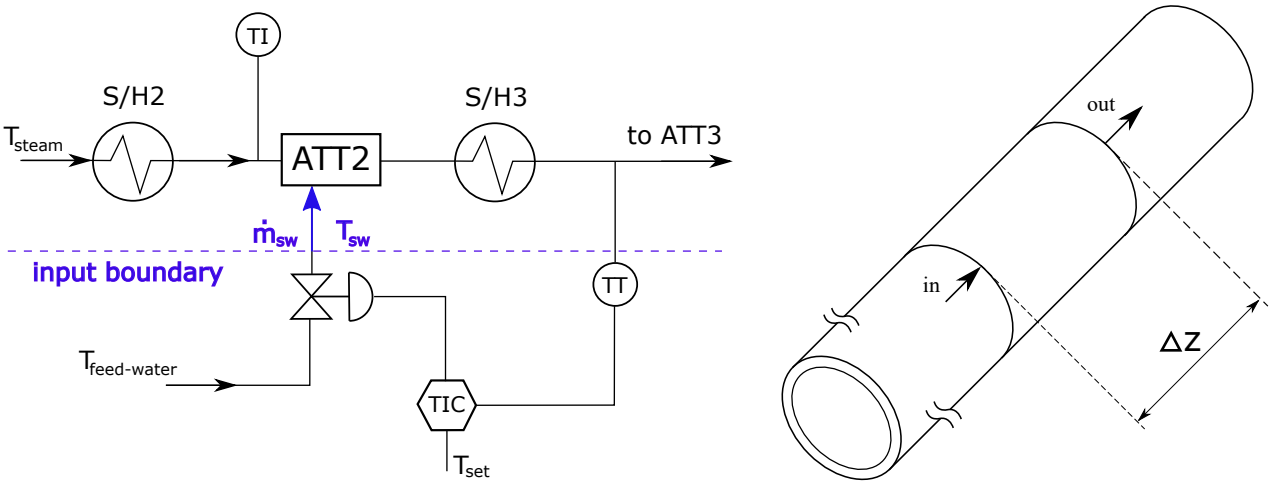
where $h_{f,in}$ and $h_{f,out}$ are the fluid specific enthalpies before and after attemperation and $\dot{m}_{sw}$ is the spray water mass flow rate. From this, the effective temperature of the fluid after attemperation can be expressed as

$$T_{f,out} = \frac{\dot{m}_f T_{f,in} - \dot{m}_{sw}\big(\Delta h/c_{p,f} + (c_{p,sw}/c_{p,f} - 1)T_{sat} - (c_{p,sw}/c_{p,f})T_{sw}\big)}{\dot{m}_{f,in} + \dot{m}_{sw}}, \tag{7.45}$$

where Δh is the latent heat of evaporation at feedwater temperature, $c_{p,sw}$ and $c_{p,f}$ are the specific heat capacities of the spray water and fluid, and T_{sat} is the saturation temperature of water at feedwater pressure.

7.2.7 Tube and Fluid Dynamics

It follows now to determine the properties of interest associated with the state estimator, i.e., the rate of change of tube wall temperature and fluid enthalpy found in [33]. Consider the constant volume discrestization of a single tube in Fig. 7.2.7b. The economiser, division wall, helical evaporator and vertical evaporator are discretized into 10 segments each, and S/H1, S/H2, S/H3 and S/H4 into 16, 4, 4 and 4 segments respectively. The number of discrete segments per boiler element is chosen according to its length such that $\Delta Z < 100$ m. All parallel tubes along the boiler length are effectively lumped into a single tube, thus losing information regarding any axial variation that might exist in the tube and fluid dynamics. The heat energy of each segment of volume V



(a) Simplified P&ID for ATT2 showing the model boundary. **(b)** Discretisation scheme of a single tube.

Figure 7.2.7: (a) Simplified P&ID for ATT2 showing the model boundary. (b) Discretization scheme of a single tube.

is increased by the total heat available to the heat-exchanger, $\dot{q}$, in (7.27), and decreased by the heat rate into the fluid, $\dot{q}_f$, so that

$$\begin{aligned} \frac{d(m_m c_{p,m} T_m)}{dt} &= \dot{q} - \dot{q}_f, \\ \therefore \frac{dT_m}{dt} &= \frac{\dot{q} - \dot{q}_f}{m_m c_{p,m}}, \end{aligned} \tag{7.46}$$

where m_m, T_m and $c_{p,m}$ are the mass, temperature and specific heat capacities of the tube wall segment. The fluid's internal heat energy, $\rho_f V h$, where h is the fluid's specific enthalpy, is increased by the heat rate carried by the fluid into each segment such that

$$\begin{aligned} \frac{d(\rho_f V h_{out})}{dt} &= \dot{m}_{in} h_{in} - \dot{m}_{out} h_{out} + \dot{q}_f, \\ \therefore \rho_f V \frac{dh_{out}}{dt} + h_{out} \frac{d(\rho_f V)}{dt} &= \dot{m}_{in} h_{in} - \dot{m}_{out} h_{out} + \dot{q}_f, \end{aligned} \tag{7.47}$$

where $\dot{m}$ is the fluid mass flow rate. According to the conservation of mass

$$\frac{d\rho_f V}{dt} = \dot{m}_{in} - \dot{m}_{out}. \tag{7.48}$$

Combining (7.47) and (7.48) followed by some algebraic manipulation gives

$$\rho_f V\frac{dh_{out}}{dt} + h_{out}(\dot{m}_{in} - \dot{m}_{out}) = \dot{m}_{in}h_{in} - \dot{m}_{out}h_{out} + \dot{q}_f$$
$$\therefore \rho_f V\frac{dh_{out}}{dt} = \dot{m}_{in}h_{in} - \dot{m}_{in}h_{out} + \dot{q}_f$$
$$\therefore \frac{dh_{out}}{dt} = \frac{\dot{m}_{in}(h_{in} - h_{out}) + \dot{q}_f}{\rho_f V}. \tag{7.49}$$

Since the tube volume is fixed, the conservation of mass equation can be written as

$$V\frac{d\rho_f}{dt} = \dot{m}_{in} - \dot{m}_{out},$$
$$\dot{m}_{out} = \dot{m}_{in} - V\frac{d\rho_f}{dt}. \tag{7.50}$$

Since density is a function of both pressure and specific enthalpy, where p_f is known and h can be calculated, its time derivative (follows from the chain rule) and can be written explicitly as

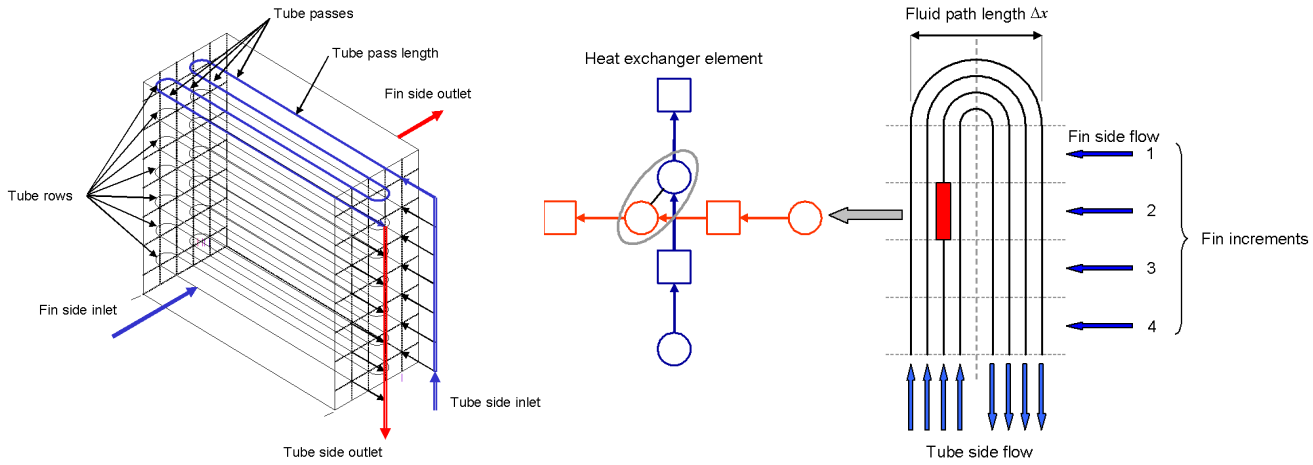
$$\frac{d\rho_f}{dt} = \frac{\partial\rho_f}{\partial h}\frac{dh}{dt} + \frac{\partial\rho_f}{\partial p_f}\frac{dp_f}{dt}. \tag{7.51}$$

Substituting (7.51) into (7.50) gives the mass flow rate out of each tube segment:

$$\dot{m}_{out} = \dot{m}_{in} - V\left(\frac{\partial\rho_f}{\partial h}\frac{dh}{dt} + \frac{\partial\rho_f}{\partial p}\frac{dp}{dt}\right). \tag{7.52}$$

7.3 Model 2: Flownex® Boiler Model

Flownex® is a commercial thermal-hydraulic simulation environment based on sophisticated computational fluid dynamic (CFD) design methodologies that is used in this section to develop a complementary boiler model to that in Section 7.2. Flownex® is based on a 1-dimensional flow network that is solved, either dynamically or in steady-state, according to a set of partial differential equations. The model features an interactive prompt that allows for changes from between 60 - 100 % of the boiler's maximum continuous rating (MCR). Plant data is used to adjust the set of input boundary conditions at various loads. The boundary conditions include the economiser fluid inlet pressure and temperature; S/H4 outlet fluid mass flow rate; the respective reheaters' inlet fluid temperature and pressure and outlet mass flow rates; the vertical evaporator heat rate; the coal mass flow rate; the combustion supply air mass flow rate and temperature; and the flue gas pressure and mass flow rate. The model is grouped into various sub-systems, namely the convection pass, the downcomers, the division wall, the evaporators and the furnace. Their individual modelling is discussed in what follows.



(a) Details of a standard heat exchanger. *source: Flownex® manual.*

(b) A single pass heat exchanger showing individual elements. *source: Flownex® manual.*

Figure 7.3.1: (a) Details of a standard heat exchanger. (b) A single pass heat exchanger showing individual elements

7.3.1 Convection Pass

The convection elements (consisting of the economiser, 4 superheaters and 2 reheaters) and flue gas system is modelled using respective finned-tube heat exchanger components. They are made to represent a conventional heat exchanger with flue gas flowing perpendicularly over the collection of tubes as shown in Fig. 7.3.1a. The heat exchanger is divided into several tube passes and rows and further discretised in the direction of the gas flow as shown in Fig. 7.3.1b. A single row heat exchanger in Fig. 7.3.1b would therefore result in a total of 32 discrete segments (or smaller heat exchanger elements). The model of the convection pass is shown in Fig. 7.3.2. The series of heat exchanger elements on the left represent external flue gas flow (dry-side) while those on the right represent the internal fluid flow within the boiler tubes (water-side). The dry-side elements have convective heat transfer profiles formulated using empirical data from similar boiler units. The heat profiles incorporates radiative heat transfer by scaling the characteristic curves accordingly. The further away from the furnace the heat exchanger is, the less radiative heat absorption occurs. The convection pass has three attemperators, one at the exit of the evaporator, ATT1, one at the exit of S/H2, ATT2, and the other at the exit of S/H3, ATT3. They are modelled using advanced PID controller components with integral and constant parameters taken from site data. Fluid temperature is used as the process value and the spray water mass flow as the controlled output. The set-points are fixed according to plant data. The attemperators can be replaced by spray water inputs as was done in the MATLAB® model.

7.3.2 Downcomers and Division wall

The downcomers are included in model 2 to facilitate more accurate pressure dynamics (not present in the MATLAB® model). Downcomer 1 takes the fluid from the outlet of the economiser to the inlet of the division

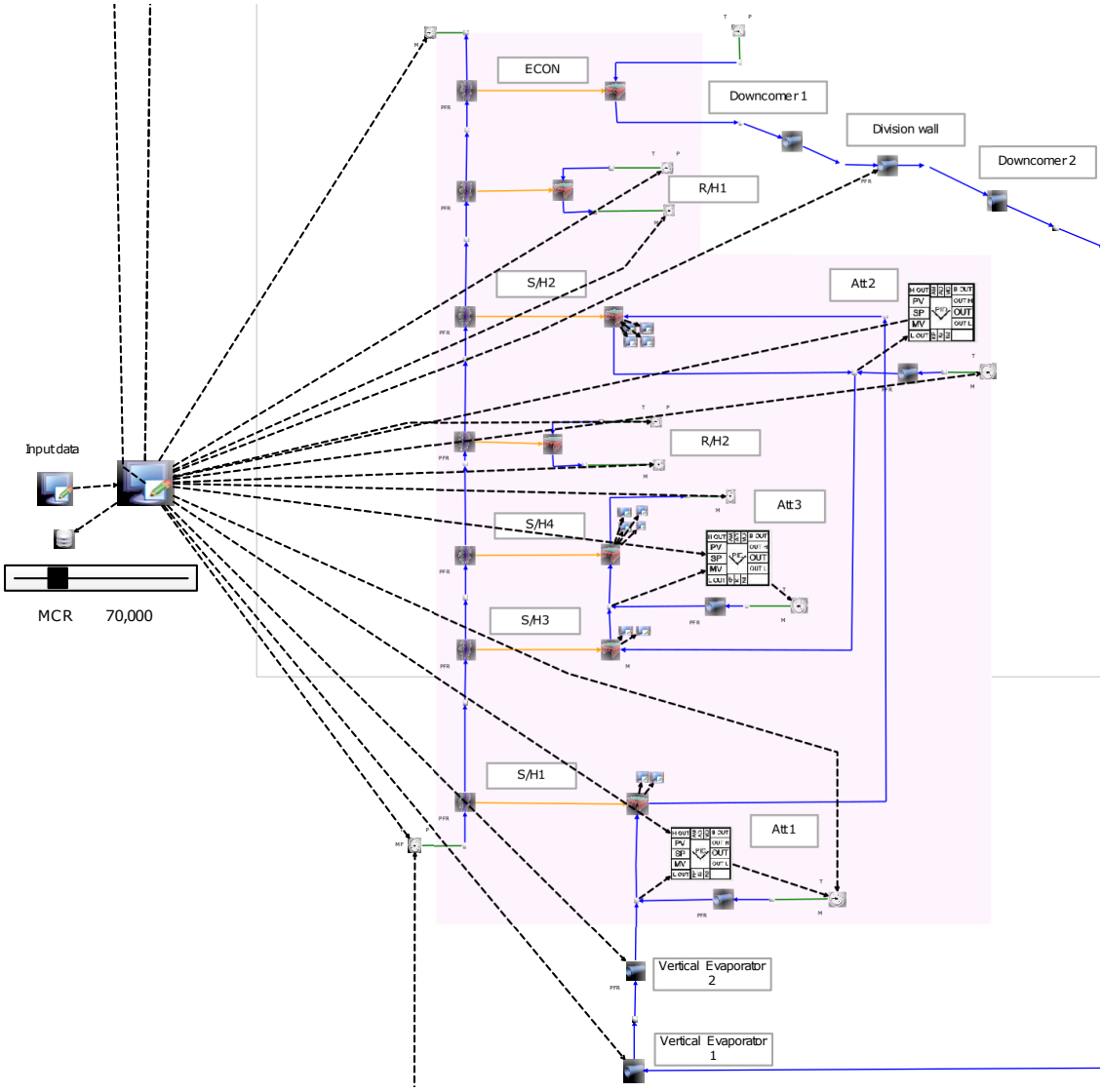


Figure 7.3.2: Flownex® model of the convection pass (shaded region), downcomers, division wall and vertical evaporators.

wall. The fluid then goes via the second downcomer to the helical tubes. They are modelled using simple pipe elements as can be seen in Fig. 7.3.2. Since the downcomers are positioned away from the furnace, they are assumed to experience no heat absorption. The fluid falls solely via gravity with a pressure drop that is calculated for Darcy-Weisbach pipes according to:

$$\Delta p = \frac{KLv_f^2\rho_f}{2d_i}, \tag{7.53}$$

where K is the forward loss coefficient to be calculated and L is the length of the tube. The unknown K of each boiler element is calibrated according to known pressure drops across the boiler at various loads.

7.3.3 Vertical Evaporator

The vertical evaporator can be split into two separate evaporators with different physical characteristics, namely vertical evaporator 1 and vertical evaporator 2. It is modelled using two pipe elements as shown in Fig. 7.3.2. Since heat transferred to the vertical tubes is too complicated to model (with less obvious radiative and convective contributions) it is taken as a fixed heat transfer through the tube walls that changes based on load (according to plant data). The pressure drop is modelled as in Section 7.3.2.

7.3.4 Helical Evaporator and Radiative Heat Transfer

The helical structure of the tube is modelled indirectly by calibrating the tube's loss coefficient according to plant data, which in turn dictates the pressure drop across the evaporator length (as explained in Section 7.3.2). The helical evaporator is segmented into ten sections, each represented by a pipe component so that heat transferred to each segment can be modelled very coarsely (see Fig. 7.3.3). The fluid's main source of heat is via radiation that is transferred to the fluid using three respective heat transfer elements - (1) a radiative heat transfer element that models the transfer of heat to the outer pipe wall, (2) a conductive heat transfer element that models heat transferred through the tube wall and a convective heat transfer element that models heat transferred from the inner tube wall to the fluid. Fig. 7.3.4 depicts a simple way (used in [123]) to calculate radiative heat transfer at the outer surface of the tubes using coarsely segmented surfaces from which viewing angles can be calculated according to furnace dimensions. Here the furnace waterwalls and surrounding flame are divided into 10 and 4 equally lumped segments respectively. From this, viewing angles between flame segments and furnace wall segments can be calculated, needed as input to the radiative heat transfer elements in Fig. 7.3.3. Each helical evaporator segment thus sees a sum of heat transferred from the 4 flame segments. The flue gas and waterwalls are seen as black bodies and both contribute to the total radiative properties of the furnace. The emissivity of clean furnace tube walls is typically 0.85 [125], but assuming an ash deposit layer on the wall surface reduces the wall emissivity to the range of $0.4 - 0.7$ for ash particles of 120µm in diameter[126]. Since the exact value of the wall emissivity is based on the ash deposit layer (which depends on several factors), this value is hard to obtain. The furnace wall emissivity is chosen as a pessimistic $\varepsilon_m = 0.45$. The flue gas emissivity depends on its gas composition and particulate characteristics and is chosen based on typical values for similar boilers, i.e., $\varepsilon_g = 0.5$ [125].

7.3.5 Furnace

The model for furnace combustion is shown in the shaded region in Fig. 7.3.3. It is modelled using an adiabatic flame component and 3 pipe elements that represent the coal, air supply and furnace conditions from which the average temperature of the flame is provided as output. This is then used, firstly, as input to the radiative heat transfer elements of the helical evaporator (rudimentary approximation), and secondly, to calculate the furnace flue gas exit temperature at the entrance of the convection pass.

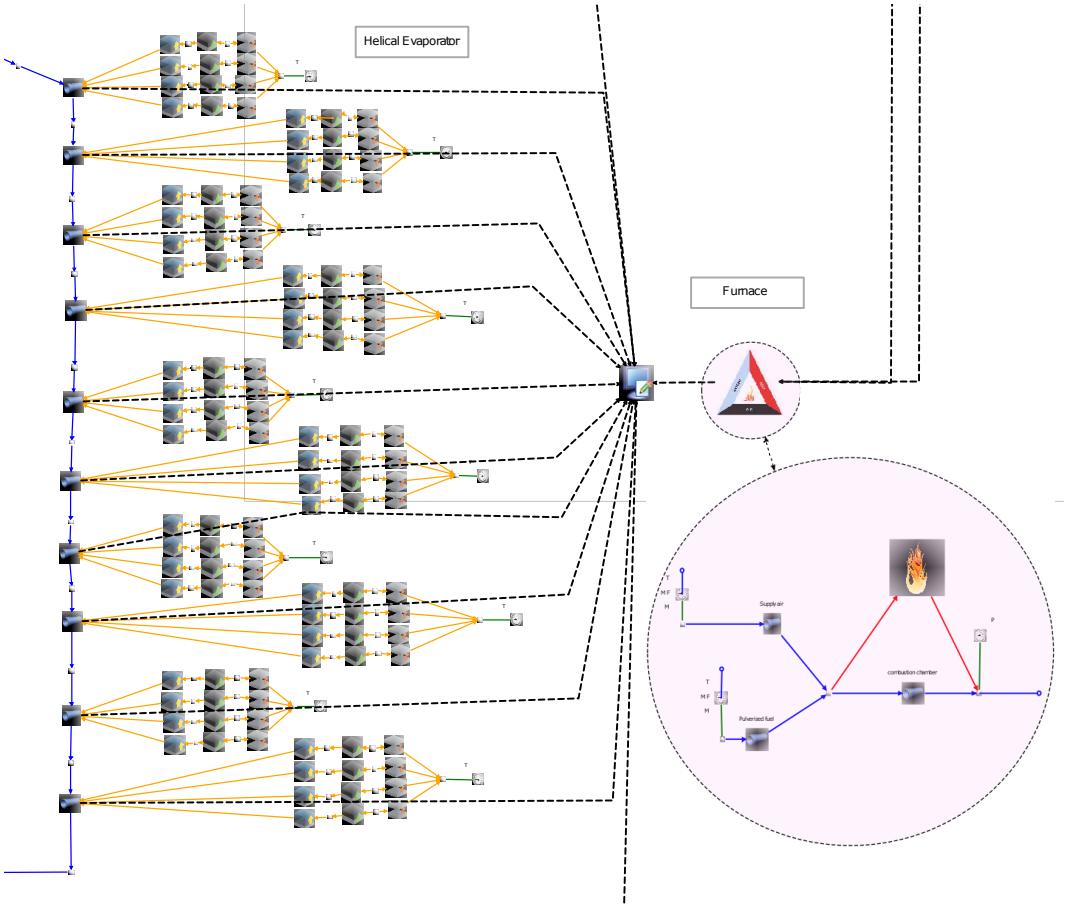


Figure 7.3.3: Flownex® model of the helical evaporator showing the furnace combustion sub-system (shaded region).

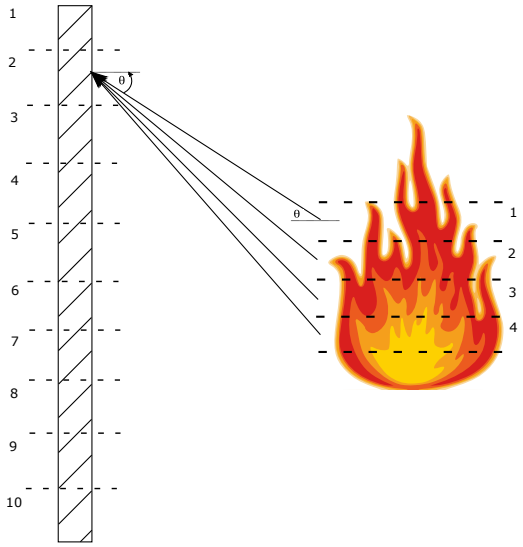


Figure 7.3.4: A diagram of the radiative heat transfer model.

7.4 Model Validation

This section compares the output of the MATLAB® and Flownex® models in Sections 7.2 and 7.3 and then compares the output of the MATLAB® model to actual plant data.

7.4.1 MATLAB® and Flownex® Models

In the first test the MATLAB® and Flownex® models are run to steady-state at 70 % MCR followed by a demand change as shown in Fig. 7.4.1. The ramped sequence in Fig. 7.4.1 is not intended to portray a realistic boiler input, but merely to reflect non-trivial changes in boiler input conditions over a short period (with respect to the boiler's time constant). The transient responses of the fluid temperature, fluid enthalpy and wall temperature at the exit of the superheaters are shown in Figs. 7.4.2a, 7.4.2b and 7.4.2c. In the second test both models are again run to 70 % steady-state followed by a change in only the fluid mass flow rate at the economiser inlet as shown in Fig. 7.4.3a. In this case the outputs at the respective superheaters are shown in Figs. 7.4.3b, 7.4.3c and 7.4.3d.

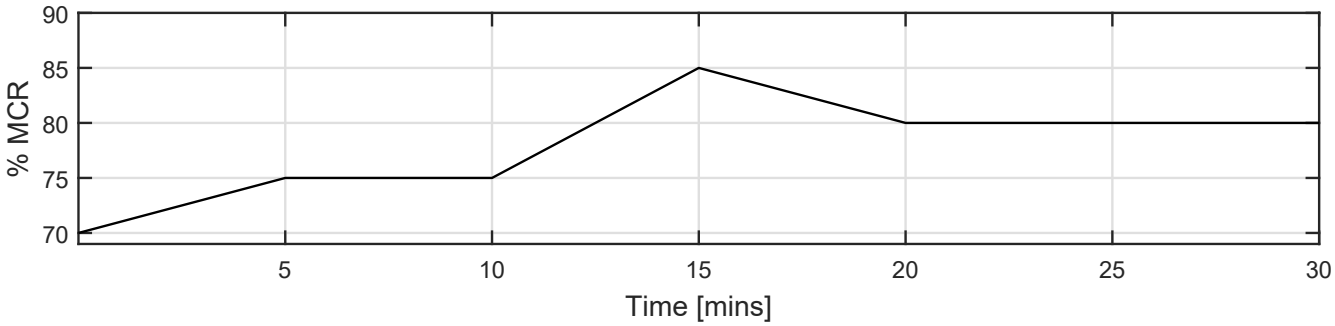
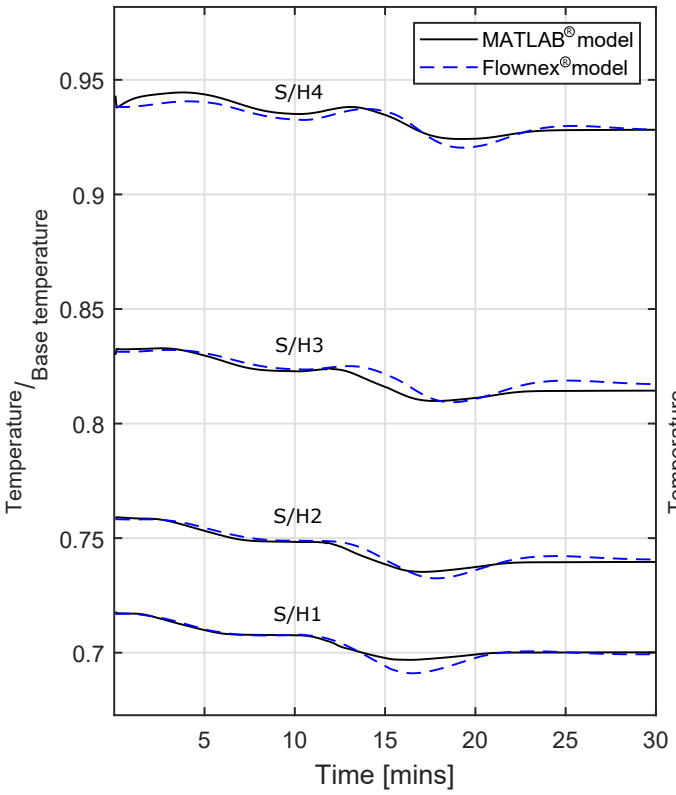


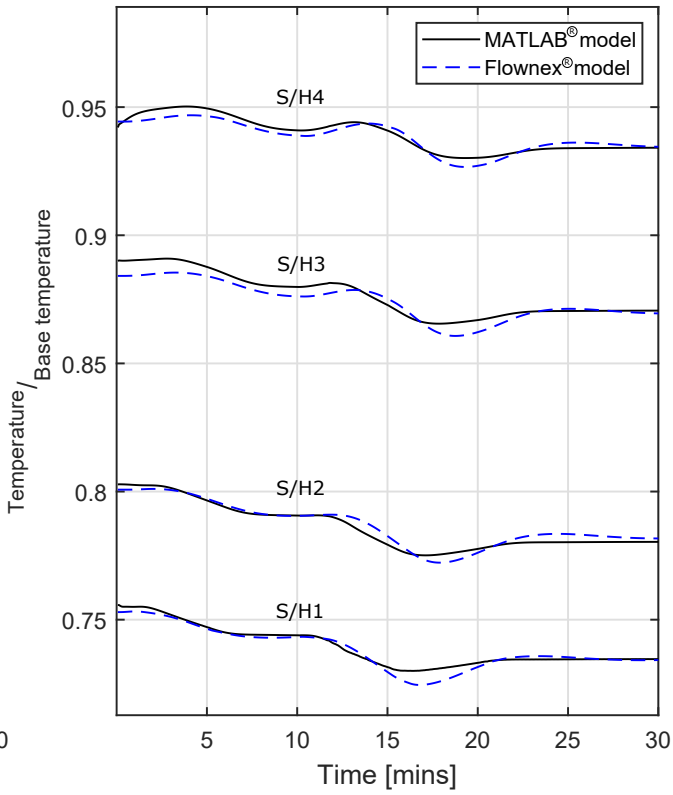
Figure 7.4.1: System demand.

Discussion

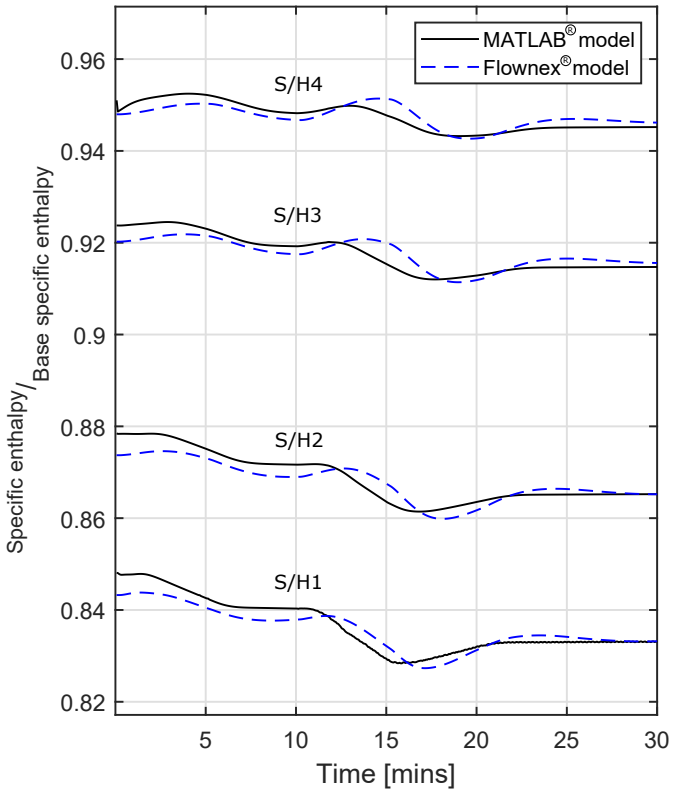
According to Figs. 7.4.2a, 7.4.2b and 7.4.2c it can be seen that the two models display similar transient and steady-state behaviour. The slight differences in transient behaviour can be attributed to the incorporation of more exact pressure dynamics in the Flownex® model. Furthermore, explicit radiative heat transfer is modelled in the Flownex® model, whereas the MATLAB® model simply uses a fixed heat rate into the helical tube walls based on the boiler load. Another reason for the difference, as mentioned before, is that the Flownex® model is solved according to a set of PDEs, unlike the set of ODEs in the MATLAB® model. Despite the slight variation between the two models, it is evident that the MATLAB® model captures the general behaviour of the boiler within reasonable error, i.e., of the same order of magnitudes as that of typical boiler process and parameter uncertainty. It furthermore executes much faster (not shown explicitly) since it uses much simpler modelling philosophies. The final test in Figs. 7.4.3b, 7.4.3c and 7.4.3d sees a purely sonic disturbance originating at the economiser inlet. From these plots it can be seen that the two models exhibit similar dynamic behaviour that coincides when considering standard boiler uncertainty.



(a) Normalised fluid temperature.

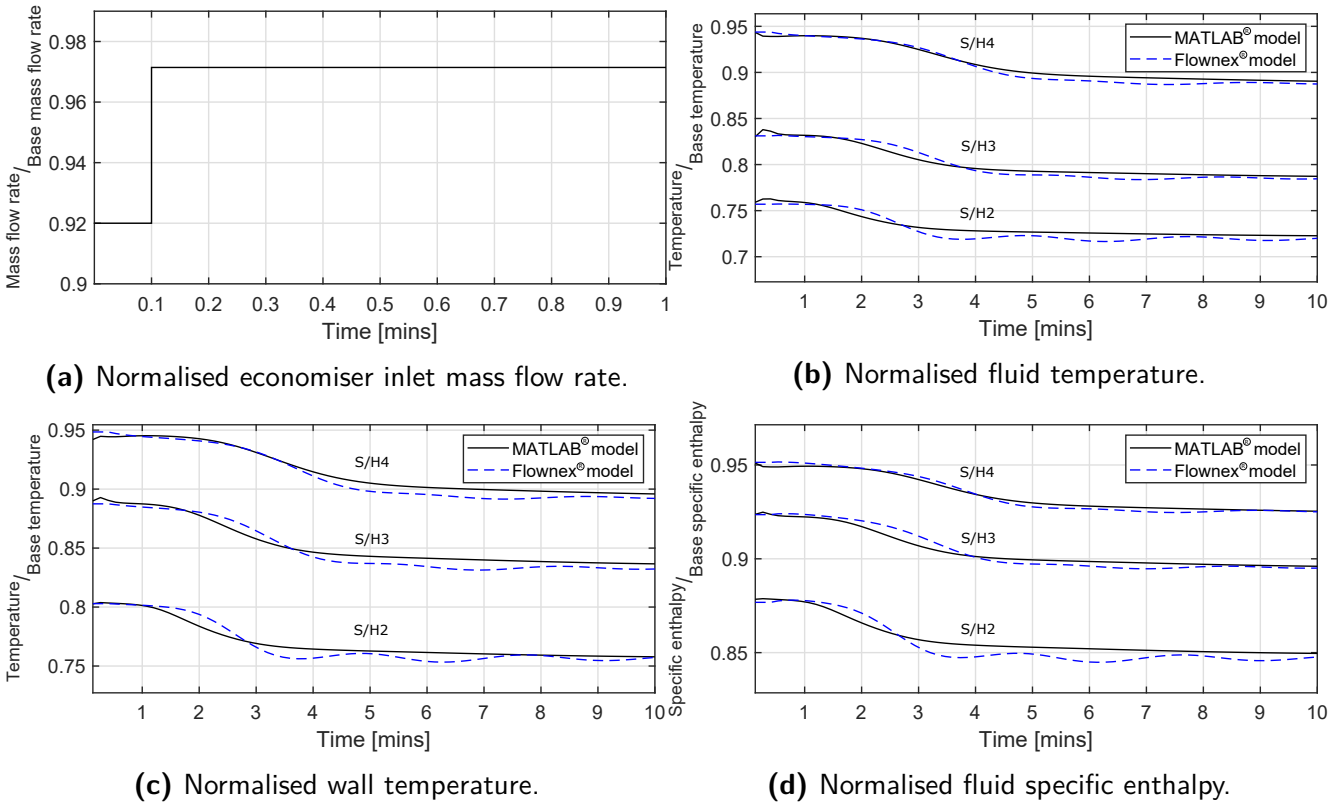


(b) Normalised wall temperature.



(c) Normalised fluid specific enthalpy.

Figure 7.4.2: Normalised wall temperatures and fluid specific enthalpies at the exit of the respective superheaters.



(a) Normalised economiser inlet mass flow rate.

(b) Normalised fluid temperature.

(c) Normalised wall temperature.

(d) Normalised fluid specific enthalpy.

Figure 7.4.3: System demand, normalised wall temperatures and fluid specific enthalpies at the exit of the respective superheaters.

7.4.2 MATLAB® Model and Plant Data

In Figs. 7.4.5a and 7.4.5b the output of the MATLAB® model is compared to data from the actual plant over a period that includes a relatively constant demand, followed by a significant demand change as shown in Fig. 7.4.4. The input boundary conditions, available from the EtaPro® database, include the feedwater mass flow rate, pressure and temperature, and are used as inputs to the MATLAB® model, while the remainder of the input boundary conditions are calculated based on various load conditions as done for the models. Two nodes in the boiler are chosen for validation, namely the fluid temperature at the exit of the economiser, Fig. 7.4.5a, representing the highest flue gas location in the boiler, as well as the fluid temperature at the exit of ATT 1, Fig. 7.4.5b, depicting the entry to, and lowest point in the convection pass. These two nodes occur at distinct locations in the boiler and are chosen different to the discrete nodes in Fig. 7.4.2 and 7.4.3.

Discussion

Figs. 7.4.5a and 7.4.5b show slight differences in the dynamic behaviour of the MATLAB® model when compared to actual plant data. There also exists a slight lead in fluid temperature at the exit of ATT 1 in Fig. 7.4.5b which is noticeable at significant load changes. These issues are most likely a result of an inaccurate representation of the thermal inertia associated with the MATLAB® model due to uncertain parameters, changes in oper-

ating conditions and the simplified modelling process. Despite the inaccuracies, the MATLAB® model output falls within standard uncertainty of the actual plant output. The general dynamic behaviour of the MATLAB® model is thus considered to be acceptable.

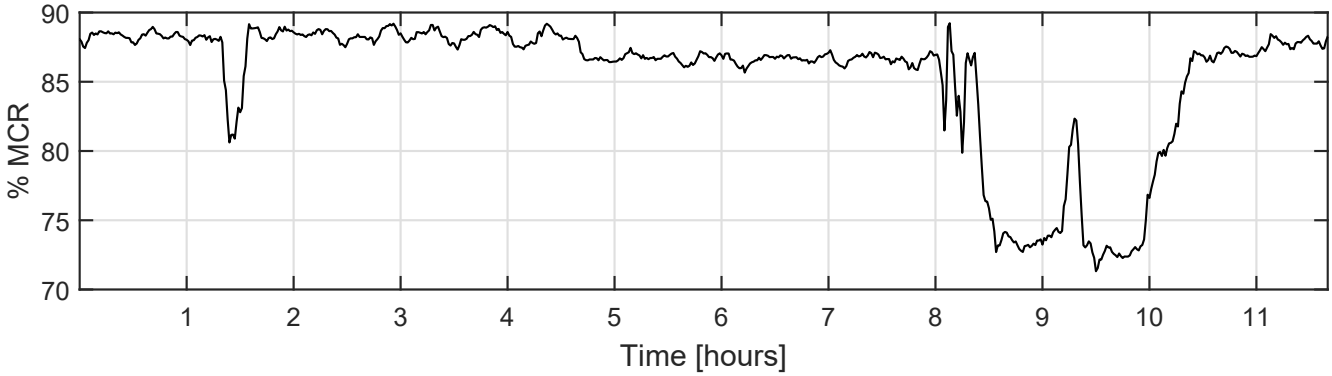
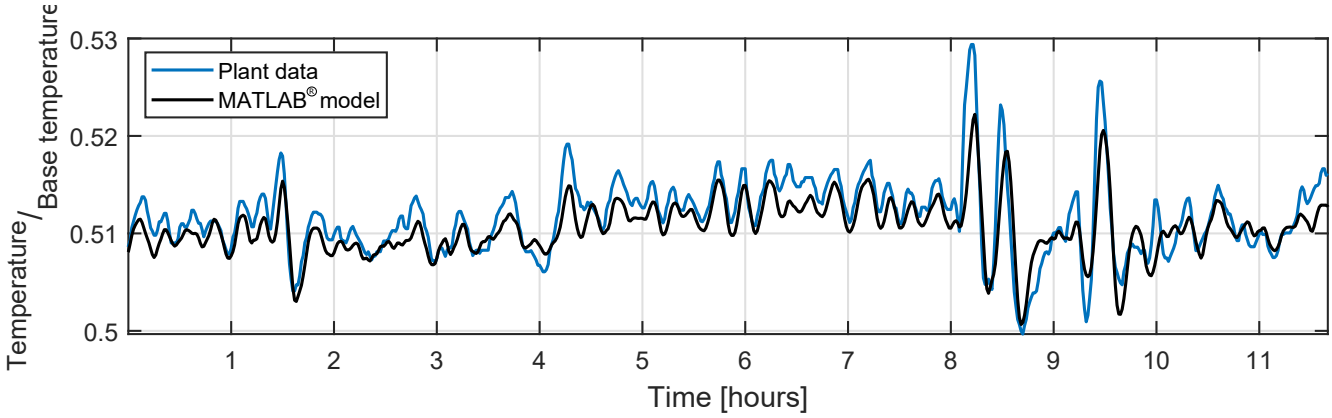
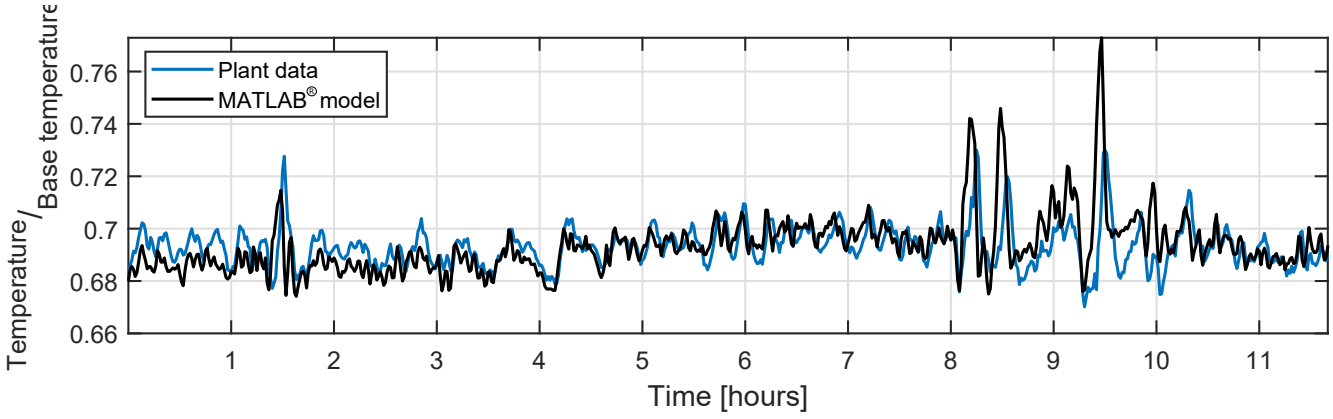


Figure 7.4.4: System demand.



(a) Normalised fluid temperature at the exit of the economiser.



(b) Normalised fluid temperature at the exit of ATT1.

Figure 7.4.5: Normalised fluid temperatures.

7.5 Conclusion

In this chapter, two models of the boiler are developed. The first is a mathematical model in MATLAB® and the second a PDE-based Flownex® model. It was shown that the MATLAB® model is in agreement with both the Flownex® model and actual plant data when considering standard boiler uncertainty. From this it is concluded that the MATLAB® model can be accepted for use in the filter, which should in retrospect correct any tolerable discrepancies in the model. It should be noted that the intention of this chapter is not to develop a painfully accurate filter model of the boiler, but rather to develop one that can adequately represent the dynamics of the respective boiler elements in order to demonstrate the feasibility of the filtering process when applied to the boiler. Furthermore, the MATLAB® model serves as a template for more refined models. More accurate mathematical models are always beneficial in the filter and its development is left as a future recommendation.

"The reduction of theory to practice is generally a long and difficult process ... Yet reports of applications often deal only with the successful final results and are thus, in a sense, very misleading, especially to the uninitiated."

Andrew Jazwinski in [7]

8

State Estimation of a Once-Through Power Boiler

State estimation in practical non-linear applications are approximate (as discussed in Chapter 4), since there is no-closed form solution of the associated non-linear equations, and suboptimal, because it is not possible to model all sources of error in the system. In such cases, a systematic design of the filter can help shift the filter closer to optimality and facilitate desired performance. The contribution of this chapter is to properly set up and design a non-linear CD-KF according to the algorithms in Part 1 in order to estimate over a hundred key state variables associated with the main pressure components of a power boiler, i.e., tube wall temperatures and fluid specific enthalpies. This can be used as a first step into condition-based monitoring of the boiler such as heat uptake at the furnace walls, fault detection such as fouling, and optimised feedback control that, for example, makes use of specific enthalpy states at various locations across the boiler. The filter is designed in Sections 8.1, 8.2, 8.3, 8.4, 8.6 and 8.5 and then applied to the common problem of fouling in Section 8.7 in order to, firstly, test the filter's performance and, secondly, to demonstrate the filter's application to fault detection. This is followed in Section 8.8 by its application to actual plant data to further illustrate its utility.

8.1 State Identification and Dynamic State Equations

A fluid's specific enthalpy is a characteristic thermodynamic property that cannot be measured directly. It signifies the fluid's heat content (at a given pressure and constant volume), useful in identifying behavioural inconsistencies and is hence a preferred state variable. Tube wall temperature is another key variable that can be used to monitor the physical health of the plant. If a sufficient number of distributed segments are used in the state

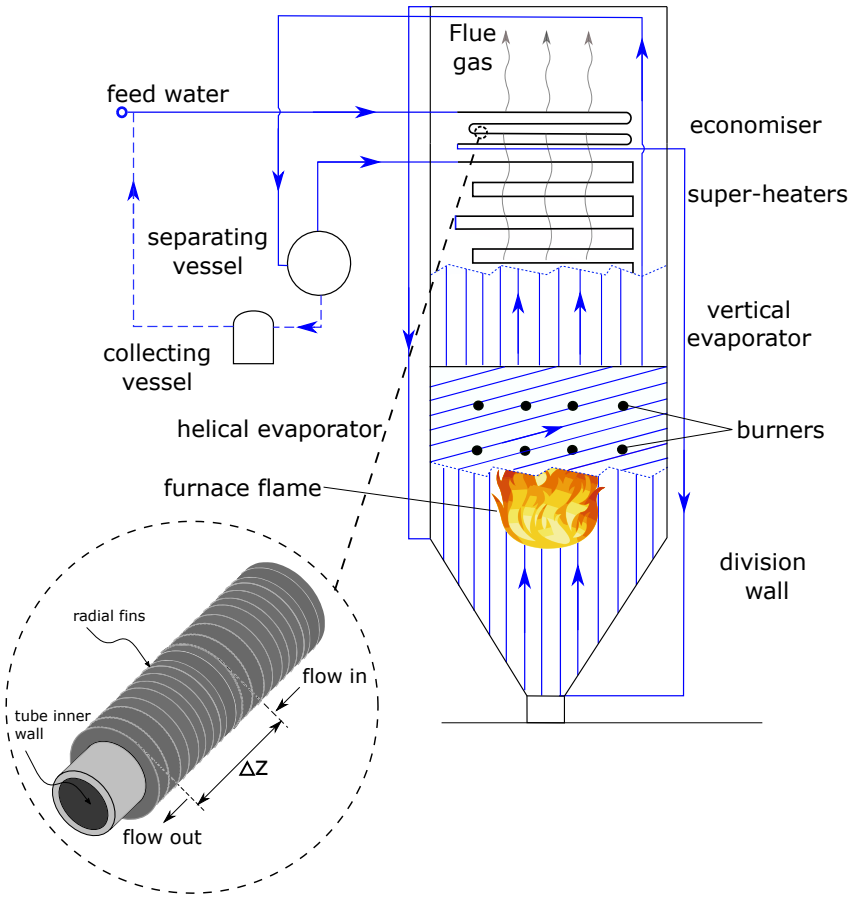


Figure 8.1.1: A diagrammatic representation of the boiler and one of its discrete segments.

estimation model, then wall temperature can indicate where along the boiler length is film boiling occurring or where along the boiler length is the tube subjected to high levels of thermal stress (in the event that these issues arise). The collection of parallel tubes along the boiler length is divided into n/2 spacial segments, shown in Fig. 8.1.1, corresponding to the state vector

$$x = [T_{m1}, \ldots, T_{m\,n/2}, h_1, \ldots, h_{n/2}]^T, \tag{8.1}$$

where n=136 is the total number of state variables. Other important variables such as drum vapour quality or moisture content in pulverised coal can also be included in the state vector. For the purpose of this application, it suffices to consider only wall temperature and fluid specific enthalpy.

The MATLAB® model in Chapter 7 was shown to be an adequate dynamic model for use in the filter, with state differential equations per segment based on those from Section 7.2.7:

$$\frac{dT_m}{dt} = \frac{\dot{q} - \dot{q}_f}{m_m c_{p,m}}, \tag{8.2}$$

$$\frac{dh}{dt} = \frac{\dot{m}_{in}(h_{in} - h_{out}) + \dot{q}_f}{\rho_f V}. \tag{8.3}$$

These equations are derived from a set of partial differential equations (PDEs) based on energy, mass and momentum balances that have been reduced to a 1-dimensional lumped initial value problem that can be solved using linearly implicit RK methods. This dynamic model satisfies the assumption of Markovian state evolution needed for Kalman filtering.

8.2 Boiler Inputs and Measurement Model

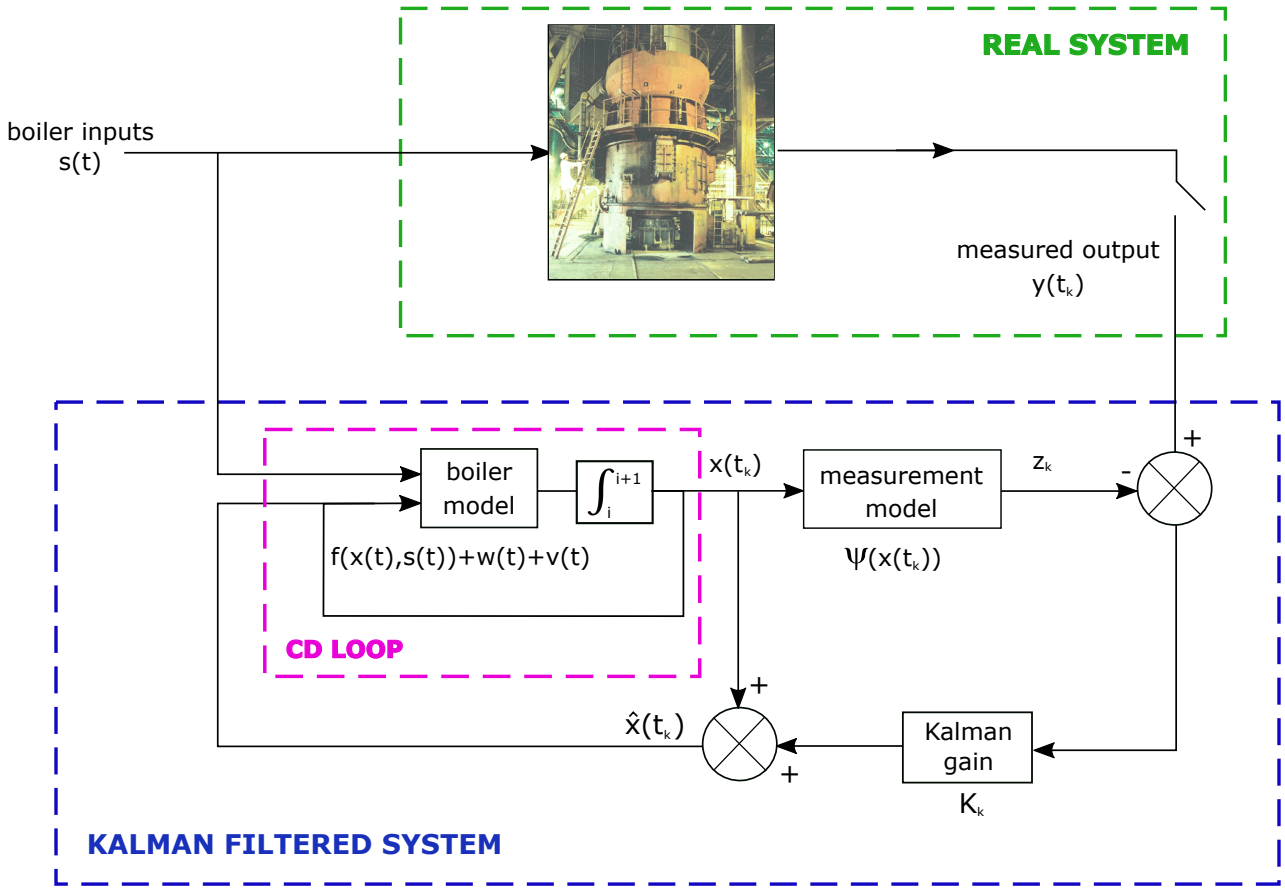


Figure 8.2.1: Simplified diagram of the Kalman filter.

Inputs to the boiler, $s(t)$, available from the EtaPro® database, and sampled at 10 s intervals over a 12 hour period, are used as input boundary conditions to the filter model as depicted in Fig. 8.2.1 and include:

- Economiser feedwater temperature, pressure and mass flow rate
- Mass flow rate of ATT 1 spray water into the 2 respective subheaders (legs).
- Mass flow rate of ATT 2 spray water into the 4 respective legs.
- Mass flow rate of ATT 3 spray water into the 4 respective legs.

Relevant potential output measurements, $y(t_k)$, available from the EtaPro® database, according to the boiler P&IDs, and sampled at 10 s intervals over the same 12 hour period as the inputs include:

1. Economiser outlet fluid temperature.
2. Fluid temperature of the 2 respective legs at the inlet and outlet of ATT 1.
3. Separating vessel outlet fluid steam temperature. This is approximately equal to the outlet fluid temperature of attemperator 1 which makes sense for normal boiler operation.
4. Fluid temperature of the 4 respective legs at the inlet and outlet of ATT 2.
5. Fluid temperature of the 4 respective legs at the inlet and outlet of ATT 3.
6. Main steam temperature of the 4 legs at the outlet of SH/4.
7. Metal temperatures of the 4 legs at the outlet of SH/2.
8. Metal temperatures of the 4 legs at the outlet of SH/3.
9. Metal temperatures of the 4 legs at the outlet of SH/4.

Local measurements only hold information associated with state variables upstream from it [127]. It is therefore preferable to include more measurements in the measurement configuration that exist further downstream since they penetrate more observable states. There are, however, limited measurements available overall. The steam temperature at the inlet of the attemperators, points numbered (2), (4) and (5) in the measurement output list, and the main steam temperature at the exit of S/H4, point numbered (6), are chosen as measurement variables in the filter. The remainder of the output measurements are withheld from the filter and reserved for performance testing purposes. In practice, all available output measurements should be fed to the filter to help reduce the state error covariance estimates and improve accuracy. The measurement function at time k can now be written, based on the IAPWS_IF97 formulation, as

$$\begin{aligned}\Psi(x(t_k)) = [&\text{IAPWS}(`T_{f,40,k}\,', p_{40,k}, h_{40,k}), \text{IAPWS}(`T_{f,60,k}\,', p_{60,k}, h_{60,k}), \text{IAPWS}(`T_{f,64,k}\,', p_{64,k}, h_{64,k}),\\ &\text{IAPWS}(`T_{f,68,k}\,', p_{68,k}, h_{68,k})]^T,\end{aligned} \tag{8.4}$$

where IAPWS$(`T_f\,', p, h)$ represents a pseudo IAPWS_IF97 function that takes fluid pressure and specific enthalpy, p and h, as inputs and outputs the fluid temperature T_f.

8.2.1 Stochastic Observability

It follows to verify whether the measurement configuration in (8.4) would allow for complete system observability. Define the stochastic observability matrix [7],[27] for all $t > t_o$ as :

$$\mathscr{O}(t, t_o) = \int_{t_o}^{t} \Pi(\tau, t) H^T(\tau) R^{-1}(\tau) H(\tau) \Pi^T(\tau, t) d\tau, \tag{8.5}$$

where $\Pi(t+\delta, t)$ is the state transition matrix such that $x(t+\delta) = \Pi(t+\delta, t)x(t)$, R is the measurement covariance and $H(t) = \frac{\partial \Psi(x)}{\partial x}|_{x=\bar{x}(t)}$ is the measurement Jacobian. A continuous-time (or CD) stochastic system is completely observable if (8.5) is positive-definite. Since the measurement Jacobian is not available, it can be replaced by its HOLM equivalent such that

$$\mathscr{O}(t, t_o) = \int_{t_o}^{t} \Pi(\tau, t) \mathcal{H}^T(\tau) R^{-1}(\tau) \mathcal{H}(\tau) \Pi^T(\tau, t) d\tau, \tag{8.6}$$

The integral in (8.5) and (8.6) is hard to solve in practice over a simulation trajectory. Consider segmenting the trajectory over integration time steps, δ, such that the localised observability matrix at each integration time instant, i, is

$$\mathscr{O}(t_{i+1}, t_i) = \int_{t_i}^{t_{i+1}} \Pi(\tau, t_i) \mathcal{H}^T(\tau) R^{-1}(\tau) \mathcal{H}(\tau) \Pi^T(\tau, t_i) d\tau, \quad t_o < t_i < t_{i+1} < t, \tag{8.7}$$

where $\Pi(t_{i+1}, t_i)$ is the state transition matrix formed according to the TH implicit numerical method with solution $x(t_{i+1}) = x(t_i) + \left(I - \mathcal{J}(x(t_i))\frac{\delta}{2}\right)^{-1} f(x(t_i))\delta$. The state transition matrix associated with the TH scheme after substituting $f(x(t_i) \approx \mathcal{J}(x(t_i))x(t_i)$ is

$$\Pi(t_{i+1}, t_i) = I + \left(I - \mathcal{J}(x(t_i))\frac{\delta}{2}\right)^{-1} \mathcal{J}(x(t_i))\delta. \tag{8.8}$$

The integral in (8.7) can now be solved using the explicit Euler scheme assuming that the inside of the integral in (8.7) is sufficiently smooth and noting that $\Pi(t_i, t_i) = I$. In order for the matrix in (8.5) to be positive-definite, all matrices in (8.7) for i = 1,...L-1, must be positive-definite, where L is the number of integration time instances in the test trajectory. The inside of the integral in (8.7) is evaluated at each instance, separated by a $\delta = 0.2$ s interval [1] over a two minute test trajectory. The measurement HOLM is available from Algorithm 1B at each measurement instance for $\Delta = 10$ s. In between measurements, the measurement HOLM is taken as the most recently calculated HOLM. At each integration instance, the matrices are found to be positive-definite via a Cholesky decomposition. The measurement configuration therefore allows complete observability.

[1] relatively small in comparison to the boiler's time constant

8.3 Error Budget

As mentioned before, the more comprehensive and accurate the account of errors in the filter are, the closer to optimal the filter will be and therefore the better its performance. This section provides a detailed account of all sources of error and uncertainty associated with state processes and measurements.

8.3.1 Measurement Covariance

The quality of measurement instruments used varies based on where it is located within the boiler and what is being measured. It is assumed in this case, however, that all temperature sensors are the same. Industrial thermocouples typically have a tolerance of around ± 3 °C at 500°C [128]. Assuming a uniform probability distribution and independence between sensors gives a variance of $R_k = 1.73^2 I_m$ [°C]^2. Here the systematic component of the measurement uncertainty is represented as statistical error and there are no off-diagonal terms in R_k and hence no state vector augmentation. This is based on the assumption that a bias error can be regarded (in a naive sense) as a larger fluctuation. To motivate this, consider the case of two Gaussian variables $(\sim \mathcal{N}(\bar{x}, \sigma^2))$ in Fig. 8.3.1. In this case the variable $(\sim \mathcal{N}(1, 2^2)$ represents the true biased measurement variable and $(\sim \mathcal{N}(0, 3^2)$ a zero mean variable with a systemic error of 1 simply added to the variance. What this shows is that the variable with more variance results in larger fluctuations and covers most of the area spanned by the true biased error but includes an unnecessary space (in the negative y-quadrant). This approach thus provides more vague information to the filter about the possible locations of the measurement when compared to properly modelling the bias.

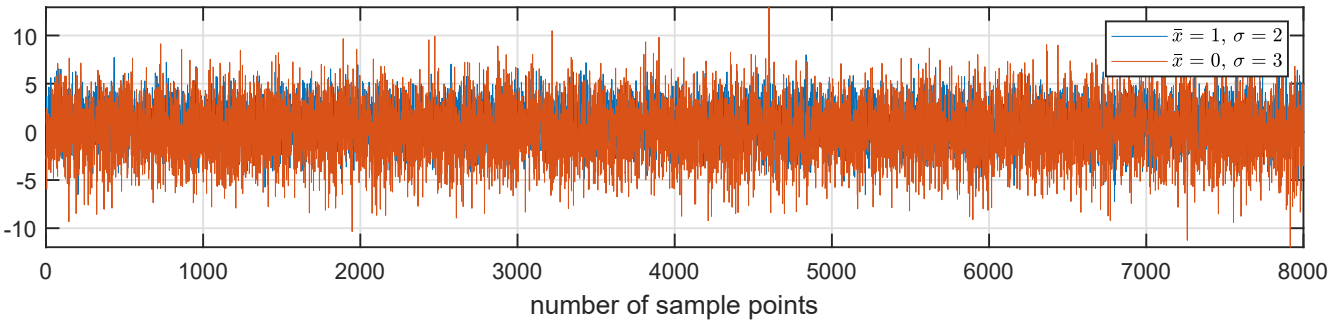


Figure 8.3.1: A simulation of two Gaussian measurement variables. The orange variable has zero mean and a larger variance to represent the simpler measurement model. The variable in blue represents a biased measurement model.

8.3.2 Process Covariance

According to the SDE in (5.4), there are two error models associated with the state vector - the purely time-varying model, with scaling $q(t)$, and the state-dependent noise model with scaling $b(x(t), t)$ (discussed further

in Section 8.5). The continuous-time process covariance, $Q_c(t) = q(t)Q(t)q(t)^T$, comprises model uncertainty, spatial discretisation errors and disturbances.

Model uncertainty is dominated by unpredictable furnace behaviour that arises as a result of inconsistent mill functioning and variations in the quality of raw materials used, namely, the particle size distribution of pulverised coal, the calorific value of coal, the fraction of volatile matter in coal, mill condition and pulverised fuel uptake and transport. This results in incorrect model parameters and input boundary conditions associated with the filter. It is further exacerbated by the simplified models of the furnace and a lack of measurements in that region. Deviations in parameters due to potential faults in the boiler also contribute to model uncertainty. For the demonstrated case of fouling, the heat transfer coefficients associated with the affected convective element would be slightly lower than a fault-free one and would be reflected in its Colburn factor, j_H. The main sources of model uncertainty are summarised in Table 8.3.1.

Table 8.3.1: Model uncertainty

source	**uncertainty**
Ultimate analysis of H_2O in coal	$\pm$ 0.02
Ultimate analysis of H_2 in coal	$\pm$ 0.02
Furnace supply temperature	$\pm$ 10 °C
Pulverised coal mass flow rate	$\pm$ 3 kg/s
Emissivity of flue gas	$\pm$ 0.05
S/H4 Colburn factor	$\pm$ 0.006
S/H3 Colburn factor	$\pm$ 0.006
S/H2 Colburn factor	$\pm$ 0.003
S/H1 Colburn factor	$\pm$ 0.007

Sudden plant parameter changes due to external influences, or disturbances, effectively introduce more uncertainty into the plant. This can be seen in quantities such as the flue gas and feedwater fluid mass flow rate. Their variations are summarised in Table 8.3.2.

Table 8.3.2: Plant disturbances

source	**variation**
Mass flow rate of flue gas	$\pm$ 20 kg/s
Mass flow rate of inlet fluid	$\pm$ 5 kg/s
Temperature of inlet fluid	$\pm$ 3 °C

The error brought about by ignoring flue gas temperature variation in the direction perpendicular to the flow of the flue gas is taken to be much less than other sources of uncertainty and is therefore not included in the error budget. Spatial and temporal discretisation further contributes to state error. With a small enough integration

step size, however, neglected higher-order terms in the TH solution of the state differential equations leads to insignificant integration errors in comparison to other sources of error in the system. When integration step sizes greater than $\delta = 0.4$ s are used (observed in simulation), the integration error can lead to a greater than 0.05°C/s change that must be accounted for and is discussed further in Section 8.5.

Since the uncertainty in steam enthalpy and wall temperature depend on multiple variables and parameters described above, the combined uncertainty of the system is not simply additive and follow standard error propagation techniques. Instead of lengthy error propagation calculations, however, the combined uncertainty is obtained by selecting permutations of worst case parameters in Table 8.3.1 and Table 8.3.2 and observing its effect in simulation. The dynamic model is solved using an explicit Euler method with a relatively small integration time step of $\delta = 0.05$ s. The effects of spatial discretisation error are included in this analysis by doubling the number of discrete segments of each element. This is then compared to an error-free model and effectively results in a maximum deviation in wall temperature and fluid enthalpy, read from a plot such as that in Fig. 8.3.2 and summarised in Table 8.3.3. Assuming a uniform distribution, the resultant covariance parameters for each state, such that

$$Q_c = \text{diag}(Q_{T_{m1}}, \ldots, Q_{T_{m68}}, Q_{h_1}, \ldots, Q_{h_{68}}),$$

is given in Table 8.3.4. For simplicity, the slight variation along the respective elements themselves is ignored. These covariances can also be obtained via Monte Carlo methods.

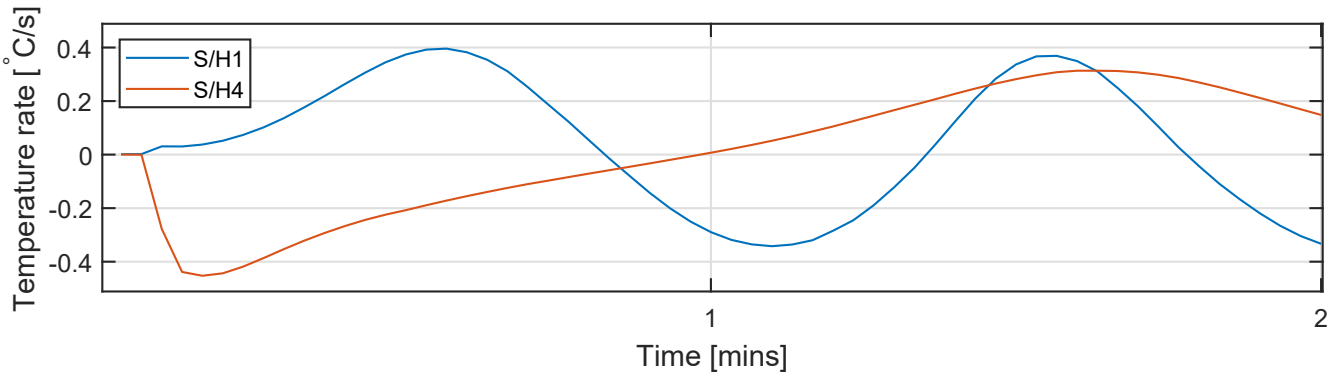


Figure 8.3.2: Plots of the wall temperature deviation rate for S/H1 and S/H4.

8.3.3 Initial State Error Covariance

The initial state covariance is chosen as a diagonal matrix with equal non-zero elements for the respective temperature and enthalpy state entries. The covariance associated with the predicted state mean should converge to the correct covariance value given a reasonable guess of the uncertainty in the initial state mean estimate, cho-

	Maximum wall temperature change [°C/s]	Maximum enthalpy change [kg/kJ/s]
Economiser	0.25	2.63
Division wall	0.25	2.94
Helical evaporator	0.3	3.39
Vertical evaporator 2	0.33	4.64
Vertical evaporator 1	0.35	4.95
S/H1	0.4	4.16
S/H2	0.5	3.86
S/H3	0.4	2.72
S/H4	0.45	2.79

Table 8.3.3: Maximum possible variation in wall temperature and fluid enthalpy of the respective boiler elements.

	Temperature cov. [°C²/s]	Enthalpy cov. [kg²/kJ²/s]
Economiser	$Q_{T_{m1}} = ... = Q_{T_{m10}} = 0.17^2$	$Q_{h_1} = ... = Q_{h_{10}} = 1.52^2$
Division wall	$Q_{T_{m11}} = ... = Q_{T_{m20}} = 0.17^2$	$Q_{h_{11}} = ... = Q_{h_{20}} = 1.70^2$
Helical evaporator	$Q_{T_{m21}} = ... = Q_{T_{m30}} = 0.18^2$	$Q_{h_{21}} = ... = Q_{h_{30}} = 1.96^2$
Vertical evaporator 2	$Q_{T_{m31}} = ... = Q_{T_{m35}} = 0.19^2$	$Q_{h_{31}} = ... = Q_{h_{35}} = 2.68^2$
Vertical evaporator 1	$Q_{T_{m36}} = ... = Q_{T_{m40}} = 0.20^2$	$Q_{h_{36}} = ... = Q_{h_{40}} = 2.86^2$
S/H1	$Q_{T_{m41}} = ... = Q_{T_{m56}} = 0.23^2$	$Q_{h_{41}} = ... = Q_{h_{56}} = 2.40^2$
S/H2	$Q_{T_{m57}} = ... = Q_{T_{m60}} = 0.29^2$	$Q_{h_{57}} = ... = Q_{h_{60}} = 2.23^2$
S/H3	$Q_{T_{m61}} = ... = Q_{T_{m64}} = 0.23^2$	$Q_{h_{61}} = ... = Q_{h_{64}} = 1.57^2$
S/H4	$Q_{T_{m65}} = ... = Q_{T_{m68}} = 0.26^2$	$Q_{h_{65}} = ... = Q_{h_{68}} = 1.61^2$

Table 8.3.4: Covariances of the respective boiler elements.

sen as $P_0 = \text{diag}(P_{0,T_{m1}}, ..., P_{0,T_{m68}}, P_{0,h_1} ..., P_{0,h_{68}})$ with $P_{0,T_{m1}} = ... = P_{0,T_{m68}} = 0.5^2$ [°C]² and $P_{0,h_1} = ... = P_{0,h_{68}} = 5^2$ [kg/kJ]².

8.4 Computational Factors

In the interest of efficiency, the following section discusses potential ways to reduce the filter's computational time. It must be noted that although most of the following approaches are feasible, they are not employed in this thesis. For completeness sake, the reasons why are given below.

1. Decouple states - This is synonymous with modular modelling [129] which partitions the boiler into several modules and then calculates Jacobians and filter solutions separately for each module. This approach will not be used because (1) Algorithms 2A and 1B calculate the entire Jacobian matrix equivalent almost effortlessly and (2), the arithmetic associated with the n × n matrices is solved more efficiently using the default MATLAB® sparse solver.

2. Pre-filter - In this approach periodic measurements are averaged over an interval to give the filter enough time to complete its calculations without leaving some measurements unused. Since the greatest com-

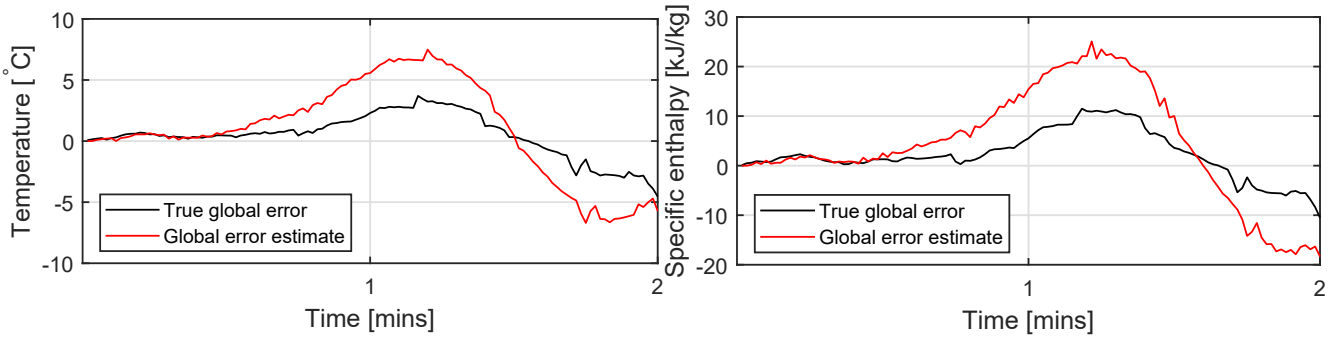
putational workload occurs during state propagation, delaying a measurement update would make the covariance grow without contributing significantly to improved computational speed.

3. Pre-compute gain - An *a priori* gain is calculated, for example, a steady-state gain, and kept as a constant throughout the filtering process. This simplification results in poorer performance for a boiler [108] and will therefore not be used in the filter design.

4. Process measurements one at a time - Instead of an m-dimensional measurement vector, measurements are processed 1 at a time, separated by an instant, in order to reduce computational complexity. This would make a marginal difference in the execution time of Algorithm 2A and 1B since the state propagation stage of the filter dominates the computational load.

8.5 The State-Dependent Noise Term

The process covariances discussed in Section 8.3.2 are modelled as constant terms as in the case of the standard CD-KF. Algorithm 2A, to be used in the boiler application, includes a state-dependent noise term that accounts for the numerical error associated with temporal discretisation during simulation as explained in Chapter 5. Its formulation begins by calculating the local and global error estimates of the boiler according to (3.41) and (3.34). This is then followed by calculation of the less complex local error model in (5.37). For the purpose of this application, it suffices to obtain the state-dependent noise model for a single integration time step, thus avoiding the explicit time dependence of the form in (5.37). This makes sense because the filter does not employ variational integration methods. An integration step size of 2 s is chosen for use in the filter. At this step size, the global error estimate is calculated using the local error formula in (3.41). Using this, the global error is then calculated according to (3.34). The estimated global error is then verified according to the true global error. The true global error is calculated using an Euler approximation with a very small integration time step. The local error estimate according to (3.34) is therefore concluded to be acceptable. The less complex local error estimate is then calculated and is of the form $l_{i+1} = \mathcal{D}f(x_i)$ (similar to that of (5.37) but without the time dependence). Here $\mathcal{D}$ is obtained via regression as explained in Section 5.3.2 but is assumed diagonal in order to simplify the regression analysis. The global error estimate can then be calculated using the less complex local error estimate and compared to the true global error. This comparison is shown in Fig. 8.5.1 for discrete segment number n=62 at 2 s integration intervals for a short period. With respect to these temperature and specific enthalpy error plots it can be seen that the global error estimates form an unexaggerated overbound on the true global error. This implies that the local error estimate is a modest representation of the true local error. The local error estimate model is therefore accepted for use in the boiler application.



(a) Global error in T_{m62}.

(b) Global error in T_{m62}.

Figure 8.5.1: Global error of the wall temperature and fluid enthalpy state for discrete segment number n = 62.

8.6 Algorithm Design in MATLAB®

The general flow of the non-linear CD-KF based on Algorithms 2A and 1B is shown in Fig. 8.6.1. The function `UTF` (point numbered (2) in Algorithm 2A) distributes the 2n+1 function evaluations amidst multiple local cores via the MATLAB® parallel *for* loop, the results of which are then collected into a single matrix to form the drift UT. As the number of cores on the local computer increases, the faster the UTs are calculated. Using the drift UT, the drift HOLM is extracted according to point numbered (3) in Algorithm 2A. The local error and local error HOLM associated with the implicit solver is calculated according to points numbered (5),(6) and (7) in Algorithm 2A. The state mean and error covariance estimates at the next time step are then calculated according to points numbered (8) and (9). The input conditions of the boiler in the time update loop are held at the current conditions until the next set of inputs from the plant becomes available. Once the time update loop has completed, the function `MeasUpdate` extracts the measurement HOLM, which is used to perform the measurement update using the new observation according to Algorithm 1B. Details pertaining to the dynamic boiler function $f(x(t), t)$ are given in Chapter 7 and is composed of the differential equations given in (8.2) and (8.3). Its flow of variables, summarised in Fig. 8.6.2, allows the sequential and parallel calculations of the various processes to be separated. The parallel calculations (enclosed in the dashed blue box), independent of the other discrete segments, are vectorised for faster execution and `IAPWS_IF97` table look-up, while the current fluid mass flow rate is calculated sequentially.

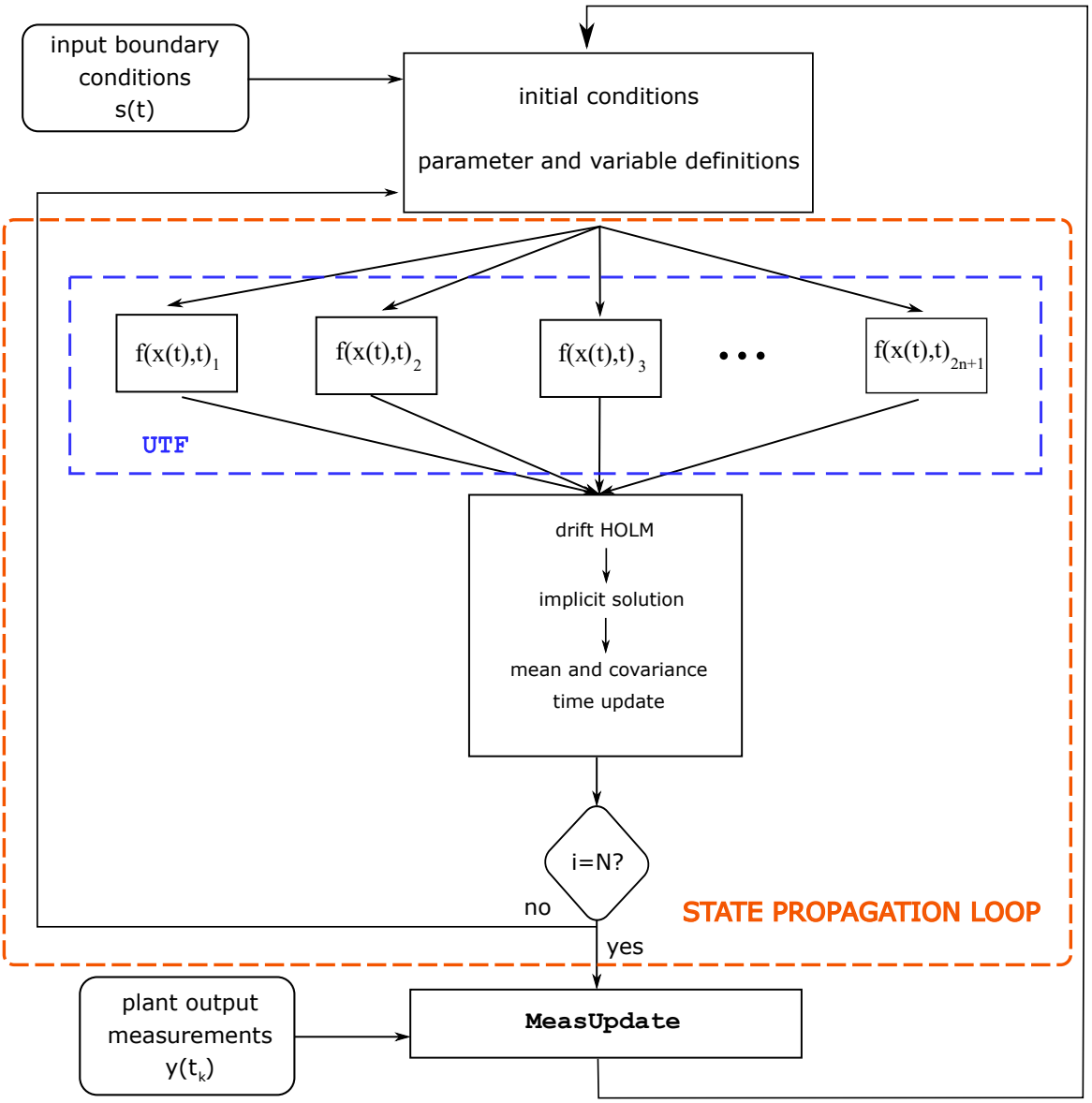


Figure 8.6.1: Overview of the main program layout.

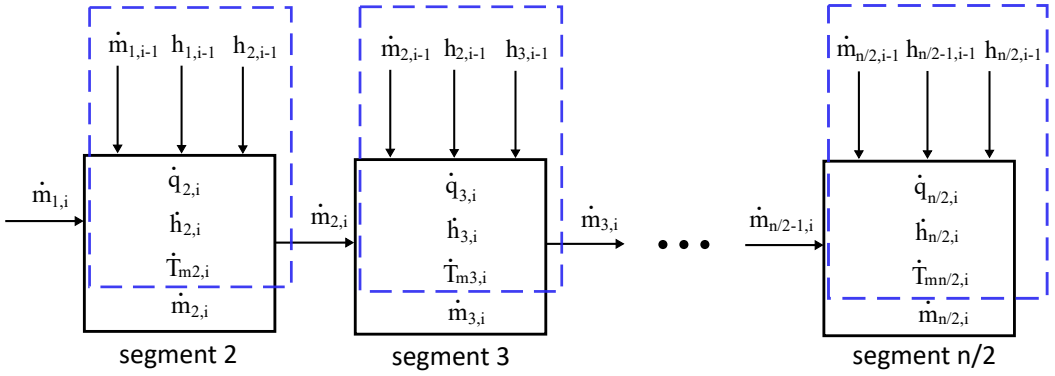


Figure 8.6.2: A diagram showing the flow of variables across the discrete segments of the boiler model.

8.7 Case Study 1: Fouling Simulation

This case study further describes the set-up of the filter specific to detecting fouling on S/H4, chosen arbitrarily as the faulty convective element. It aims to (1) test the performance of the proposed filter, and (2) demonstrate the concept of fault detection using the proposed filter. The CD-HCKF_Milstein is employed (Algorithms 2A and 1B) - suited to the system's large state vector and inherently stiff dynamics. Alternatively, algorithms 3A and 1B can be employed, however, since the additional effort required to implement the CD-HCKF_Milstein, which offers the potential for better estimates, is not much more, the CD-HCKF_Milstein will be used. Aspects of the filter designed in Sections 8.1 to 8.5 are applicable. To re-iterate, the integration step size is $\delta = 2$ s and the stochastic error model is based on the local error formulated in Section 8.5. The unscented parameters are $a = 0.65$, $\beta = 2$ and $\kappa = 0$, where a is chosen heuristically in the open interval $(0,1)$. The Flownex® model from Section 7.3 represents the 'real' plant and is used to construct filter observation data. Disturbances, uncertainties and faults are introduced into the Flownex® model, from which sampled outputs are corrupted with noise before being fed to the filter as measurements such that $\Delta = 10$ s. The Flownex® model is used in this manner on the following grounds:

- The Flownex® model has already been developed (for model validation purposes in Chapter 7) and is readily available.
- The Flownex® model includes pressure dynamics and can provide a more realistic account of plant behaviour.
- The Flownex® model can execute in reasonable time when used as a stand-alone simulation.
- Since the Flownex® model is the 'true' system in this case, the true internal state variables are available for comparison to the internal state variable estimates.

In Chapter 7 it was shown that the general behaviour of the mathematical MATLAB® model is comparable to both the Flownex® model and the actual plant output within typical boiler uncertainty (see Table 8.3.4). The reasons for using the MATLAB® model as the filter's model can therefore be summarised as follows:

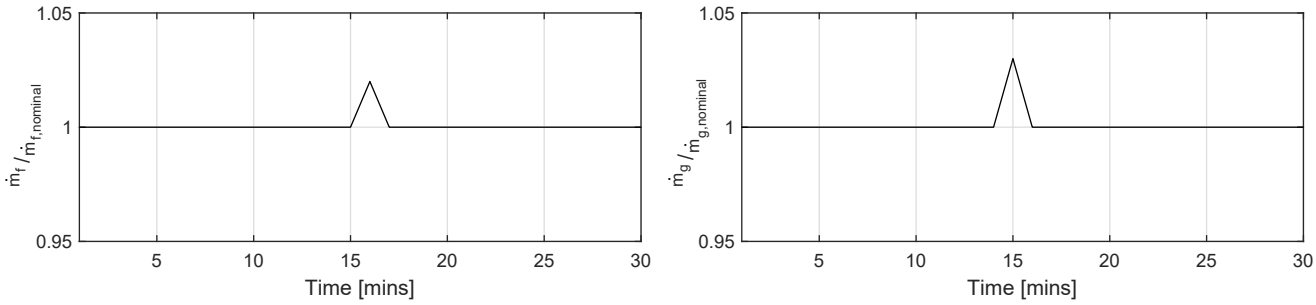
- The MATLAB® model is Markov - a necessary attribute of a model intended for use in a KF.
- The MATLAB® model runs faster than the PDE based Flownex® model.
- The MATLAB® model displays the correct boiler behaviour within standard uncertainty.

8.7.1 Corruption of Measurements from Flownex®

Given the measurement configuration in (8.4), simulated Flownex® data of the fluid temperature at the exit of the evaporator, S/H2 and S/H3 and S/H4 are sampled at 10 s intervals and corrupted with white Gaussian noise with a variance of $\sigma^2 = 1.73^2$ before being fed to the filter. This variance corresponds to the variance in Section 8.3.1.

8.7.2 Disturbances and Uncertainties in the Flownex® Model

Plant disturbances are modelled as sudden changes in plant conditions such as the input mass flow rate of flue gas, the mass flow rate of feedwater and the temperature of feedwater. To simulate the effect of a few disturbances in the system during the filtering process, changes to the variables of the Flownex® model are made according to the plots in Fig. 8.7.1.



(a) Change in the mass flow rate of feedwater. **(b)** Change in the input mass flow rate of flue gas.

Figure 8.7.1: Boiler disturbances.

Uncertainty in the modelling process is simulated by changing certain parameters in the filter model such that they no longer coincide with that of the Flownex® model. See Table 8.7.1 for a list of parameters changed in the filter model. These correspond to between 2% - 5% changes from the nominal.

Parameter	Difference in the MATLAB® Model
Ultimate analysis of H_2O in coal	+ 0.015
Ultimate analysis of H_2 in coal	- 0.015
Furnace supply air temperature	+ 5 °C
Coal mass flow rate	- 2 kg/s
Emissivity of flue gas particles	+ 0.02

Table 8.7.1: Model uncertainty.

8.7.3 Fault Simulation in Flownex®

The effect of external deposit formation on heat exchangers is understood using empirical correlations that depend on fuel and ash quality [130] and thus differs per boiler and operating conditions. This is represented by a fouling factor, given as $R_{foul} = \frac{1}{U} - \frac{1}{U_{foul}}$, where U is the overall heat transfer coefficient in (7.32) and U_{foul} is the heat transfer coefficient of the fouled heat exchanger. External fouling factors in coal fired boilers are typically in the range of $(0.005, 0.01)$ m²K/W [131], [132]. The effect of fouling is simulated in Flownex® by reducing the Colburn factor, j_H, in (7.19), associated with S/H4's heat exchanger chart by 6×10^{-3}(see Fig. 8.7.2), corresponding to $R_{foul} = 0.005$ and reducing its overall heat transfer coefficient by approximately 20 %. The effect of fouling is initiated as soon as the filter simulation begins.

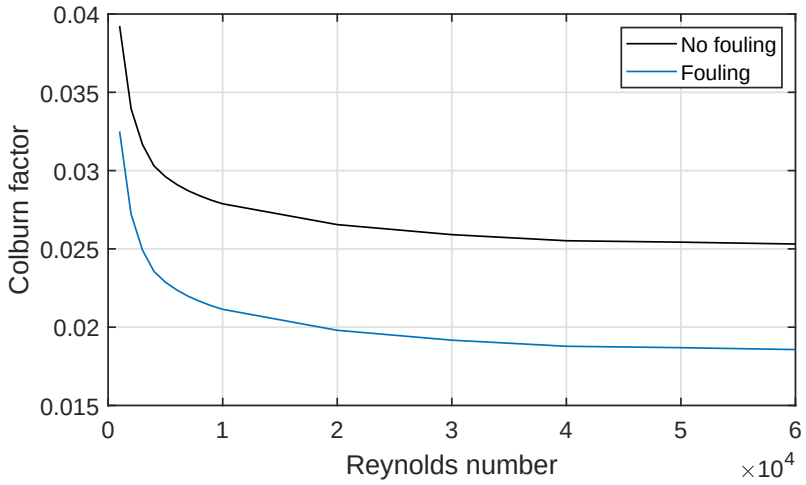


Figure 8.7.2: A graph showing the effects of fouling on Colburn factor.

8.7.4 Input Boundary Conditions and Initial Conditions

Input boundary conditions for both models are calculated using actual plant data in steady-state at various loads. These boundary conditions are listed explicitly in Sections 7.2 and 7.3. Some of these inputs correspond to those available from the actual plant in real-time [2] and are listed in Section 8.1. Before the filtering process begins, both models are run to steady-state at 70 % MCR without fouling on S/H4. The initial mean state vector estimate is taken as the steady-state value of the filter's model (MATLAB® model) in open loop, i.e., $\hat{x}_{0|0} = x_{steady-state}$. The initial state error covariance is given in Section 8.3.3. Starting at 70 % MCR, the filter is made to go through a load change over a 30 minute period according to Fig. 8.7.3. Disturbances are injected into the Flownex® model at times shown in Fig. 8.7.1.

[2] Plant data from the real-time EtaPro® database are used in Section 8.8

8.7.5 Results

The mean squared error, MSE $= \sqrt{\sum_{k=1}^{L}(x_{true,k} - \hat{x_k})^2/L}$, and the average standard deviation estimate, $\sqrt{\hat{P}} = \sqrt{\sum_{k=1}^{K}\hat{P}_k/L}$, of the last 40 states are shown in Fig. 8.7.4. Here L is the total number of observation points in the simulation trajectory. The trajectories of select state variables are plotted at each observation point in Figs. 8.7.5 and 8.7.6. The filter's innovation $K_k\Delta y_k = K_k(y(x(t_k)) - \hat{z}_k)$ of each state is plotted in Fig. 8.7.7.

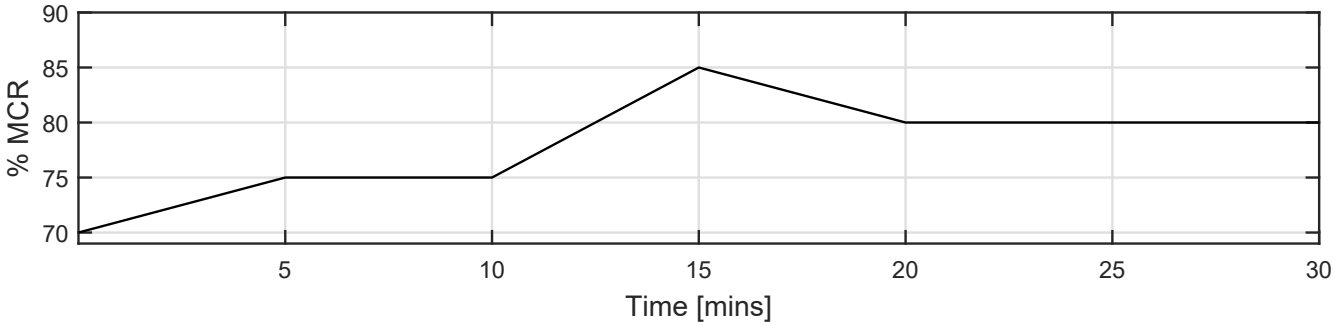
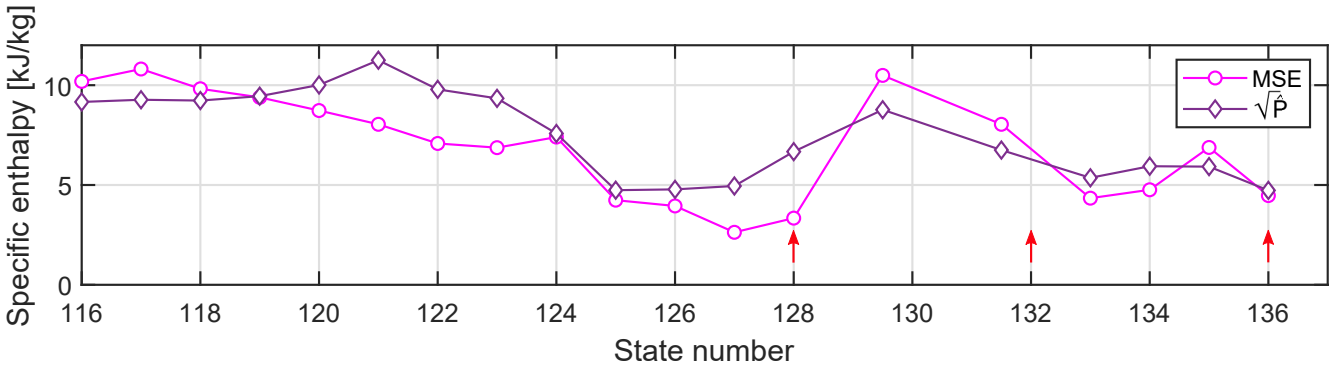
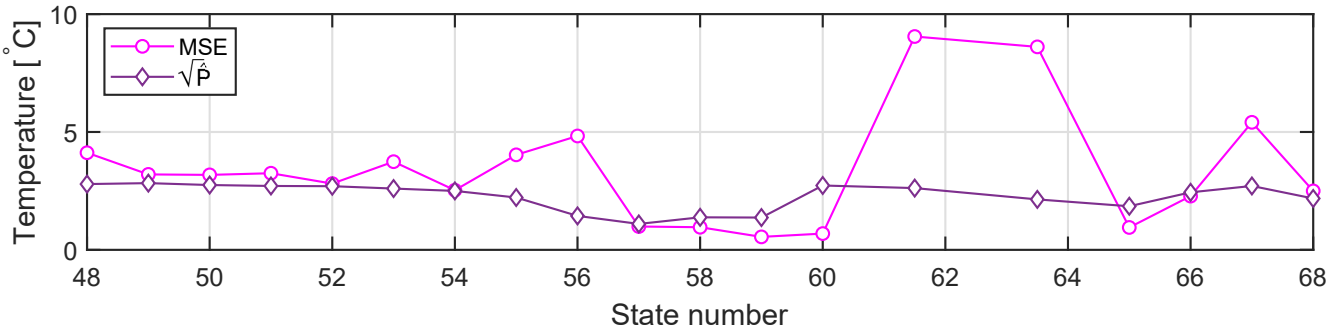


Figure 8.7.3: System demand.

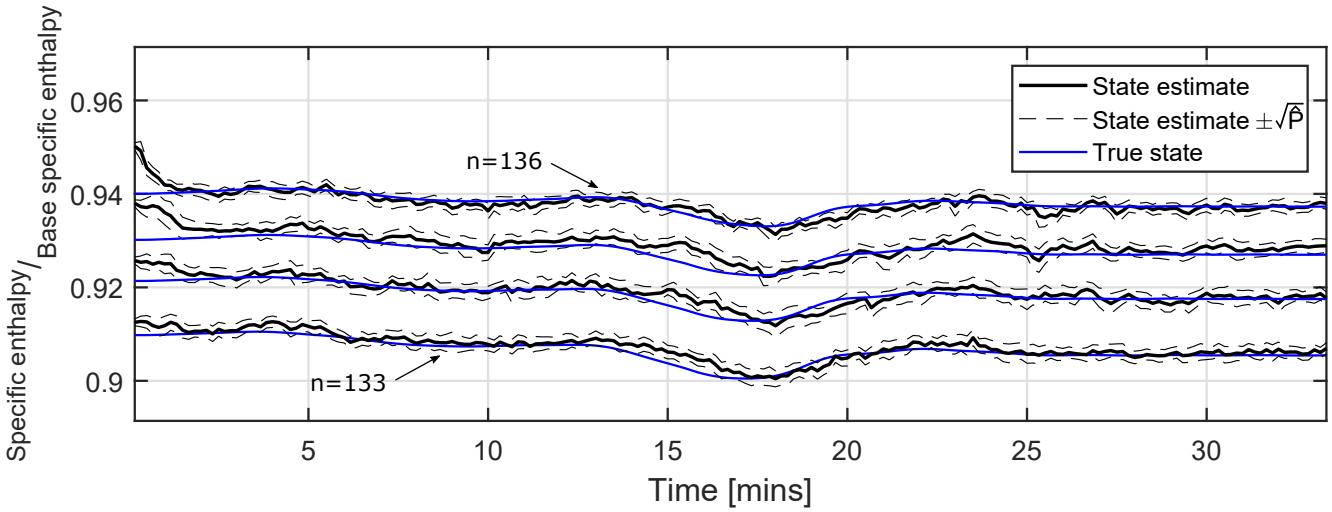


(a) State covariance errors of the specific enthalpy states.

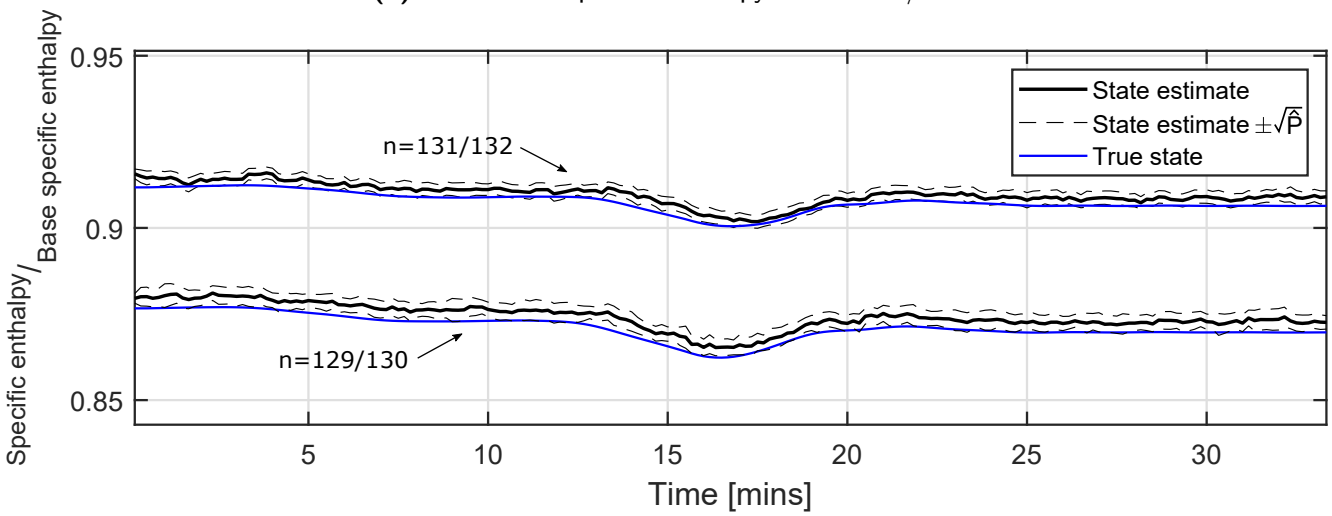


(b) State covariance errors of the wall temperature states.

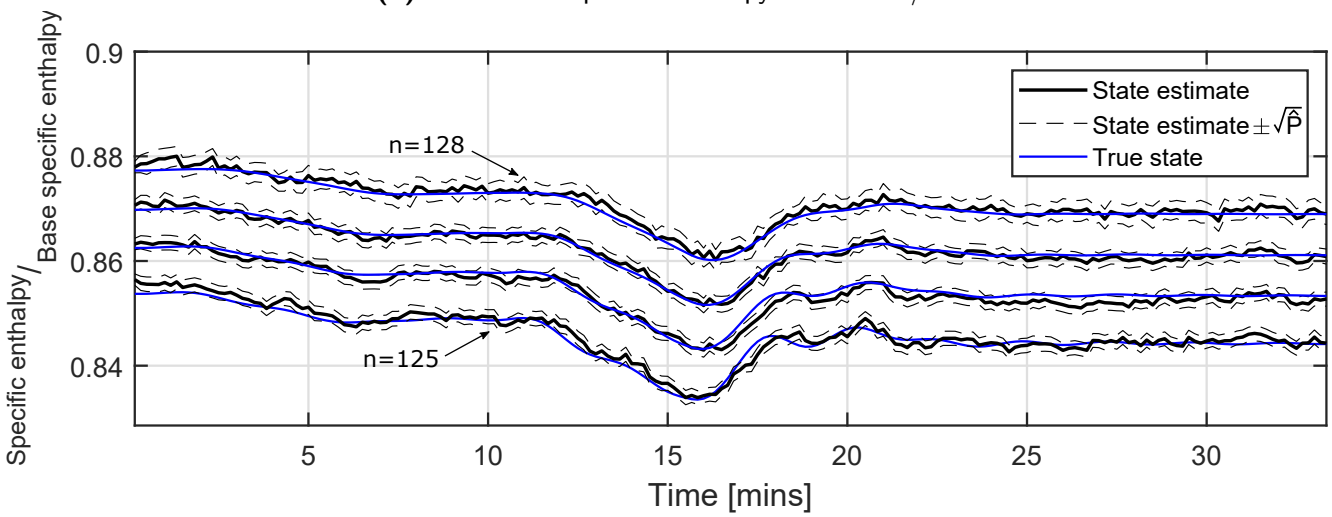
Figure 8.7.4: MSE and $\sqrt{\hat{P}}$ for the respective state variables over a 30 minute trajectory. The red arrows indicate which discrete segments are associated with measurements.



(a) Normalised specific enthalpy states of S/H4.

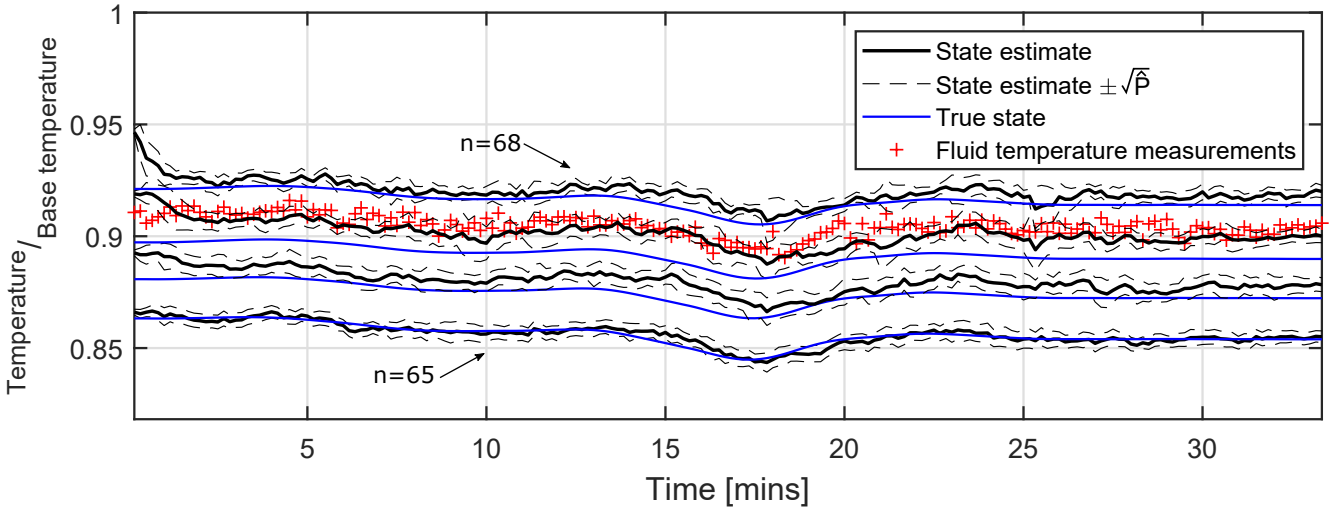


(b) Normalised specific enthalpy states of S/H3

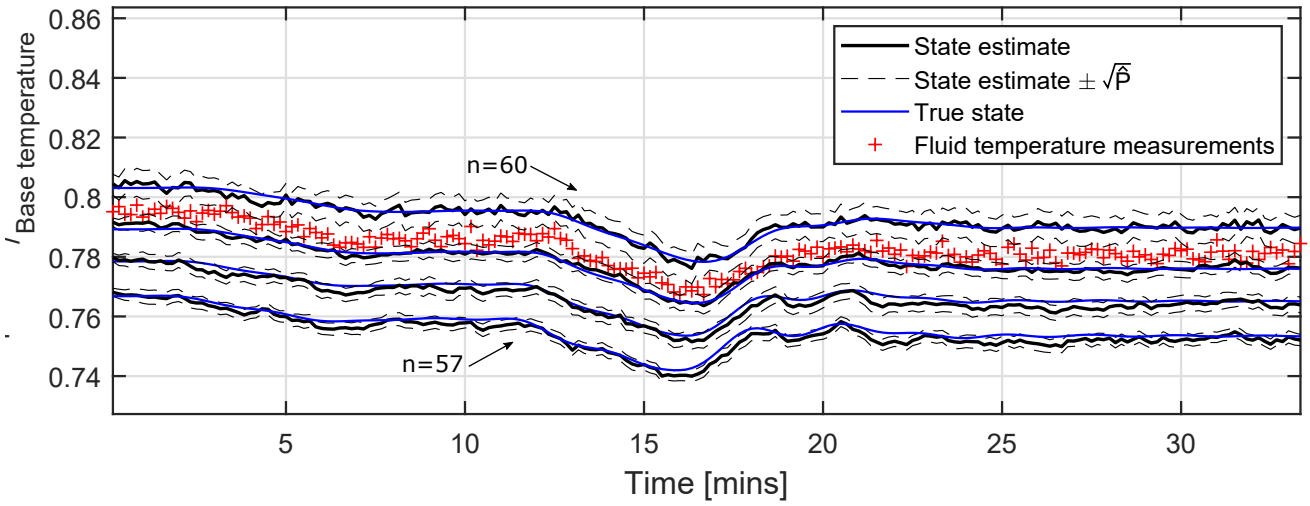


(c) Normalised specific enthalpy states of S/H2

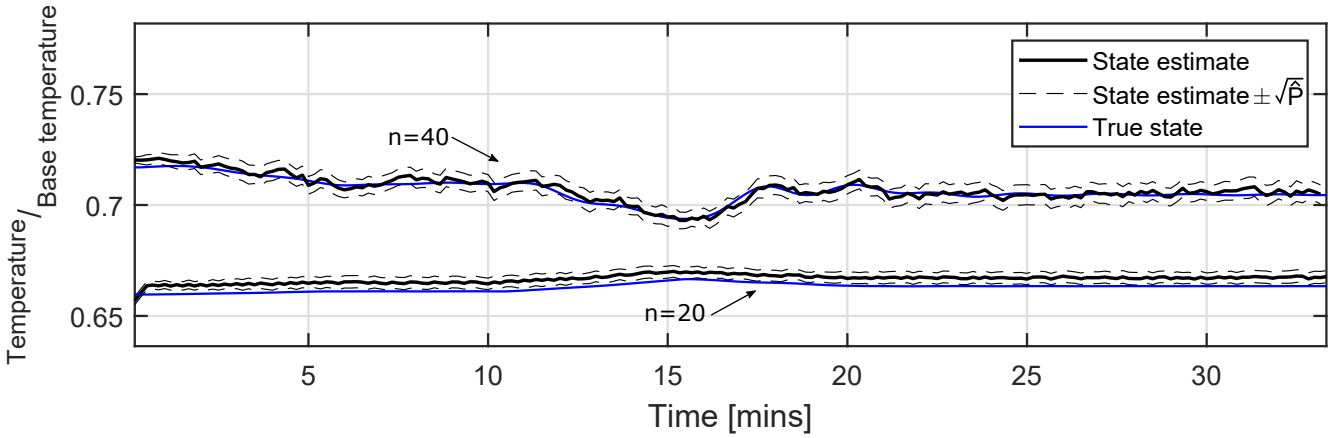
Figure 8.7.5: Normalised state estimates and their true trajectories for S/H4, S/H3 and S/H2.



(a) Normalised temperature states of S/H4

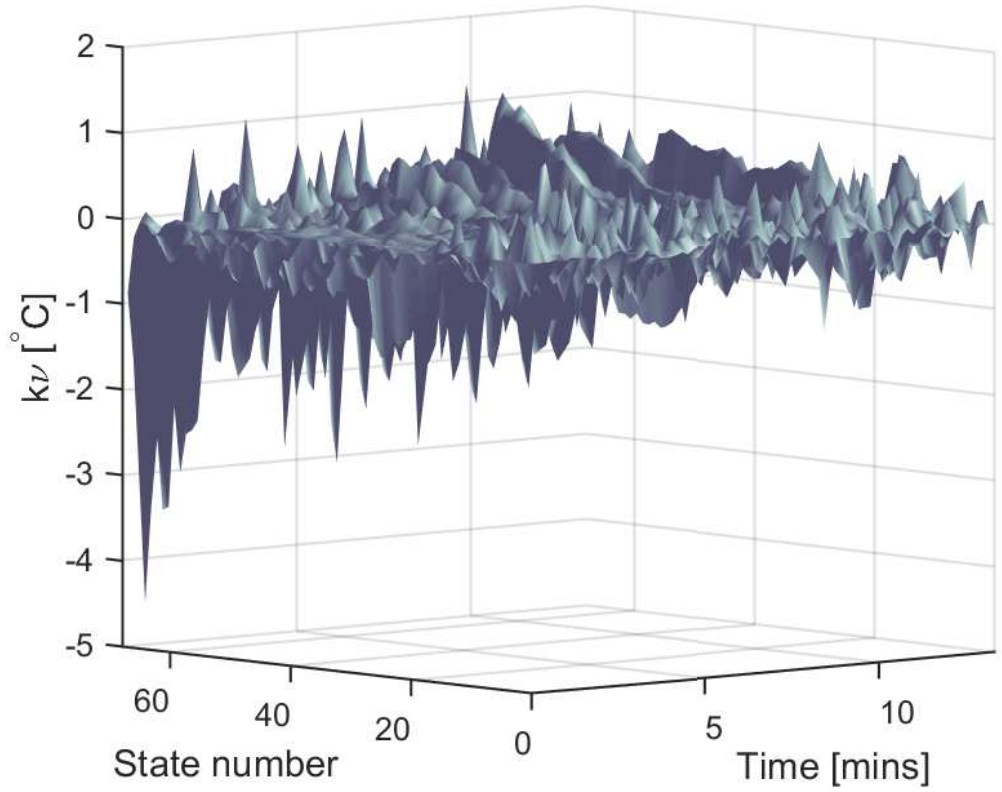


(b) Normalised temperature states of S/H2

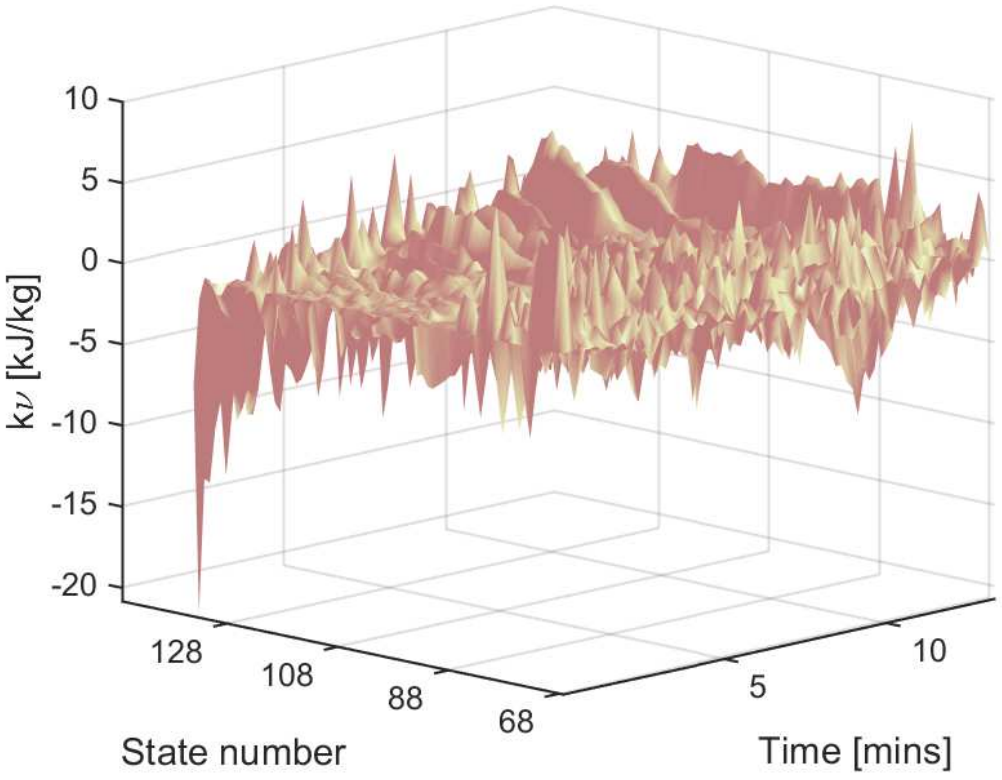


(c) Normalised temperature states of the evaporator and division wall.

Figure 8.7.6: Normalised state estimates and their true trajectories for S/H4, S/H2, the evaporator and the division wall.



(a) Filter innovation of the specific enthalpy states along the boiler length.



(b) Filter innovation of the wall temperature states along the boiler length.

Figure 8.7.7: Filter innovation of the wall temperature and fluid specific enthalpy states.

8.7.6 Discussion

According to Fig. 8.7.4 the average estimated standard deviation, $\sqrt{\hat{P}}$, of the filter coincides, in most cases, with the MSE of each respective state variable and is larger for state variables further away from the measurement points. It can thus be said that the covariances produced by the filter for most state variables are within reason. In the case of segments 61,62,63 and 64 (corresponding to S/H3), the high MSE is likely due to a discrepancy in the modelling of S/H3 together with the large non-linearity of the temperature state variables. The model discrepancy is further evident by the noticeable offset from the true state variables in Fig. 8.7.5b and indicates a likely bias error associated with S/H3 in the Flownex® model, which was limited to 2 discrete segments. For the remainder of the state variables in Figs. 8.7.5 and 8.7.6, the state estimates fall within one standard deviation of the true state variables along most of their trajectories.

According to the state estimates of S/H4 in Figs. 8.7.5a and 8.7.6a the total steam specific enthalpy change and wall temperature change in the presence of fouling are reduced from the total specific enthalpy change and total temperature change when no fault occurred. The specific enthalpy and temperature change of the other superheaters, however, remain the same. It can therefore be said that S/H4 alone sees an increase in thermal heat resistance. The reduction in specific enthalpy or temperature change can be used at the plant's supervisory level to flag the occurrence of fouling. In the simplest case, a threshold of, say, 10×10^{-3} p.u. could be imposed on these variables. A specific enthalpy difference change of $(13 \pm 2) \times 10^{-3}$ p.u. and a wall temperature difference change of $(25 \pm 5) \times 10^{-3}$ p.u. (read off Figs. 8.7.5a and 8.7.6a) would therefore flag S/H4 for fouling. This significant change in the state variables associated with SH/4 is also evident in the persistent one-sided filter innovation of states 67, 68, 135 and 136 in Figs. 8.7.7a and 8.7.7b, indicating that the heat transferred to this superheater is not as high as predicted. The non-zero innovation term associated with all of the state variables in these plots suggests that the measurement configuration does indeed allow observation of all state variables as theorised in Section 8.2.1.

8.7.7 Filter Efficiency

The total filter execution time over a 60 s trajectory for varying numbers of local computer cores used to run the filter is shown in Fig. 8.7.8. The 2n+1 (273) UTs are parallelized in the main program, depicted in Fig. 8.6.1. This means that the UTs are distributed amongst the various cores, reducing the time taken for the algorithm to run. The filter's total execution time over a 60 s trajectory as a function of the number of CPU cores, with a clock speed of 2.6GHz, is shown in Fig. 8.7.8. Although this plot implies that real-time execution cannot be achieved with the resources made available to the experiment, extrapolating the data indicates that 18 CPU cores or more are required to have the filter working in real time for a total of 136 states. This prediction is valid since the UTs dominate the computational load and also because the predicted number of CPU cores is less than 2n+1.

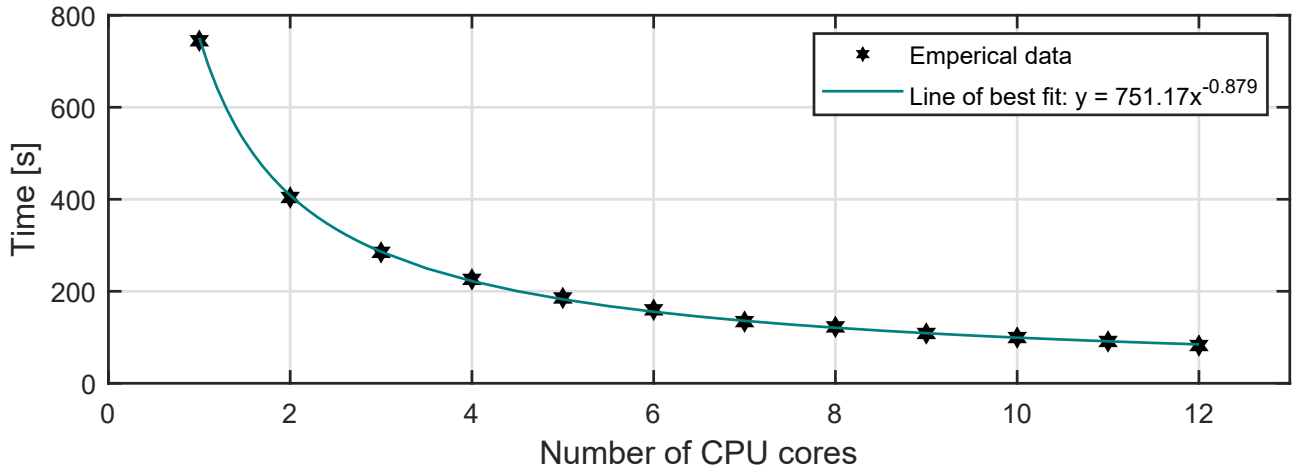


Figure 8.7.8: Filter execution time over a 60 s trajectory.

8.8 Case Study 2: State Estimation Applied to Measured Plant Data

In this case study the filter is given observation data from the EtaPro® database (in place of those from the Flownex® model in Section 8.7) over a period where the main steam outlet temperature was observed to undergo a significant increase. See Fig. 8.8.1. The steep increase in steam temperature is likely due to the reduction in spray water at around 40 minutes as can be seen in Fig. 8.8.2a. The period of interest includes a time of more or less constant demand to allow the main steam outlet temperature to reach steady-state. This is then followed by a ramp down in system demand as shown in Fig. 8.8.2b. The purpose of this investigation is to (1) test the performance of the proposed filter, and (2) estimate the state of the plant given actual measurements and thus verify whether its transient behaviour agrees with physical dynamics and input conditions. The input boundary conditions drives the filter boiler model and constitutes the feedwater mass flow rate, temperature and pressure, and the spray water mass flow rates of the 3 attemperators, taken from the EtaPro® database. The feedwater inlet pressure in Fig. 8.8.2c undergoes constant change that induces transient behaviour, ideal for properly assessing the performance of the filter. Since the system is identical to the one modelled in Section 8.7, the set-up of the filter in Sections 8.1 to 8.7 remain applicable. Four types of variables are referred to in this section and are defined below.

1. Primary state estimate - A state estimate from the filter belonging to either a wall temperature or fluid specific enthalpy variable in $\hat{x}$, i.e., either $\hat{T}_m$ or $\hat{h}$.

2. Secondary variable estimate - A secondary variable constructed by the filter and made available by the model, but not an actual state variable belonging to x, i.e., it does not have an associated innovation term). By the detectability property of the filter these variables are also estimates but constructed indirectly.

3. Filter measurement - Measured variables from the plant obtained through the EtaPro® database and used in the filter's measurement update process.

4. Measurement - Measured variables from the plant obtained through the EtaPro® database and not used in the filter.

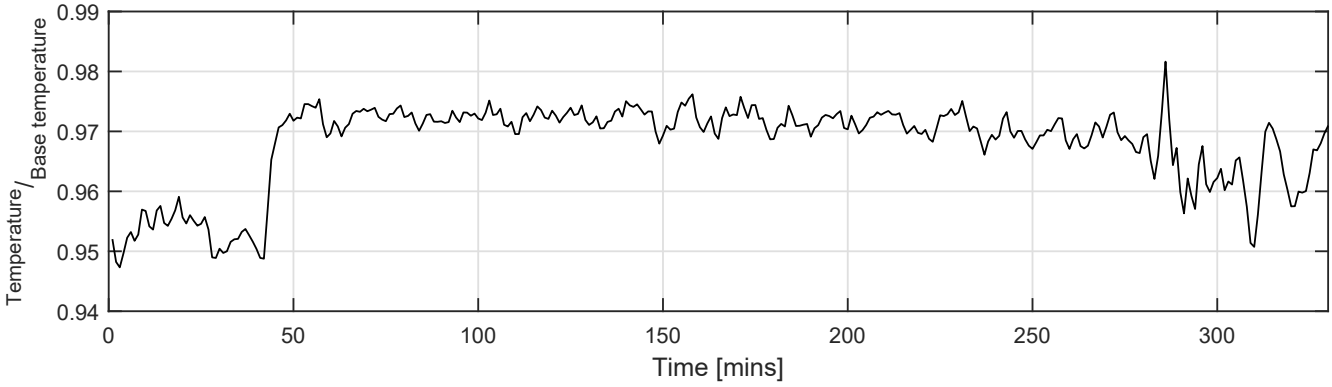
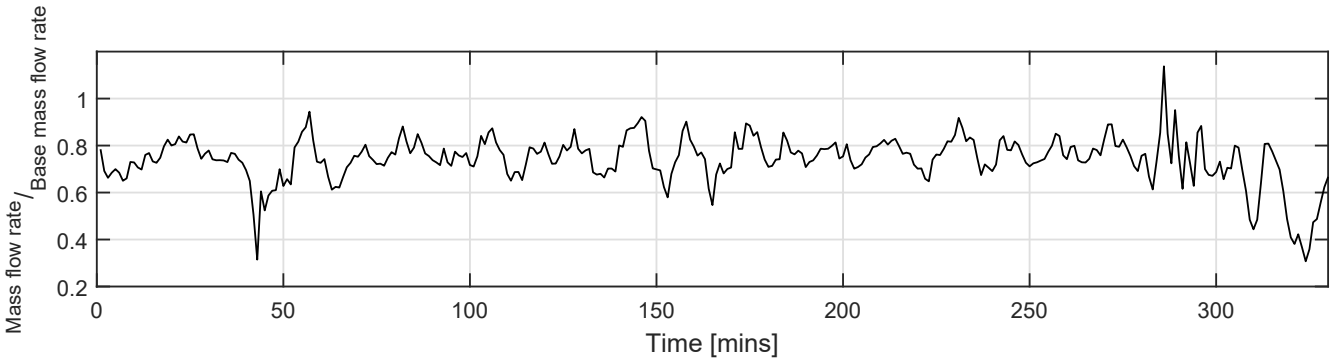
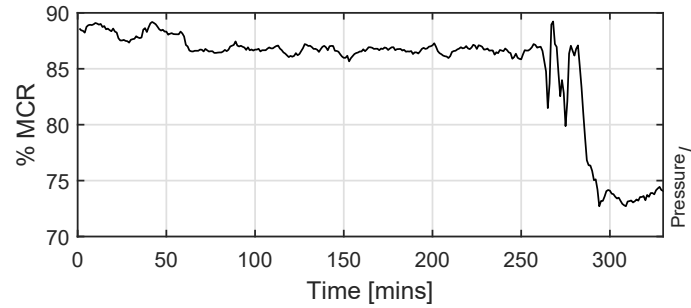


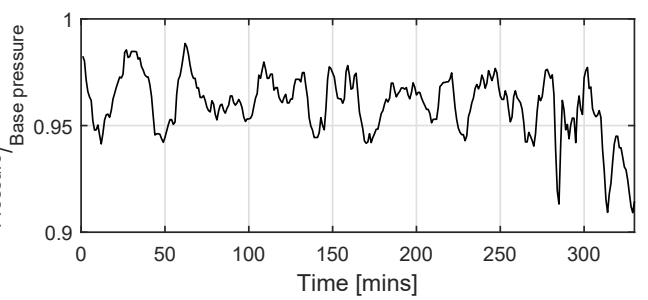
Figure 8.8.1: Normalised main steam temperature at the exit of S/H4. A steep temperature rise can be seen at 40 minutes.



(a) Normalised mass flow rate of ATT3 spray water.



(b) System demand.



(c) Normalised feedwater pressure.

Figure 8.8.2: Input and boundary conditions of the boiler model.

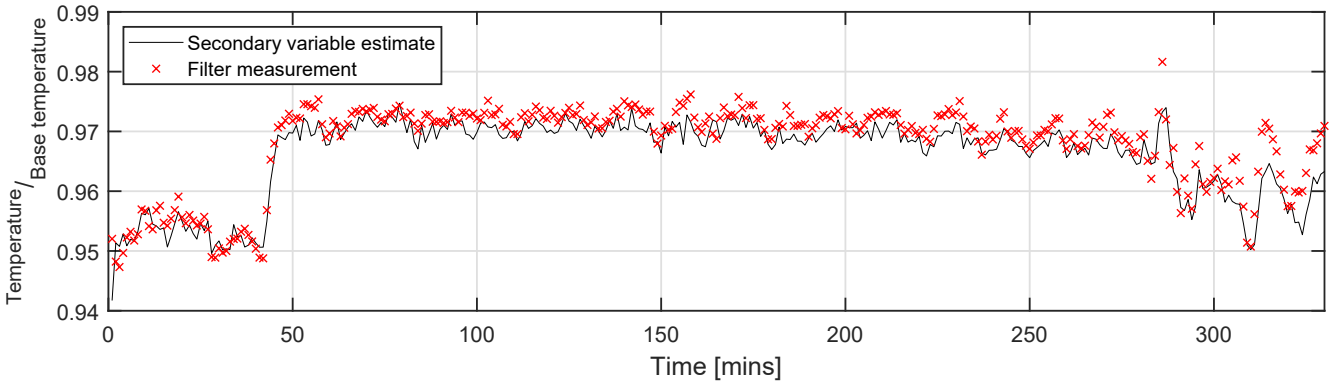
The redundant measurements associated with the 4 respective subheaders at the exit of S/H2 and S/H3 should ideally be the same, but due to slight variation in the axial direction of the boiler as well as systematic measurement error, they typically vary by a few degrees. These measurements are averaged before being fed to the filter

and have a standard deviation of $\sigma = 2.24^{\circ}\text{C}$ and $\sigma = 2.53^{\circ}\text{C}$ respectively. Based on the standard deviations of these measurements, the measurement noise covariance is chosen as $R_k = 2.5^2 I_{\text{m}}[^{\circ}\text{C}]^2$. The filter makes an observation every 10 s, i.e., $\Delta = 10$ s, and has an integration step size of $\delta = 2$ s. The measurement function is given by (8.4). The process covariance and initial state error covariance goes according to that given in Sections 8.3.2 and 8.3.3.

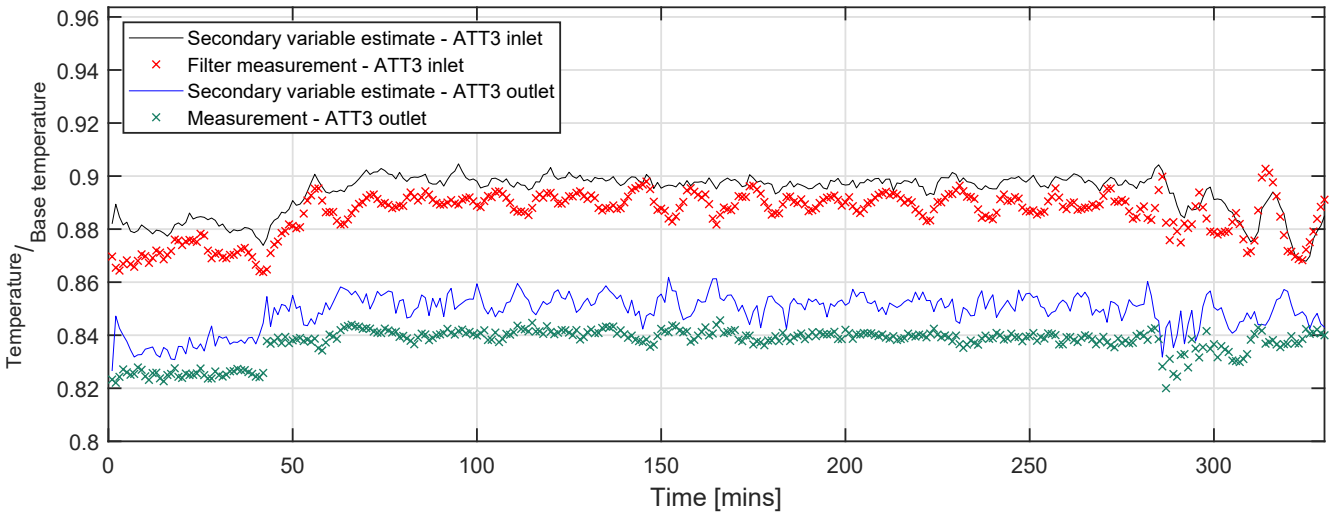
8.8.1 Results and Discussion

The steam temperatures at the exit of the superheaters and vertical evaporator before and after attemperation are shown in Figs. 8.8.3a to 8.8.4b. Fig. 8.8.3a shows secondary variable estimates and the corresponding filter measurements for S/H4. Fig. 8.8.3b shows fluid temperature at the exit of S/H3 before and after attemperation. Here the secondary variable estimates and the corresponding filter measurements at the inlet of ATT3 are shown. On the same plot is the fluid temperature at the outlet of ATT3 according to measurements (not fed to the filter) and according to the filter. This is shown in order to compare internally constructed data from the filter to measurements received directly from the plant. A similar set of plots are shown at the exit of S/H2 and the vertical evaporator in Figs. 8.8.4a and 8.8.4b respectively. In Figs. 8.8.3a and 8.8.4a the measurements fed to the filter as well as the measurements not fed to the filter agree with the corresponding filter constructed variable. This indicates that the measurements of these superheaters agree with the physical processes and the many associated uncertainties that the filter has been made aware of. In Fig. 8.8.3b, however, the set of measurements are both offset from the secondary filter estimates, similar to the simulation scenario in Section 8.7 (see Fig. 8.7.5b). It was explained in Section 8.7 that the high MSEs associated with the state variables of S/H3 are likely due to modelling error in S/H3 of the Flownex® model. The fact that the discrepancy between measurements and state estimates exists points to an error in the MATLAB® model rather than the Flownex® model, or an error in both models. The higher heat transfer characteristics of S/H3 together with larger unaccounted for parameter uncertainty is the likely cause of this apparent divergence. Apparent divergence is a common problem in the event of excessive model error. Another possible reason for this discrepancy is large bias error on the measurements of S/H3, however, the mutual offset in the estimates strongly suggests modelling error. A measurement discrepancy can in fact be seen in Fig 8.8.4b where the filter measurements and their corresponding secondary variable estimates at the inlet of ATT1 correspond; however, the measurements and their corresponding secondary variable estimates at the outlet of ATT1 do not agree. The filter suggests, based on measurement error, process error, input boundary conditions and plant dynamics, that the fluid temperature at the exit of ATT1 should be lower. In the context of this sub-system it is obvious why this makes sense, i.e., the fluid temperature at the exit of the attemperator should be lower than the fluid temperature at the inlet of the attemperator. Fig. 8.8.4c shows wall temperature state estimates at the exit of S/H4, S/H3 and S/H2 in comparison to their corresponding plant measurements (not given to the filter). Here it can be seen that the primary states estimates of S/H2 and S/H4 agree within 1 standard deviation of the actual plant measurements. The wall temperature estimates at the exit of S/H3, however, do not agree. The full set of wall temperature and fluid specific enthalpy estimates across the boiler length is shown in Figs. 8.8.5a and 8.8.5b. It can be seen here

that the filter is able to construct internal variables that cannot be measured directly. The correctness of all of these state estimates cannot be verified as was done for the state variables in the previous plots; however, based on the aforementioned analyses, there is confidence in the filter's ability to construct state mean and covariance estimates provided that the model is satisfactory. A general observation when looking at these plots is that the wall temperature profile across the boiler length corresponds to typical temperatures across the various phases of heat transfer regimes, becoming flat in the evaporator region, while specific enthalpy shows a more consistent increase as expected. The filter furthermore executes consistently over the 5 hour period without losing positive-definiteness and thus, again, confirms the application of theory from Part 1 of this thesis. It should be noted that the large shift of the estimates from their initial estimates shown in Figs. 8.8.3a to 8.8.4c is because the filter did not begin in steady-state with the plant input boundary conditions from the EtaPro® database. A similar shift is expected at the start of on-line implementation.

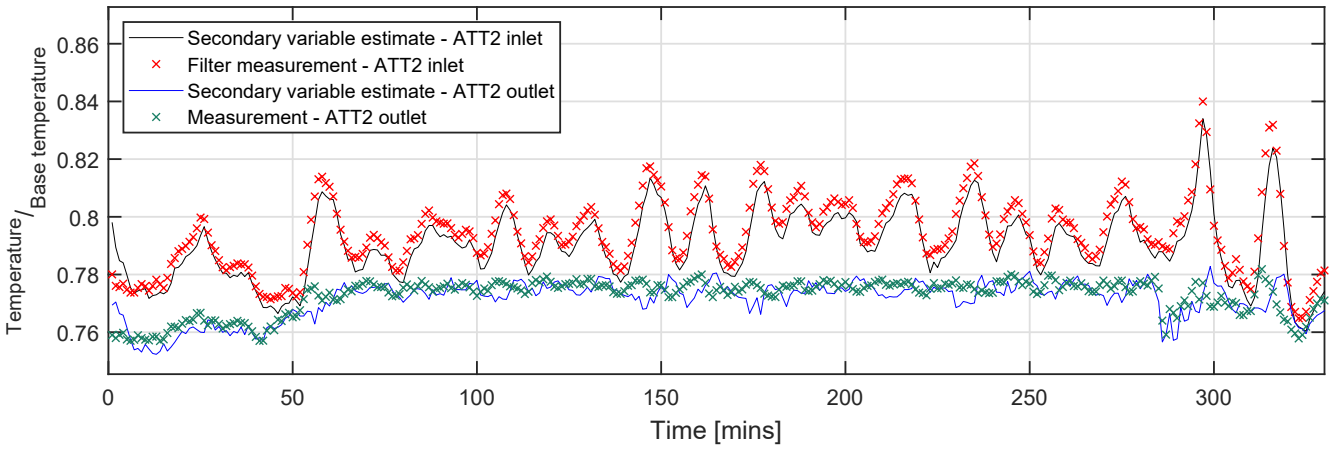


(a) Normalised main steam temperature at the exit of SH/4.

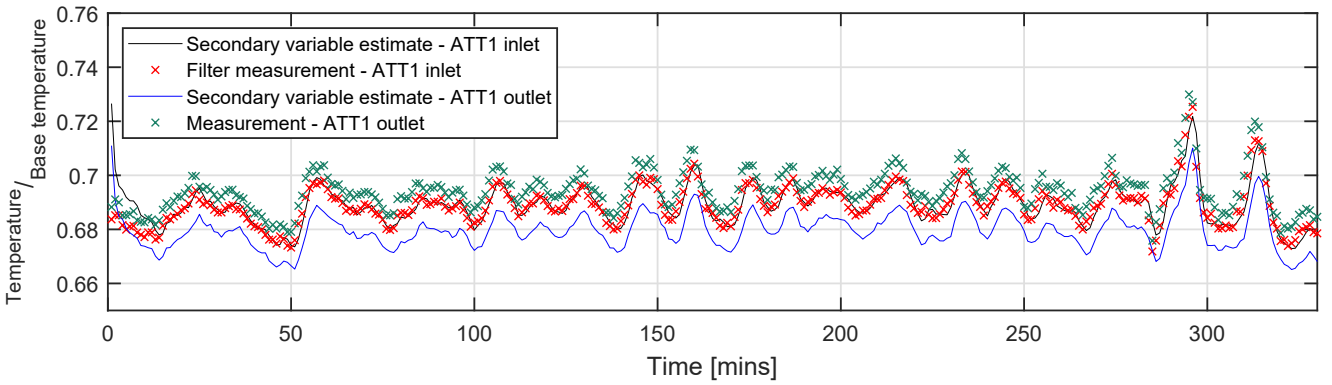


(b) Normalised fluid temperature at the exit of SH/3 before and after attemperation.

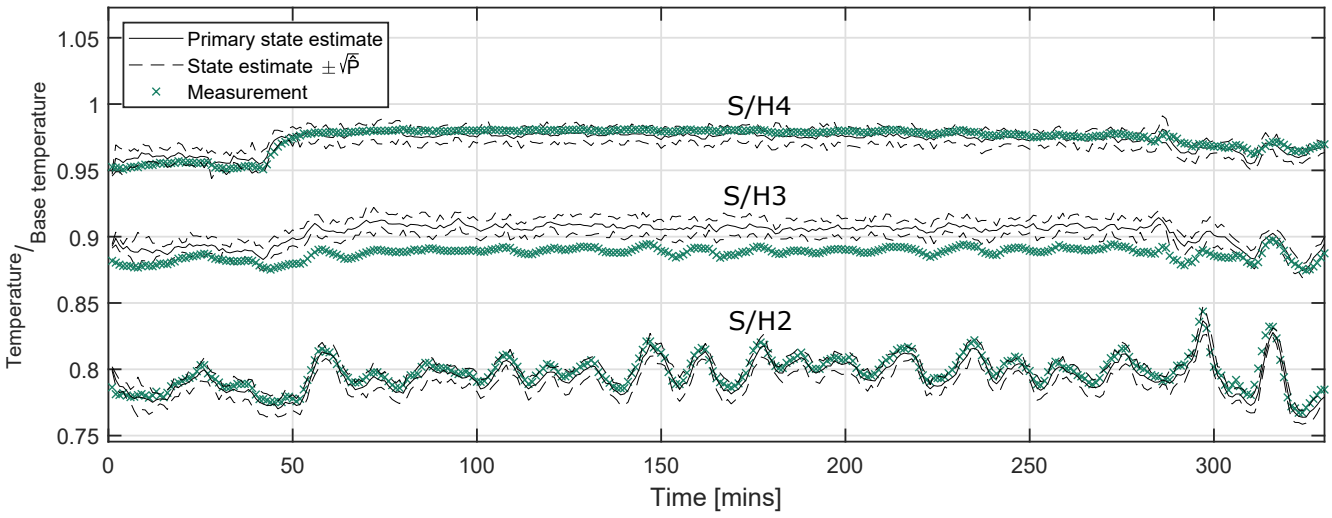
Figure 8.8.3: Normalised fluid temperature estimates and measurements.



(a) Normalised fluid temperature at the exit of SH/2 before and after attemperation.

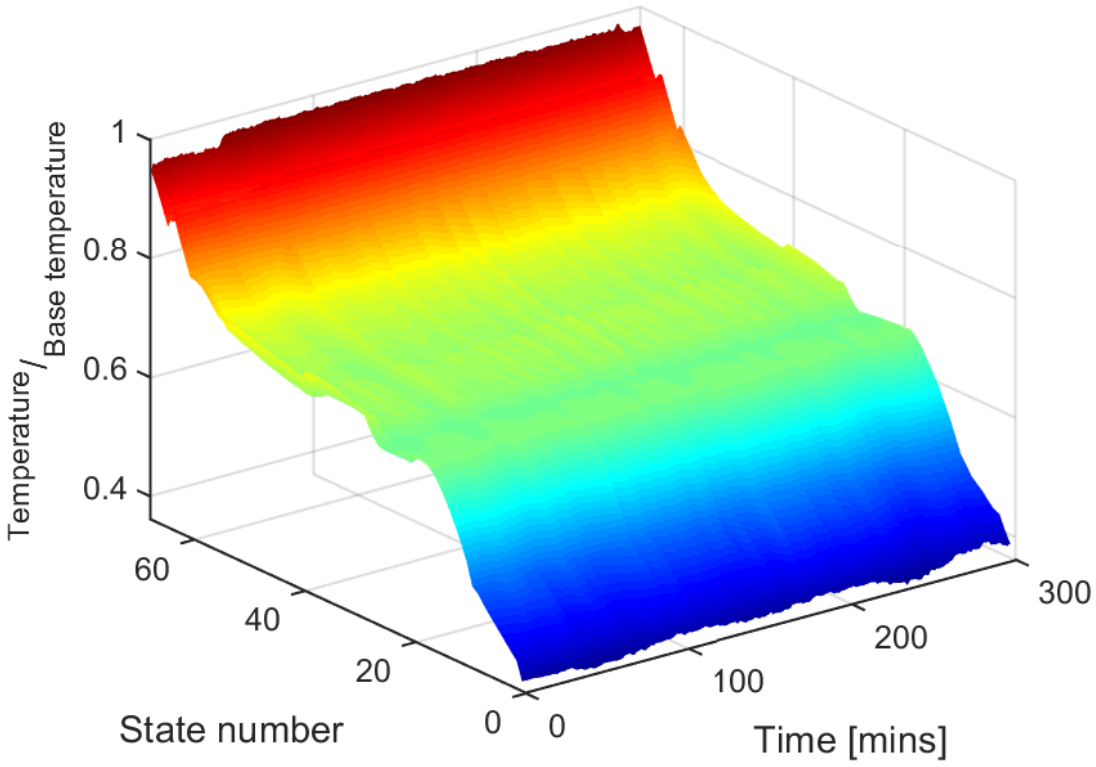


(b) Normalised fluid temperature at the exit of the vertical evaporator before and after attemperation.

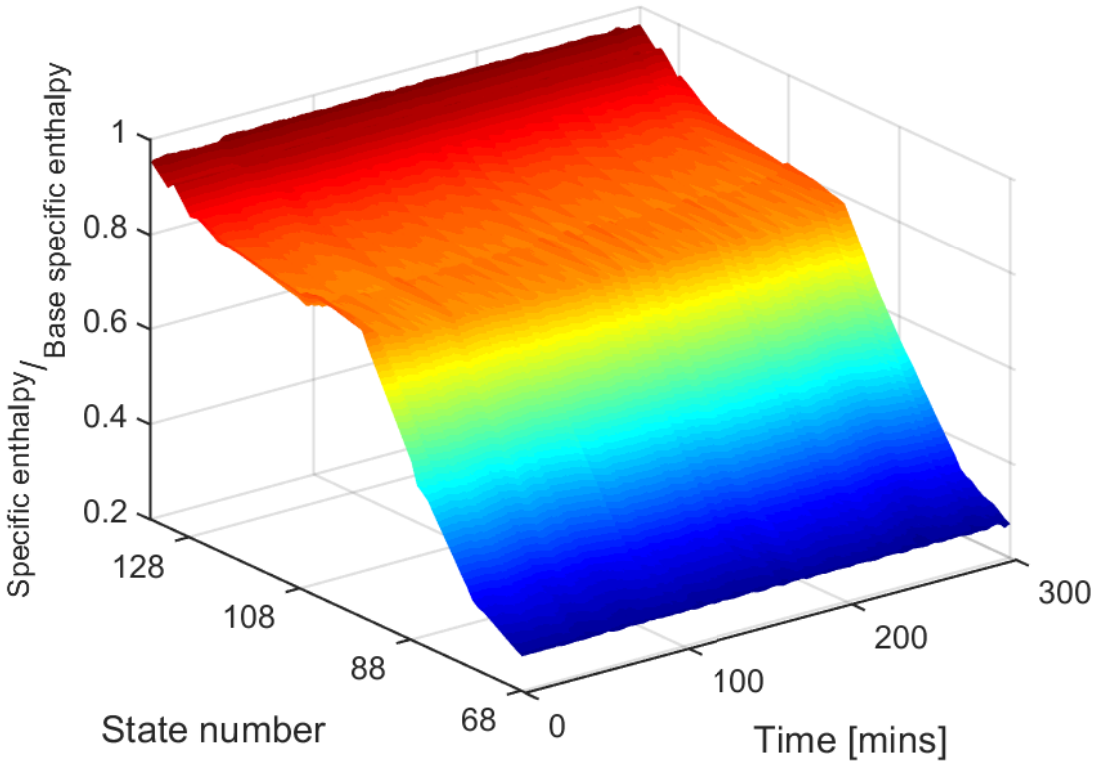


(c) Normalised wall temperature states at the exit of SH/2, S/H3 and S/H4.

Figure 8.8.4: Normalised fluid temperature estimates, wall temperature estimates and measurements.



(a) Normalised wall temperature states across the boiler length.



(b) Normalised fluid specific enthalpy states across the boiler length.

Figure 8.8.5: Normalised wall temperature and fluid specific enthalpy states across the boiler length.

9 Conclusion

9.1 Findings and Contributions

In part 1 of this thesis, three algorithms were developed, the CD-HCKF, the CD-HCKF_EM and the CD-HCKF_Milstein. The CD-HCKF uses the unscented transforms to extract an equivalent of the system Jacobians, termed the drift and measurement HOLM, which results in a coupling between simulation and the filtering process. The benefit of doing this is that (1) linearly implicit simulation can be employed without the need to analytically calculate the drift Jacobian, and (2) symmetric equations in the extended Kalman filtering process can be employed. This results in a more efficient (assuming that the system in question has non-trivial Jacobians) and numerically stable filter with CD-UKF level accuracy. This filter is published in [4]. The CD-HCKF_EM and CD-HCKF_Milstein have the same aforementioned properties as the CD-HCKF, but are grounded in an SDE that includes a stochastic state-dependent local error model. This local error model systematically accounts for simulation error when large temporal integration step sizes are employed, and is based on an inexpensive and relatively accurate error estimate that was developed in this thesis. The CD-HCKF_Milstein is a higher-order variant of the CD-HCKF_EM and was shown to be marginally more robust in comparison to the CD-HCKF_EM. Compared to the CD-HCKF, The CD-HCKF_EM and CD-HCKF_Milstein allow for coarser simulation steps, and hence reduce the time taken for the filter to execute. The CD-HCKF_EM and CD-HCKF_Milstein are useful in situations where computation time is limited, such as in real-time applications. Having all the CD-HCKF attributes, the CD-HCKF_EM and CD-HCKF_Milstein are also suited to stiff large-scale systems with non-trivial Jacobian calculations and a susceptibility to incurring computational error.

In part 2 of this thesis, estimation of 136 wall temperature and fluid specific enthalpy states associated with the

main components of a once-through utility boiler was performed using the CD-HCKF_Milstein. This application formed a core motivation for the development of the filters in Part 1. Two models of the boiler were developed, comprising the furnace, economiser, division wall, evaporators and superheaters. The first is a mathematical MATLAB® model, well suited to Kalman filtering, and the other a PDE-based model in Flownex® whose main purpose was to validate the mathematical model. On application of the filter to a simulated fault scenario, the estimated state trajectories were found to lie within one estimated standard deviation of their true state trajectories for most of the state variables. Furthermore, the average estimated state standard deviations were found to align with the MSEs of each respective state. This means that the filter is able to construct good state mean and covariance estimates. It was then shown how these state mean and covariance estimates can be used to detect fouling in the convection pass. Fouling detection in the convection pass using continuous-discrete Kalman filtering is published in [3]. A study into the efficiency of the filter indicates that the filter is capable of being implemented on-line if the number of cores in the local PC is 18 or higher. This makes on-line filtering viable in utility industries, where such resources are likely to be accessible. To validate the workability of the filter on an actual plant, data from the EtaPro® database were fed to the estimator. The filter was found to successfully construct internal state estimates. This excludes state estimates associated with S/H_3 which exhibited apparent divergence that manifested as a result of model error, highlighting how state estimation can in fact produce unreliable state estimates if there is too much error in the system model that is not (properly) advertised to the filter. Despite the discrepancy in S/H_3, the filter estimated all of the remaining state variables along the boiler length over a 5 hour period, avoiding Jacobian calculations and numerical stability issues while boasting the potential for on-line implementation with commercially reasonable computational resources.

9.2 Recommendations

With regards to theoretical developments, it is recommended to show more rigorously that the central Jacobian is a good approximation in the solution of all sigma points, and to quantify the limitations of this assertion. Furthermore, it is suggested to formalise the stability properties of the CD-HCKF_Milstein as well as develop a better integration scheme for the diffusion terms. The latter will likely exploit the fact that the diffusion coefficient is proportional to the drift function and hence higher-order integration schemes are possibly some function of the drift Jacobian. For the application, it is recommended that the models developed in this thesis be used to their full potential by appending other key variables to the state vector, such as the mass fractions of moisture and hydrogen in coal. This will likely require a reconfiguration of the measurement function. The capability of the filter can further be extended by including more refined element models. One obvious example is the furnace model, which can be partitioned in order to produce more localised combustion behaviour. This will allow temperature profiles within the furnace to be estimated and used for boiler optimisation.

A

Stochastic Systems: A Review

A.1 Concepts and Definitions in Probability and Statistics

A.1.1 Random Variables

Let Ω be the space of all possible events $\{w_1, w_2 \ldots\}$ belonging to the probability space (Ω, A, P), where $A \subset \Omega$ is a collection of events and P the probability measure, then $X{:}\Omega \rightarrow \mathbb{R}$ is a measurable function that maps all possible events to the measurable space. Define a distribution function associated with random variable X for all real numbers, x, as

$$F_X(x) = P_X(-\infty, x) = P(\{w \in \Omega : X(w) < x\}). \tag{A.1}$$

If X is a continuous random variable then the density function is $p(x) = F'_X(x)$. The density function of the uniform distribution $X \sim U(a, b)$ is written as

$$p(x) = \begin{cases} \frac{1}{b-a}, & \text{if } x \in [a, b] \\ 0, & \text{otherwise.} \end{cases} \tag{A.2}$$

The density function of the exponential distribution is written as

$$p(x) = \begin{cases} \lambda \exp(\lambda x), & x \geq 0 \\ 0, & x < 0, \end{cases} \tag{A.3}$$

where $\lambda > 0$, and the density function of the Gaussian distribution $X \sim \mathcal{N}(\mu, \sigma^2)$ is written as

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} \right), \tag{A.4}$$

where μ is the mean and σ the standard deviation of the distribution.

A.1.2 Correlated Random Variables

It will later become necessary to simulate random variables, or a pair of correlated normal random variables, X_1 and X_2, with mean $E\{X_1\} = E\{X_2\} = 0$, covariances $E\{X_1^2\} = \delta$, $E\{X_2^2\} = \frac{\delta^3}{3}$ and cross-covariance $E\{X_1, X_2\} = \frac{\delta^2}{2}$. Given two independent Gaussian random variables $G_1 \sim \mathcal{N}(0,1)$ and $G_2 \sim \mathcal{N}(0,1)$ the correlated random variables are:

$$\begin{aligned} X_1 &= \sqrt{\delta} G_1 \\ X_2 &= \frac{1}{2}\delta^{\frac{3}{2}}\left(G_1 + \frac{1}{\sqrt{3}} G_2 \right) \end{aligned} \tag{A.5}$$

A.1.3 First and Second Moments

The first and second moments of a random variable will frequently be used and are defined as the mean and variance of its distribution function. For a continuous random variable the first moment is defined as

$$\mathrm{E}\{X\} = \int_{-\infty}^{\infty} x p(x) dx = \mu, \tag{A.6}$$

where $p(x)dx$ is the probability of X being in the range x and $x + dx$ and μ is the mean. The second central moment of X, also referred to as the variance, is calculated as

$$\mathrm{var}\{X\} = \mathrm{E}\{(X - \mathrm{E}\{X\})^2\} = \sigma^2, \tag{A.7}$$

where σ is the standard deviation and is a measure of the spread of the data in X around μ. For two correlated (or jointly distributed) random variables X and Y, the covariance is defined as

$$\mathrm{cov}\{X, Y\} = \mathrm{E}\{(X - \mathrm{E}\{X\})(Y - \mathrm{E}\{Y\})\} = \sigma_{XY}^2. \tag{A.8}$$

A.1.4 Some Properties of the Expectation and Covariance

For a set of random variables $X \in \mathbb{R}^n$, $Y \in \mathbb{R}^m$ and $Z \in \mathbb{R}^p$ and scalars $a \in \mathbb{R}$, $b \in \mathbb{R}$ and $c \in \mathbb{R}$ it follows that:

1. $\mathrm{E}\{X + Y\} = \mathrm{E}\{X\} + \mathrm{E}\{Y\}$
2. $\mathrm{E}\{aX\} = a\mathrm{E}\{X\}$

3. if $X = c$ then $\mathrm{E}\{X\} = c$
4. $\mathrm{E}\{\mathrm{E}\{X\}\} = \mathrm{E}\{X\}$
5. $\mathrm{E}\{XY^T\} = \mathrm{E}\{X\}\mathrm{E}\{Y\}^T$ if X and Y are independent
6. $\mathrm{cov}\{X, c\} = 0$
7. $\mathrm{cov}\{X, X\} = \mathrm{var}\{X\}$
8. $\mathrm{cov}\{X, Y\} = \mathrm{cov}\{Y, X\}^T$
9. $\mathrm{cov}\{aX, bY\} = ab\mathrm{cov}\{X, Y\}$
10. $\mathrm{cov}\{X + a, Y + b\} = \mathrm{cov}\{X, Y\}$
11. $\mathrm{cov}\{aX + bY, Z\} = a\mathrm{cov}\{X, Z\} + b\mathrm{cov}\{Y, Z\}$
12. $\mathrm{cov}\{X, Y\} = \mathrm{E}\{XY^T\} - \mathrm{E}\{X\}E\{Y\}^T$
13. $\mathrm{E}\{X^4\} = 3(\mathrm{var}\{X\})^2$ for a normal distribution

A.1.5 Conditional Expectations

In Bayesian estimation, the conditional expectation is used to calculated information about a random variable after an observation is made. The probability density of $x \in \mathbb{R}$, given that some event $A = \{w \in \Omega : a \leq X(w) \leq b\}$ has occurred for the continuous-time case is

$$p(X = x|A) = \begin{cases} \frac{p(x)}{\int_a^b p(s)ds}, & a \leq x \leq b \\ 0, & x < a, x > b. \end{cases} \tag{A.9}$$

This gives the expectation of X as

$$\mathrm{E}\{X|A\} = \int_{-\infty}^{\infty} xp(x|A)dx = \frac{\int_a^b xp(x)dx}{\int_a^b p(x)dx}, \tag{A.10}$$

where $p(X = x|A)$ is notationally simplified to $p(x|A)$.

A.1.6 Weak and Strong Convergence Criteria

Given an infinite sequence of random variables $X_1, X_2, X_3, \ldots, X_n$, where $X = \lim_{n\to\infty} X_n$ there exists two different forms of convergence. The first is strong convergence. This requires the behaviour of X_n to closely resemble X such that for $\mathrm{E}\{|X_n|\} < \infty$, where $n = 1, 2, \ldots$, and $\mathrm{E}\{|X|\} < \infty$

$$\lim_{n\to\infty} \mathrm{E}\{|X_n - X|\} = 0. \tag{A.11}$$

Another form of strong convergence is mean square convergence where

$$\lim_{n\to\infty} \mathrm{E}\{|X_n - X|^2\} = 0. \tag{A.12}$$

To satisfy the criterion for weak convergence, their distributions only need to resemble each other such that for all test functions $f : \mathbb{R} \to \mathbb{R}$ it follows that

$$\lim_{n\to\infty} \int_{-\infty}^{\infty} f(x) dF_{X_n}(x) = \int_{-\infty}^{\infty} f(x) dF_X(x). \tag{A.13}$$

A.2 Stochastic Processes

A **stochastic process** is a sequence of random variables at distinct time instants, $x = \{x(t), t \in T\}$ where $x : T \times \Omega \to \mathbb{R}$. A stochastic process is called a **Markov chain** when the future state depends only on the current state. In the discrete case this means that

$$P(x_{n+1}|x_n) = P(x_{n+1}|x_1, x_2, \ldots, x_n), \tag{A.14}$$

for all $x_1, x_2, \ldots, x_n \in \Omega$. Similarly, for a continuous-time random variable $x(t)$ it follows that

$$P(x(t_1)|x_{t_0}) = P(x(t_1)|x(t)), \tag{A.15}$$

for all $t \leq t_0 \leq t_1$.

A continuous-time stochastic process has **independent increments** if increments $x(t_{j+1}) - x(t_j)$ for j = 0,1,... n-1 of any $t_0 < t_1 < \ldots < t_n \in T$ are independent. The initial state $x(t_0)$ and the increments $x(t_j) - x(t_0)$ for j = 2,3, ...,n-1 are also independent. Examples of this are the Poisson and Wiener processes.

A **Wiener Process** $v = \{v(t), t \geq 0\}$ is defined such that for all $0 \leq s \leq t$ it follows that

$$v(0) = 0, \quad \mathrm{E}\{v(t)\} = 0, \quad \mathrm{var}\{v(t) - v(s)\} = t - s. \tag{A.16}$$

A **stationary process** in the stochastic sense is one whose joint distributions are all invariant under time displacements such that

$$F(t_1 + \delta, t_2 + \delta, \ldots, t_n + \delta) = F(t_1, t_2, \ldots, t_n). \tag{A.17}$$

A.2.1 Fubini's Theorem

It can be shown that for any stochastic process there is one that is jointly measurable which can replace it. Label this replacement as $x = \{x_t, t \in T\}$, then by Fubini's theorem [50], it follows that for $[t_o, t] \times \Omega \subseteq T \times \Omega$

$$\int_{t_o}^{t} E\{x_s\} ds = E\Big\{ \int_{t_o}^{t} x_s ds \Big\}. \tag{A.18}$$

This is an important result that implies that the integration and expectation operations can be interchanged.

A.3 *Itô* Calculus

An **ordinary differential equation** (ODE) is given by

$$dx(t) = f(x(t), t)dt, \tag{A.19}$$

where $f(x(t), t)$ is the drift function and is Lipschitz continuous and $x(t; x(t_o), t_o) = x(t; x(s; x(t_o), t_o), s)$. The latter is the deterministic version of the Markov property which states that future state variables are determined by its current state variables, and not its past state variables. The solution of the general ODE in (A.19) is discussed in depth in Chapter 3. The stochastic equivalent of (A.19), referred to as a **stochastic differential equation** (SDE), is written as

$$dx(t) = f(x(t), t)dt + b(x(t), t)\xi(t)dt, \tag{A.20}$$

where $b(x(t), t)$ is an appropriately scaled matrix (the diffusion coefficient) and $\xi(t)$ is the derivative of the Wiener process, $v(t)$, or $\xi(t) = \frac{dv(t)}{dt}$ [50]. It follows that

$$dx(t) = f(x(t), t)dt + b(x(t), t)dv(t), \tag{A.21}$$

with solution

$$x = x_o + \int_o^t f(x(s), s)ds + \int_o^t b(x(s), s)dv(s). \tag{A.22}$$

A Wiener process is, however, not differentiable. The second integral in (A.22) must therefore be evaluated as an *Itô* integral.

A.3.1 *Itô*'s Lemma/Formula

The process in (A.21) can be written in compact notation as

$$dx_t = fdt + bdv_t. \tag{A.23}$$

For a smooth, scalar-valued real function, $\phi(x(t), t)$, with continuous partial derivatives $\frac{\partial \phi}{\partial t}$, $\frac{\partial \phi}{\partial x}$ and $\frac{\partial^2 \phi}{\partial x^2}$, its Taylor expansion [7] is

$$d\phi = \frac{\partial \phi}{\partial t} dt + \left(\frac{\partial \phi}{\partial x}\right)^T dx + \frac{1}{2} dx^T \frac{\partial^2 \phi}{\partial x^2} dx + \dots, \tag{A.24}$$

where

$$\frac{\partial \phi}{\partial x} = \left[\frac{\partial \phi}{\partial x_1}, \dots, \frac{\partial \phi}{\partial x_n}\right]^T \qquad \text{and} \qquad \frac{\partial^2 \phi}{\partial x^2} = \begin{bmatrix} \frac{\partial \phi^2}{\partial^2 x_1} & \frac{\partial^2 \phi}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 \phi}{\partial x_1 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \phi^2}{\partial x_n x_1} & \frac{\partial^2 \phi}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 \phi}{\partial^2 x_n} \end{bmatrix}. \tag{A.25}$$

Substituting (A.23) into (A.24) gives *Itô*'s lemma:

$$\begin{aligned} d\phi &= \frac{\partial \phi}{\partial t} dt + \left(\frac{\partial \phi}{\partial x}\right)^T (f dt + b dv_t) + \frac{1}{2} (f dt + b dv_t)^T \frac{\partial^2 \phi}{\partial x^2} (f dt + b dv_t) + \dots \\ &= \left(\frac{\partial \phi}{\partial t} + \left(\frac{\partial \phi}{\partial x}\right)^T f + \frac{1}{2} \mathrm{tr}\left(b Q b^T\right) \frac{\partial^2 \phi}{\partial x^2}\right) dt + \left(\frac{\partial \phi}{\partial x}\right)^T b dv_t. \end{aligned} \tag{A.26}$$

The last step is obtained by retaining only second-order terms in dv_t and using its expected value as done in [50], i.e., $dt^2 = dt dv_t = 0$ and $\mathrm{E}\{dv_t dv_t\} = Q dt$.

A.3.2 *Itô*-Taylor Expansion

Integrating *Itô*'s lemma in (A.26) for $x_t \in \mathbb{R}$ where $\phi = U$, $Q = I = 1$ and the explicit time derivative term is neglected as done in [7], gives

$$\begin{aligned} U(x_t) &= U(x_{t_0}) + \int_{t_0}^{t} \left(f(x_s) U'(x_s) + \frac{1}{2} b^2(x_s) U''(x_s)\right) ds + \int_{t_0}^{t} b(x_s) U'(x_s) dv_s \\ &= U(x_{t_0}) + \int_{t_0}^{t} L^0 U(x_s) ds + \int_{t_0}^{t} L^1 U(x_s) dv_s, \end{aligned} \tag{A.27}$$

with operators

$$L^0 U = f U' + \frac{1}{2} b^2 U'', \tag{A.28}$$

$$L^1 U = b U'. \tag{A.29}$$

Consider the scalar *Itô* SDE in integral form

$$x_t = x_{t_0} + \int_{t_0}^{t} f(x_s, s) ds + \int_{t_0}^{t} b(x_s, s) dv_s. \tag{A.30}$$

Replacing $U = f$ and $U = b$ in (A.27) and substituting respectively into (A.30) gives the simplest *Itô*-Taylor expansion for x_t:

$$x_t = x_{to} + \int_{to}^{t} \left(f(x_{to}) + \int_{to}^{s} L^0 f(x_z) dz + \int_{to}^{s} L^1 f(x_z) dv_z \right) ds + \int_{to}^{t} \left(b(x_{to}) + \int_{to}^{s} L^0 b(x_z) dz + \int_{to}^{s} L^1 b(x_z) dv_z \right) dv_s. \tag{A.31}$$

B

Additional Proofs

B.1 Linearisation of the Embedded Reference Formula

$$
\begin{aligned}
x^*_{i+1} &= x_i + \delta\Big(\sum_{i=1}^{s} \vartheta^*_i k_i + \vartheta^*_\alpha f(x_i) + \gamma f(x^*_{i+1})\Big) \\
&= x_i + \delta\Big(\sum_{i=1}^{s} \vartheta^*_i k_i + \vartheta^*_\alpha f(x_i) + \gamma\big[f(x_i) + J(x_i)(x^*_{i+1} - x_i) + \dots\big]\Big) \\
&\approx x_i + \delta\Big(\sum_{i=1}^{s} \vartheta^*_i k_i + \vartheta^*_\alpha f(x_i) + \gamma f(x_i) + \gamma J(x_i)x^*_{i+1} - \gamma J(x_i)x_i\Big) \\
&\approx x_i + \delta\Big(\sum_{i=1}^{s} \vartheta^*_i k_i + \vartheta^*_\alpha f(x_i) + \gamma f(x_i)\Big) + \delta\gamma J(x_i)x^*_{i+1} - \delta\gamma J(x_i)x_i \\
\therefore x^*_{i+1} - \delta\gamma J(x_i)x^*_{i+1} &\approx x_i - \delta\gamma J(x_i)x_i + \delta\Big(\sum_{i=1}^{s} \vartheta^*_i k_i + \vartheta^*_\alpha f(x_i) + \gamma f(x_i)\Big) \\
\therefore \big(I - \delta\gamma J(x_i)\big)x^*_{i+1} &\approx \big(I - \delta\gamma J(x_i)\big)x_i + \delta\Big(\sum_{i=1}^{s} \vartheta^*_i k_i + \vartheta^*_\alpha f(x_i) + \gamma f(x_i)\Big) \\
\therefore x^*_{i+1} &\approx x_i + \delta\big(I - \delta\gamma J(x_i)\big)^{-1}\Big(\sum_{i=1}^{s} \vartheta^*_i k_i + \vartheta^*_\alpha f(x_i) + \gamma f(x_i)\Big) \qquad \text{(B.1)}
\end{aligned}
$$

B.2 Stability of the Resultant MGL Scheme

The numerical solution of the resultant MGL scheme in step (6) of Algorithm 1A has approximation step

$$P(t_i + \delta t) = P(t_i) + \Phi(P(t_i), t_i)\delta, \tag{B.2}$$

and increment function

$$\Phi(P(t_i), t_i) = \rho\Big(P(t_i) + \Phi(P(t_i), t_i)\frac{\delta}{2}, t_i\Big), \tag{B.3}$$

where $\rho(x(t), t)$ is the drift function of the state error covariance. The scheme's stability will be shown using the scalar test equation $\dot{y} = \eta y$, for $\eta \in \mathbb{C}^-$. See [133] for further details surrounding the scalar test equation. The increment function in (B.3) uses a gradient evaluated at time t_i but at an intermediate covariance situated between two successive covariances $P(t_i)$ and $P(t_{i+1})$. It is not obvious at what time this intermediate covariance is evaluated, call this covariance $P(t_i + a)$ where $a = \theta\delta$ and $0 < \theta < 1$. Taylor expanding $P(t_i + a)$ around t_i gives

$$P(t_i + a) = P(t_i) + \rho\big(P(t_i + a), t_i + a\big)P(t_i)a. \tag{B.4}$$

From the scalar test equation it follows that $\rho\big(P(t_i + a), t_i + a\big) = \eta P(t_i + a)$. Substituting this into (B.4) and re-arranging it follows that

$$P(t_i + a) = \frac{P(t_i)}{(1 - \eta a)}. \tag{B.5}$$

With reference to (B.3) write

$$P(t_i) + \Phi(P(t_i), t_i)\frac{\delta t}{2} = P(t_i + a). \tag{B.6}$$

Equating (B.5) and (B.6) gives an explicit expression for the increment function:

$$\Phi(P(t_i), t_i) = \frac{2P(t_i)}{\delta}\left(\frac{1}{1 - \eta a} - 1\right).$$

Substituting the above equation into (B.2), the approximation step becomes

$$P(t_i + \delta) = P(t_i)\left(\frac{1 + \eta\theta\delta}{1 - \eta\theta\delta}\right) \tag{B.7}$$

$$= P(t_i)\Theta(\eta\theta\delta), \tag{B.8}$$

where the stability function satisfies $|\Theta(\eta\theta\delta)| \leq 1$ for any $\eta \in \mathbb{C}^-$, implying A-stability (and not strict stability since $|\Theta(\infty)| = 1$). Note that the stability function Θ in (B.7) has a similar form to that of other stability functions of implicit methods such as the Taylor-Heun scheme in (4.38), which can be found in the various literature as $\Theta_{TH} = \frac{1+\frac{\eta\delta}{2}}{1-\frac{\eta\delta}{2}}$ [62].

Bibliography

[1] "Spiral wound universal pressure boiler." https://www.babcock.com/products/supercritical-boiler. Accessed: 2021-02-15.

[2] J. G. Collier and J. R. Thome, *Convective boiling and condensation*. Clarendon Press, 1994.

[3] Z. Patel and E. Boie, "Application of the continuous-discrete kalman filter to state estimation of the convection pass of a once-through power boiler," in *2019 American Control Conference (ACC)*, pp. 3167–3173, IEEE, 2019.

[4] Z. Patel and E. Boje, "A hybrid, coupled approach to the continuous-discrete kalman filter," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 827–832, 2020.

[5] H. W. Sorenson, "Least-squares estimation: from gauss to kalman," *IEEE spectrum*, vol. 7, no. 7, pp. 63–68, 1970.

[6] T. Kailath, "A view of three decades of linear filtering theory," *IEEE Transactions on information theory*, vol. 20, no. 2, pp. 146–181, 1974.

[7] A. H. Jazwinski, *Stochastic processes and filtering theory*. Courier Corporation, 2007.

[8] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068, pp. 182–194, International Society for Optics and Photonics, 1997.

[9] I. Arasaratnam and S. Haykin, "Cubature kalman filters," *IEEE Transactions on automatic control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[10] K. Ito, "Gaussian filter for nonlinear filtering problems," in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, vol. 2, pp. 1218–1223, IEEE, 2000.

[11] D. Alspach and H. Sorenson, "Nonlinear bayesian estimation using gaussian sum approximations," *IEEE transactions on automatic control*, vol. 17, no. 4, pp. 439–448, 1972.

[12] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEE Proceedings F (Radar and Signal Processing)*, vol. 140, pp. 107–113, IET, 1993.

[13] T. Bengtsson, P. Bickel, B. Li, *et al.*, "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems," in *Probability and Statistics: Essays in honor of David A. Freedman*, pp. 316–334, Institute of Mathematical Statistics, 2008.

[14] M. F. Bugallo and P. M. Djurić, "Gaussian particle filtering in high-dimensional systems," in *2014 IEEE Workshop on Statistical Signal Processing (SSP)*, pp. 129–132, IEEE, 2014.

[15] J. H. Kotecha and P. M. Djuric, "Gaussian sum particle filtering," *IEEE Transactions on signal processing*, vol. 51, no. 10, pp. 2602–2612, 2003.

[16] G. Evensen, "The ensemble kalman filter: Theoretical formulation and practical implementation," *Ocean dynamics*, vol. 53, no. 4, pp. 343–367, 2003.

[17] M. Roth, C. Fritsche, G. Hendeby, and F. Gustafison, "The ensemble kalman filter and its relations to other nonlinear filters," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, pp. 1236–1240, IEEE, 2015.

[18] S. Gillijns, O. B. Mendoza, J. Chandrasekar, B. De Moor, D. Bernstein, and A. Ridley, "What is the ensemble kalman filter and how well does it work?," in *2006 American Control Conference*, pp. 6–pp, IEEE, 2006.

[19] S. Sarkka, "On unscented kalman filtering for state estimation of continuous-time nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1631–1641, 2007.

[20] P. Frogerais, J.-J. Bellanger, and L. Senhadji, "Various ways to compute the continuous-discrete extended kalman filter," *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 1000–1004, 2011.

[21] R. Van Der Merwe and E. A. Wan, "The square-root unscented kalman filter for state and parameter-estimation," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, vol. 6, pp. 3461–3464, IEEE, 2001.

[22] B. Jia, M. Xin, and Y. Cheng, "Sparse gauss-hermite quadrature filter with application to spacecraft attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 2, pp. 367–379, 2011.

[23] B. Jia, M. Xin, and Y. Cheng, "High-degree cubature kalman filter," *Automatica*, vol. 49, no. 2, pp. 510–518, 2013.

[24] B. Jagan, S. Koteswara Rao, and K. Lakshmi, "Concert assessment of unscented and cubature kalman filters for target tracking," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 9, pp. 48–56, 05 2017.

[25] M. NøRgaard, N. K. Poulsen, and O. Ravn, "New developments in state estimation for nonlinear systems," *Automatica*, vol. 36, no. 11, pp. 1627–1638, 2000.

[26] M. Šimandl and J. Duník, "Derivative-free estimation methods: New results and performance analysis," *Automatica*, vol. 45, no. 7, pp. 1749–1757, 2009.

[27] A. Gelb, *Applied optimal estimation*. MIT press, 1974.

[28] S. Sarkka and J. Sarmavuori, "Gaussian filtering and smoothing for continuous-discrete dynamic systems," *Signal Processing*, vol. 93, no. 2, pp. 500–510, 2013.

[29] G. Y. Kulikov and M. V. Kulikova, "Accurate numerical implementation of the continuous-discrete extended kalman filter," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 273–279, 2013.

[30] G. Y. Kulikov and M. V. Kulikova, "The continuous–discrete extended kalman filter revisited," *Russian Journal of Numerical Analysis and Mathematical Modelling*, vol. 32, no. 1, pp. 27–38, 2017.

[31] T. Mazzoni, "Computational aspects of continuous–discrete extended kalman-filtering," *Computational Statistics*, vol. 23, no. 4, pp. 519–539, 2008.

[32] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving ordinary differential equations. 1, Nonstiff problems*. Springer-Vlg, 1991.

[33] E. Eitelberg and E. Boje, "Quasi steady state modelling of an evaporator," in *Proceedings of The First African Control Conference*, pp. 169–174, 2003.

[34] G. Y. Kulikov, "Embedded symmetric nested implicit runge–kutta methods of gauss and lobatto types for solving stiff ordinary differential equations and hamiltonian systems," *Computational Mathematics and Mathematical Physics*, vol. 55, no. 6, pp. 983–1003, 2015.

[35] G. Y. Kulikov and M. V. Kulikova, "Accurate continuous–discrete unscented kalman filtering for estimation of nonlinear continuous-time stochastic models in radar tracking," *Signal Processing*, vol. 139, pp. 25–35, 2017.

[36] M. Jiandong, W. Wei, and H. Wenbin, "Nonlinear numerical method for stiff systems," in *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, vol. 6, pp. V6–437, IEEE, 2010.

[37] L. M. Skvortsov, "Diagonally implicit runge-kutta methods for stiff problems," *Computational mathematics and mathematical physics*, vol. 46, no. 12, pp. 2110–2123, 2006.

[38] M. Liu, L. Zhang, and C. Zhang, "Study on banded implicit runge–kutta methods for solving stiff differential equations," *Mathematical Problems in Engineering*, vol. 2019, 2019.

[39] T. Knudsen and J. Leth, "A new continuous discrete unscented kalman filter," *IEEE Transactions on Automatic Control*, vol. 64, no. 5, pp. 2198–2205, 2018.

[40] S. Lv, J. Chen, and Z. Liu, "Udut continuous-discrete unscented kalman filtering," in *2008 Second International Symposium on Intelligent Information Technology Application*, vol. 2, pp. 876–879, IEEE, 2008.

[41] G. Y. Kulikov and M. Kulikova, "Accurate cubature and extended kalman filtering methods for estimating continuous-time nonlinear stochastic systems with discrete measurements," *Applied Numerical Mathematics*, vol. 111, pp. 260–275, 2017.

[42] J. Wang, D. Zhang, and X. Shao, "New version of continuous–discrete cubature kalman filtering for nonlinear continuous–discrete systems," *ISA Transactions*, vol. 91, pp. 174–183, 2019.

[43] G. Y. Kulikov and M. Kulikova, "Nirk-based cholesky-factorized square-root accurate continuous-discrete unscented kalman filters for state estimation in nonlinear continuous-time stochastic models with discrete measurements," *Applied Numerical Mathematics*, vol. 147, pp. 196–221, 2020.

[44] M. V. Kulikova and G. Y. Kulikov, "Svd-based factored-form cubature kalman filtering for continuous-time stochastic systems with discrete measurements," *Automatica*, vol. 120, p. 109110, 2020.

[45] I. Arasaratnam, S. Haykin, and T. R. Hurd, "Cubature kalman filtering for continuous-discrete systems: theory and simulations," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 4977–4993, 2010.

[46] G. Y. Kulikov and M. V. Kulikova, "Square-root accurate continuous-discrete extended-unscented kalman filtering methods with embedded orthogonal and j-orthogonal qr decompositions for estimation of nonlinear continuous-time stochastic models in radar tracking," *Signal Processing*, vol. 166, p. 107253, 2020.

[47] G. Y. Kulikov and M. V. Kulikova, "Accuracy issues in kalman filtering state estimation of stiff continuous-discrete stochastic models arisen in engineering research," in *2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 800–805, IEEE, 2018.

[48] S. Trimpe and R. D'Andrea, "Event-based state estimation with variance-based triggering," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3266–3281, 2014.

[49] Y. Niu, Y. Cong, and L. Wu, "Event-triggered continuous-discrete kalman filter with controllable estimation error," *IEEE Access*, vol. 6, pp. 42482–42496, 2018.

[50] P. E. Kloeden and E. Platen, *Numerical solution of stochastic differential equations*, vol. 23. Springer Science & Business Media, 2013.

[51] P. Deuflhard, E. Hairer, and J. Zugck, "One-step and extrapolation methods for differential-algebraic systems," *Numerische Mathematik*, vol. 51, no. 5, pp. 501–516, 1987.

[52] A. Cardone, Z. Jackiewicz, A. Sandu, and H. Zhang, "Extrapolated implicit–explicit runge–kutta methods," *Mathematical Modelling and Analysis*, vol. 19, no. 1, pp. 18–43, 2014.

[53] G. Y. Kulikov, "On implicit extrapolation methods for ordinary differential equations," *Russian Journal of Numerical Analysis and Mathematical Modelling*, vol. 17, no. 1, pp. 41–70, 2002.

[54] J. J. de Swart and G. Söderlind, "On the construction of error estimators for implicit runge-kutta methods," *Journal of Computational and Applied Mathematics*, vol. 86, no. 2, pp. 347–358, 1997.

[55] L. F. Shampine and L. S. Baca, "Error estimators for stiff differential equations," *Journal of computational and applied mathematics*, vol. 11, no. 2, pp. 197–207, 1984.

[56] L. F. Shampine and H. Watts, "Global error estimates for ordinary differential equations," *ACM Transactions on Mathematical Software (TOMS)*, vol. 2, no. 2, pp. 172–186, 1976.

[57] J. Lang and J. G. Verwer, "On global error estimation and control for initial value problems," *SIAM Journal on Scientific Computing*, vol. 29, no. 4, pp. 1460–1475, 2007.

[58] E. Harier and G. Wanner, *Solving ordinary differential equations ii: stiff and differential algebraic problems*. Springer Verlag., 1991.

[59] R. Alexander, "Diagonally implicit runge–kutta methods for stiff ode's," *SIAM Journal on Numerical Analysis*, vol. 14, no. 6, pp. 1006–1021, 1977.

[60] A. Granados, "Implicit runge-kutta algorithm using newton-raphson method," 1998.

[61] J. Caillaud and L. Padmanabhan, "An improved semi-implicit runge-kutta method for stiff systems," *The Chemical Engineering Journal*, vol. 2, no. 4, pp. 227–232, 1971.

[62] R. Piché and A. Ellman, "Numerical integration of fluid power circuit models using two-stage semi-implicit runge-kutta methods," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 208, no. 3, pp. 167–175, 1994.

[63] L. Shampine, "Local extrapolation in the solution of ordinary differential equations," *Mathematics of Computation*, vol. 27, no. 121, pp. 91–97, 1973.

[64] L. F. Shampine, "Error estimation and control for odes," *Journal of Scientific Computing*, vol. 25, no. 1-2, pp. 3–16, 2005.

[65] G. Y. Kulikov, "Cheap global error estimation in some runge–kutta pairs," *IMA Journal of Numerical Analysis*, vol. 33, no. 1, pp. 136–163, 2013.

[66] K. Atkinson, W. Han, and D. E. Stewart, *Numerical solution of ordinary differential equations*, vol. 108. John Wiley & Sons, 2011.

[67] E. Eitelberg, "Numerical simulation of stiff systems with a diagonal splitting method," *Mathematics and Computers in Simulation*, vol. 21, no. 1, pp. 109–115, 1979.

[68] F. Gustafsson and G. Hendeby, "Some relations between extended and unscented kalman filters," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 545–555, 2011.

[69] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[70] S. G. Mohinder and P. A. Angus, "Kalman filtering: theory and practice using matlab," *John Wileys and Sons*, 2001.

[71] G. Y. Kulikov and M. V. Kulikova, "The accurate continuous-discrete extended kalman filter for radar tracking," *IEEE Transactions on Signal Processing*, vol. 64, no. 4, pp. 948–958, 2015.

[72] J. Sarmavuori and S. Sarkka, "Fourier-hermite kalman filter," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1511–1515, 2011.

[73] P. R. Conrad, M. Girolami, S. Särkkä, A. Stuart, and K. Zygalakis, "Statistical analysis of differential equations: introducing probability measures on numerical solutions," *Statistics and Computing*, vol. 27, no. 4, pp. 1065–1082, 2017.

[74] H. C. Lie, A. M. Stuart, and T. J. Sullivan, "Strong convergence rates of probabilistic integrators for ordinary differential equations," *Statistics and Computing*, vol. 29, no. 6, pp. 1265–1283, 2019.

[75] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of 1995 American Control Conference-ACC'95*, vol. 3, pp. 1628–1632, IEEE, 1995.

[76] Y. Ye, S. F. Koch, and J. Zhang, "Determinants of household electricity consumption in south africa," *Energy economics*, vol. 75, pp. 120–133, 2018.

[77] G. Mantashe, "Integrated resource plan (irp2019)," tech. rep., Department of Mineral Resources and Energy, 2019.

[78] K. Ratshomo and R. Nembahe, "The south african energy sector report," tech. rep., Department of Energy, 2019.

[79] A. T. Nair, T. Radhakrishnan, K. Srinivasan, and S. R. Valsalam, "Kalman filter based state estimation of a thermal power plant," in *2011 International Conference on Process Automation, Control and Computing*, pp. 1–5, IEEE, 2011.

[80] J. Wallace and R. Clarke, "The application of kalman filtering estimation techniques in power station control systems," *IEEE Transactions on Automatic Control*, vol. 28, no. 3, pp. 416–427, 1983.

[81] Y. Sato, M. Nomura, H. Matsumoto, and M. Iioka, "Steam temperature prediction control for thermal power plant," *IEEE transactions on power apparatus and systems*, no. 9, pp. 2382–2387, 1984.

[82] G. Jonsson, O. Palsson, and K. Sejling, "Modeling and parameter estimation of heat exchangers—a statistical approach," 1992.

[83] G. R. Jonsson, S. Lalot, O. P. Palsson, and B. Desmet, "Use of extended kalman filtering in detecting fouling in heat exchangers," *International journal of heat and mass transfer*, vol. 50, no. 13-14, pp. 2643–2655, 2007.

[84] S. A. Kumari and S. Srinivasan, "Optimal estimation based tube leak detection in a thermal power plant reheater," in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, pp. 2694–2697, IEEE, 2017.

[85] T. D. Liacco, "The role of state estimation in power system operation," *IFAC Proceedings Volumes*, vol. 15, no. 4, pp. 1531–1533, 1982.

[86] A. Abur and A. G. Exposito, *Power system state estimation: theory and implementation*. CRC press, 2004.

[87] S. R. Valsalam, S. Anish, and B. R. Singh, "Boiler modelling and optimal control of steam temperature in power plants," *IFAC Proceedings Volumes*, vol. 42, no. 9, pp. 125–130, 2009.

[88] T. Matona, "Power system status update," *Presentation, Eskom*, 2014.

[89] R. T. J. Sibanda, *Reliability improvement of the boiler-coal processing plant in eskom using reliability centred maintenance principles*. PhD thesis, 2016.

[90] H. Archary, W. Schmitz, and L. Jestin, "Mass flow and particle size monitoring of pulverised fuel vertical spindle mills," *Chemical and Process Engineering*, vol. 37, no. 2, pp. 175–197, 2016.

[91] A. Govender, *Development of a condition monitoring philosophy for a pulverised fuel vertical spindle mill*. PhD thesis, 2016.

[92] S. Yang, "A condition-based preventive maintenance arrangement for thermal power plants," *Electric Power Systems Research*, vol. 72, no. 1, pp. 49–62, 2004.

[93] A. Alessandri and T. Parisini, "Model-based fault diagnosis using nonlinear estimators: A neural approach," in *Proceedings of the 1997 American Control Conference (Cat. No. 97CH36041)*, vol. 2, pp. 903–907, IEEE, 1997.

[94] A. Alessandri, T. Parisini, and R. Zoppoli, "Sliding–window neural state estimation in a power plant heater line," *International Journal of Adaptive Control and Signal Processing*, vol. 15, no. 8, pp. 815–836, 2001.

[95] S. Kannaiyan and A. A. Bhadke, "Soft computing for industrial heat exchanger," in *2019 Fifth Indian Control Conference (ICC)*, pp. 34–38, IEEE, 2019.

[96] Z. Ge, "Review on data-driven modeling and monitoring for plant-wide industrial processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 171, pp. 16–25, 2017.

[97] S. J. Mills, "Integrating intermittent renewable energy technologies with coal-fired power plant," *CCC/189. London: IEA Clean Coal Centre*, 2011.

[98] Z. Tian, L. Xu, J. Yuan, X. Zhang, and J. Wang, "Online performance monitoring platform based on the whole process models of subcritical coal-fired power plants," *Applied Thermal Engineering*, vol. 124, pp. 1368–1381, 2017.

[99] P. Duda, "Development and application of the power plant real-time temperature and stress monitoring system," *Science and Technology of Nuclear Installations*, vol. 2017, 2017.

[100] B. Jouan, S. Bergholz, and J. Rudolph, "Fatigue monitoring approaches for power plants," in *ASME 2014 Pressure Vessels and Piping Conference*, American Society of Mechanical Engineers Digital Collection, 2014.

[101] X. Chen and J. Wang, "A soft-sensing model for oxygen-content in flue gases of coal-fired power plant based on neural network," in *2018 37th Chinese Control Conference (CCC)*, pp. 3657–3661, IEEE, 2018.

[102] G. Fan and N. Rees, "An intelligent expert system (kboss) for power plant coal mill supervision and control," *Control Engineering Practice*, vol. 5, no. 1, pp. 101–108, 1997.

[103] J. Waddington and G. Maples, "The control of large coal-and oil-fired generating units," *Power Engineering Journal*, vol. 1, no. 1, pp. 25–36, 1987.

[104] M. Nomura, Y. Sato, K. Ouchi, and R. Kondo, "Steam temperature prediction control method using prediction model of multi heat exchangers for thermal power plants," *IEEJ Transactions on Electronics, Information and Systems*, vol. 127, no. 9, pp. 1408–1418, 2007.

[105] G. Heo, S. S. Choi, and S. H. Chang, "Thermal power estimation by fouling phenomena compensation using wavelet and principal component analysis," *Nuclear engineering and design*, vol. 199, no. 1-2, pp. 31–40, 2000.

[106] D. Flynn, *Thermal power plant simulation and control*. No. 43, IET, 2003.

[107] J. Mathew, S. Shankar, H. Pratheesh, C. Lajitha, *et al.*, "Implementation of high availability scada system for superheater steam temperature control in a 210 mw thermal power plant," in *2014 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pp. 1–6, IEEE, 2014.

[108] C. Tye and W. Hill, "Modeling and estimation applied to an AGR once-through boiler," 1983.

[109] K. Lo and Y. Rathamarit, "State estimation of a boiler model using the unscented kalman filter," *IET Generation, Transmission and Distribution*, vol. 2, no. 6, pp. 917–931, 2008.

[110] E. Aranda, M. Frye, and C. Qian, "Model development, state estimation, and controller design of a nonlinear utility boiler system," in *2008 IEEE International Conference on Industrial Technology*, pp. 1–6, IEEE, 2008.

[111] C. J. Backi, "Enthalpy control for the feed water supply in a once-through boiler with disturbance estimation," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1991–1996, 2017.

[112] T. Herlambang, Z. Mufarrikoh, D. Karya, and D. Rahmalia, "Estimation of water level and steam temperature using ensemble kalman filter square root (enkf-sr)," in *Journal of Physics: Conference Series*, vol. 1008, p. 012026, IOP Publishing, 2018.

[113] J. Opalka and L. Hubka, "Nonlinear state and unmeasured disturbance estimation for use in power plant superheaters control," *Procedia Engineering*, vol. 100, pp. 1539–1546, 2015.

[114] P. F. Odgaard and B. Mataji, "Observer-based fault detection and moisture estimating in coal mills," *Control Engineering Practice*, vol. 16, no. 8, pp. 909–921, 2008.

[115] G. Fan and N. Rees, "Furnace flue gas modelling and simulation," *IFAC Proceedings Volumes*, vol. 29, no. 1, pp. 6885–6890, 1996.

[116] P. Andersen, J. D. Bendtsen, J. H. Mortensen, R. Nielsen, and T. Pedersen, *Observer-based Fuel Control Using Oxygen Measurement: A Study Based on a First-principles Model of a Pulverized Coal Fired Benson Boiler*. Värmeforsk, 2005.

[117] L. Magni, J. Paderno, and F. Pretolani, "Kalman filter estimation of the coal flow in power plants," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 1270–1275, IEEE, 2005.

[118] F. P. Incropera, D. P. Dewitt, T. Bergman, and A. Lavine, "Fundamentals of mass and heat transfer," *Australia: John Willey and Sons*, 2002.

[119] "Roughness and surface coefficients." https://www.engineeringtoolbox.com/surface-roughness-ventilation-ducts-d_209.html. Accessed: 2018-03-01.

[120] W. H. Jens and P. Lottes, "Analysis of heat transfer, burnout, pressure drop and density date for high-pressure water," tech. rep., Argonne National Lab., 1951.

[121] G. Lausterer, J. Franke, and E. Eitelberg, "Mathematical modelling of a steam generator," in *Digital Computer Applications to Process Control*, pp. 411–417, Elsevier, 1980.

[122] T. L. Bergman, F. P. Incropera, D. P. DeWitt, and A. S. Lavine, *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[123] A. Rossouw, "Boiler System Modelling Using Flownex," Master's thesis, University of Cape Town, Cape Town, 2016.

[124] E. L. Keating, *Applied combustion*. CRC press, 2007.

[125] N. W. Eichhorn, *Combustion modelling of pulverised coal boiler furnaces fuelled with Eskom coals*. PhD thesis, 1998.

[126] J. Jones, P. Mason, and A. Williams, "A compilation of data on the radiant emissivity of some materials at high temperatures," *Journal of the Energy Institute*, vol. 92, no. 3, pp. 523–534, 2019.

[127] E. Boje and E. Eitelberg, "Augmented kalman filtering for a superheated steam header system," *IEEE transactions on control systems technology*, vol. 11, no. 5, pp. 773–781, 2003.

[128] K. Ehinger, D. Flach, L. G. E. Horlebein, D. R. H. H. Ilgner, T. K. H. Müller, and H. S. A. Schüssler, "Industrial temperature measurement. basics and practice," *ABB Automation Products GmbH*, 2013.

[129] G. Lausterer, J. Franke, and E. Eitelberg, "Modular modelling applied to a benson® boiler," in *Modelling and Control of Electric Power Plants*, pp. 11–19, Elsevier, 1984.

[130] Y. Cai, K. Tay, Z. Zheng, W. Yang, H. Wang, G. Zeng, Z. Li, S. K. Boon, and P. Subbaiah, "Modeling of ash formation and deposition processes in coal and biomass fired boilers: A comprehensive review," *Applied energy*, vol. 230, pp. 1447–1544, 2018.

[131] W. Wei, F. Sun, Y. Shi, and L. Ma, "Experimental research of fouling layer and prediction of acid condensation outside heat exchanger used in coal-fired boiler," *Applied Thermal Engineering*, vol. 131, pp. 486–496, 2018.

[132] R. Korbee, M. Losurdo, J. Lensselink, M. Cieplik, and F. Verhoeff, "Monitoring and modelling of gas-side boiler fouling," 2005.

[133] G. Wanner, E. Hairer, and S. P. Nørsett, "Order stars and stability theorems," *BIT Numerical Mathematics*, vol. 18, no. 4, pp. 475–489, 1978.