Candidate Number: 2204913

Project title: Signal Processing of 3D and 4D processes

Date: 8th May 2022

Department of Electronic Engineering
Royal Holloway, University of London

BEng Computer Systems Engineering Third-Year Individual Project

Abstract

The dimensionality of a signal is either defined as the number of independent variables a signal depends on, or the number of dimensions of a vector form signal, the latter definition will be used, and signal processing solutions, specifically the Least Mean Squares adaptive filter and its multichannel variants will be employed for time series prediction of real-world multi-channel weather and financial data. The algorithms (LMS, QLMS and Multi-Channel LMS) will be analysed and investigated in their prediction accuracy, and their properties explored.

Table of Contents

1. Introduction

Time series forecasting or prediction is the process of making predictions using a historical time stamped sequence of data points collected over intervals of time and using these past and current samples to predict future samples. It comes down to the question of causality and how the past influences the future. Time series forecasting is widely used in various fields such as finance and economics to predict economic growth [1] and in stock market prediction [2] to trade markets, in medicine in analysis of electrocardiograms [3] and electroencephalograms [4], in astronomy for analysis of stars [5] and other applications. Time series prediction can be conducted in various ways such as using statistical techniques which include AR (autoregressive), MA (moving-average) and ARMA (autoregressive moving-average) models [6]. Other ways include machine learning methods which include multi-layer perceptrons [7], recurrent neural networks [8], and long-short term memory cells [9]. Time series data is equivalent to a discrete-time signal, and so signal processing methods for prediction can be applied. In signal processing, linear prediction is the operation where future values are predicted from current and previous signal samples, this has a variety of applications the major one being linear predictive coding which is used in audio and speech signal processing [10].

Adaptive filters have a wide range of uses such as channel equalisation [11], system identification [12], noise cancellation [13], acoustic echo cancellation [14] and others as well as linear prediction [15]. Adaptive filters have been applied for use in time series prediction of financial [16] and other data [17].

In this project multichannel (higher dimensional) signals of wind and stock data will be investigated. The LMS Adaptive filter and its multichannel variants will be modified for time-series prediction and investigated, and each algorithm will be explored and compared. The principal question is whether the multichannel algorithms, the Quaternion LMS and Multi-Channel LMS, give better prediction accuracy than single channel LMS. The parameters of each algorithm will also be altered and explored to assess relationships and behaviour.

2. Background and Literature Review

2.0 Signal Dimensionality and Multichannel Signals

A D-dimensional signal can have various definitions. It can refer to a signal which has d independent variables such as a 2D image signal $f(x, y)$ which depends on 2 variables x and y. Or it may refer to a D-channelled multichannel signal, $\boldsymbol{x}(n) = [x_0(n), x_1(n), \dots, x_D(n)]^T$. In this paper the latter definition will be used as we will be analysing multichannel signals and signal processing techniques.

2.1 Linear Prediction

Linear Prediction is the mathematical process in which future values of a discrete-time signal are estimated as a linear combination of its prior samples.

$$\hat{x}(n) = \sum_{k=1}^{p} w_k x(n-k)$$

$\hat{x}(n)$ is the predicted value, with p prediction weights w_k multiplied by previous samples $w_k x(n-k)$.
The error is defined as $e(n) = x(n) - \hat{x}(n)$. There are various methods to optimise the parameters w_k, such as minimising the expectation of the mean squared error.

Linear Prediction is widely used in time series analysis such as stock market prediction,[2] its used in pulse code modulation and linear predictive coding, in fields of audio signal and speech processing [10] to compress the spectral envelope information of a signal, allowing good quality speech to be encoded at a low bit rate.

2.2 Adaptive Filters

2.2.1 Introduction

An adaptive filter is a digital filter whose transfer function has variable coefficients, which are updated according to an adaptive algorithm, the most common being the LMS and RLS algorithms [18]. The adaptive filter process involves the optimisation of a cost or loss function, the adaptive algorithm updates the filter coefficients in a way that minimizes this cost function.

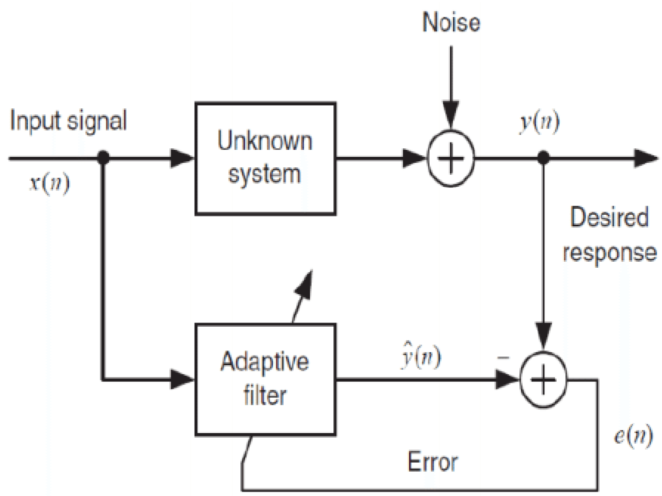


Figure 1 Adaptive Filter Block Diagram from [19]

Adaptive filters have a multitude of applications, one being system identification [12] shown in Figure 1, in which an unknown system is modelled, the adaptive filter, by minimizing the cost function will eventually come to resemble or very closely approximate the behaviour of the unknown system.

Another use of an adaptive filter is in noise cancellation [13], whereby background noise is removed from an input signal. This has wide applications in mobile phones, radio communication, active noise control in headphones and others. The process is shown in figure 2 where you have an input signal of $d(n) = s(n) + x1(n)$, the pure signal plus noise, you then record background noise $x2(n)$ that you assume to be correlated somehow with the noise $x1(n)$. The $x2(n)$ noise goes through the adaptive filter and the filter output $y(n)$ is subtracted from $d(n)$ which gives you $e(n)$. A cost function of $e(n)$ is optimized by the adaptive filter algorithm, by minimizing the error the adaptive filter produces its best estimate of $x1(n)$ from $x2(n)$, $y(n)$, giving you the noise cancelled signal $e(n) = s(n) + x1(n) – y(n) = \hat{s}(n)$. The signal noise is effectively reduced.

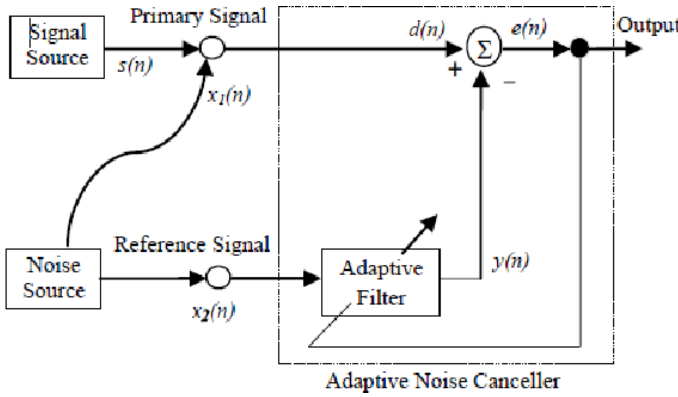


Figure 2 Adaptive Noise Canceller edited from [20]

Adaptive filters are also used to predict future samples by linear prediction. In this case the desired signal is an advanced version of the input signal, and by minimising the error between the estimated and desired signal, the adaptive filter weights converge to give the best prediction of the advanced input signal. This filter design is shown in Figure 3. Adaptive filters are used in linear predictive coding of speech [10] and other data signals, adaptive filter predictors are also used in adaptive line enhancement which is a similar process to noise cancellation. In adaptive line enhancement the input signal is both noise and a sinusoidal component, the objective is to remove this noisy component. The desired signal is an advanced version of the input signal in order to decorrelate the noise components of the signals, this way the adaptive predictor can only predict the sinusoidal component. Adaptive line enhancement has many applications from biomedical applications such as an electronic stethoscope [21] to helicopter gearbox fault diagnosis [22]. Figure 3 below shows adaptive filter prediction, where the delayed version of the input signal is sent to the adaptive filter, which is equivalent to the prior explanation.

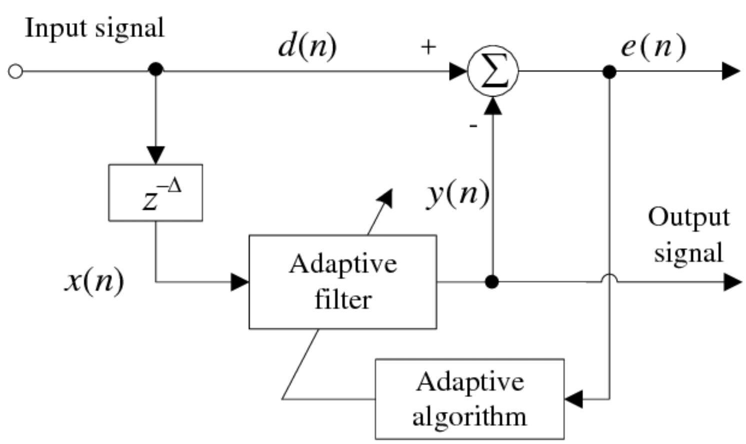


Figure 3 Adaptive filter linear prediction from [22]

Adaptive filters have myriad of other applications such as inverse modelling, which is used in channel equalisation [11] to reduce distortion from high-speed transmission on telephone channels, adaptive notch bandstop filters, echo cancellation [14], from hearing aids to foetal ECG monitoring [23] and removing ocular signal artefacts from EEG brain scans [24].

2.2.2 Adaptive Filters and Artificial Neurons

Artificial neurons or perceptrons are elementary units of artificial neural networks. They are an envisioned model of biological neurons. An artificial neuron receives 1 or multiple inputs, each input is weighted individually and is summed along with a bias and the sum is passed through an activation function.

$$y = \varphi\left(\sum_{i=1}^{m} w_i x_i + b\right)$$

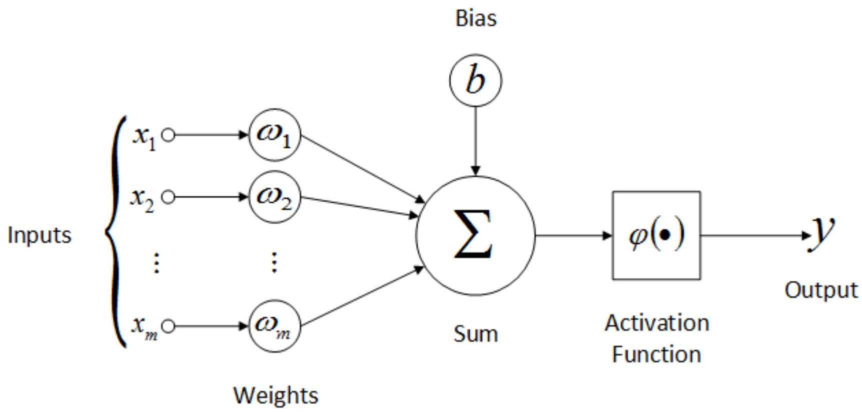


Figure 4 Diagram of artificial neuron

A neuron's weights are adjusted to optimise a cost function, normally being sum function of the error which is the difference between the desired and neuron output. In an artificial neural network backpropagation is the method to adjust connection weights to compensate for this error, the weight updates for a single neuron are often done via stochastic gradient descent. There are various activation functions whose selection is dependent on your network architecture.

An adaptive filter has p inputs, $\boldsymbol{x}(n) = [x(n), x(n-1), \dots, x(n-p+1)]^T$ and p weights, $\boldsymbol{w}(n) = \left[w_0(n), w_1(n), \dots, w_{p-1}(n)\right]^T$. The adaptive filter output is $y(n) = \boldsymbol{x}^{\boldsymbol{T}}(n)\boldsymbol{w}(n)$. In the Least Means Squares algorithm the mean square error cost function is minimised by stochastic gradient descent.

It can be concluded that a single simple artificial neuron with a constant activation function of unity and a bias of zero is essentially equivalent to an adaptive filter, with the inputs being the current sample of the signal and the $p-1$ previous samples.

Adaptive filters may employ more complicated sequential learning algorithms including that of a neural network [25], adaptive filters may also themselves be employed in neural network architectures, such as in natural language processing using convolutional neural networks [26] as well as for deep transfer learning [27].

2.3 Least Mean Squares Filter and Algorithms

2.3.1 Least Mean Squares Algorithm

Definitions

The Least Mean squares filter is a class of adaptive filter whose transfer function coefficients are obtained via minimising the cost function which is by definition the mean square of the error signal. It employs stochastic gradient descent on the error only at the current time.

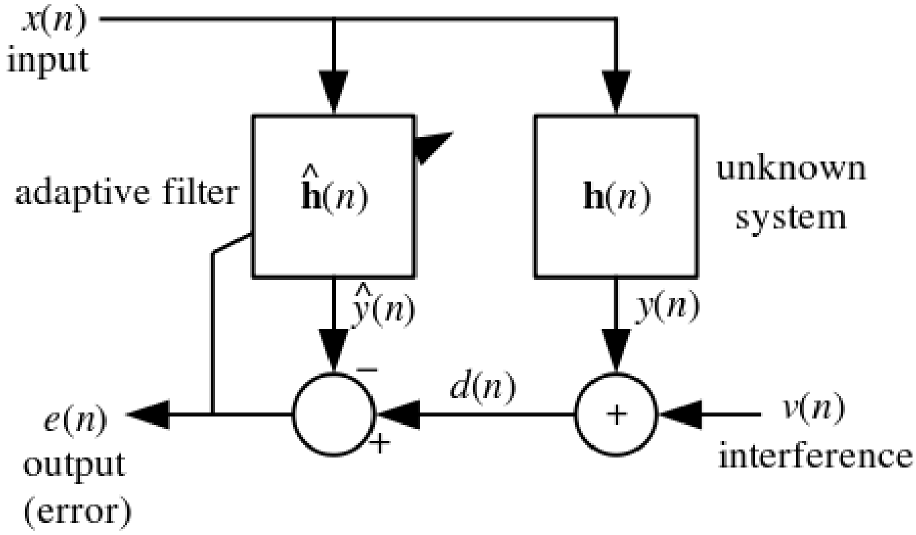


Figure 5 Diagram of Least Mean Square adaptive filter for system identification from [28]

The unknown filter has p inputs including the current input and up to $p-1$ delayed inputs, $\boldsymbol{x}(n) = [x(n), x(n-1), \dots, x(n-p+1)]^T$ and p filter coefficients, $\boldsymbol{h}(n) = \left[h_0(n), h_1(n), \dots, h_{p-1}(n)\right]^T$. The unknown filter output is $y(n) = \boldsymbol{h}^{\boldsymbol{H}}(n)\boldsymbol{x}(n)$. Where $\{\cdot\}^H$ is the Hermitian transpose. The adaptive filter has p filter coefficients, $\widehat{\boldsymbol{h}}(n) = \left[\hat{h}_0(n), \hat{h}_1(n), \dots, \hat{h}_{p-1}(n)\right]^T$. The adaptive filter output is $\hat{y}(n) = \widehat{\boldsymbol{h}}^{\boldsymbol{T}}(n)\boldsymbol{x}(n)$. Desired output is the unknown filter output plus noise, $d(n) = y(n) + v(n)$. The error signal is defined as $e(n) = d(n) - \hat{y}(n)$. The mean squared error is minimised by updating filter weights on each sample of the input signal using stochastic gradient descent. In the system identification case as shown in figure X the adaptive filter's estimated weights $\widehat{\boldsymbol{h}}(n)$ will converge to the unknown system weights $\boldsymbol{h}(n)$.

Derivation

The LMS Algorithm employs stochastic gradient descent to obtain optimum filter weights for $\widehat{\boldsymbol{h}}(n)$ by minimising the cost function defined as the mean squared error. $E\{\cdot\}$ denotes expectation or mean.

$$C(n) = E\{|e(n)|^2\}$$

This cost function is minimised by applying steepest descent, to do this we take the gradient of the cost function. We define the gradient operator as the vector of partial derivatives with respect to each of the filter weights.

$$\nabla_{\hat{\boldsymbol{h}}} = \begin{bmatrix} \frac{\partial}{\partial \hat{h}_0} \\ \frac{\partial}{\partial \hat{h}_1} \\ \dots \\ \dots \\ \dots \\ \frac{\partial}{\partial \hat{h}_{p-1}} \end{bmatrix}$$

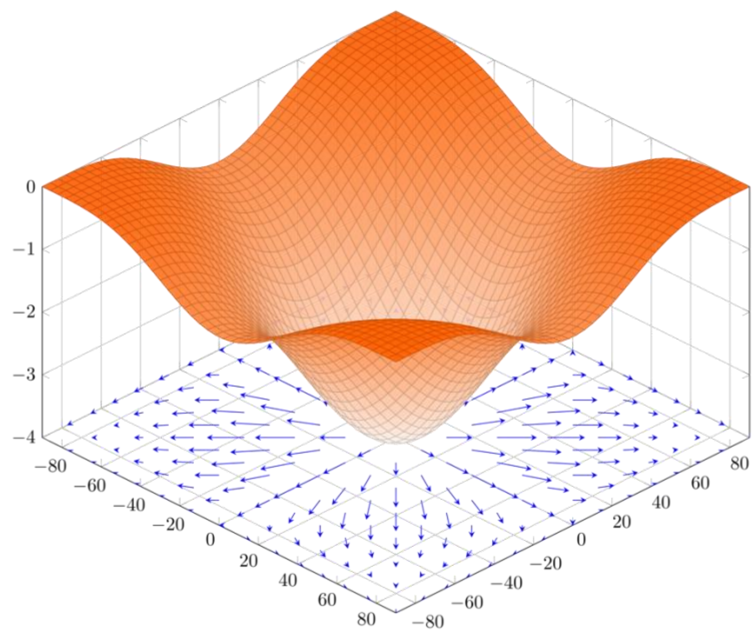


Figure 6 2 variable function with gradient vector field projected onto bottom plane from [29]

We apply the gradient operator to the cost function by taking its Hermitian transpose and multiplying it by the cost function.

$$\nabla_{\hat{\boldsymbol{h}}^H} C(n) = \nabla_{\hat{\boldsymbol{h}}^H} E\{|e(n)|^2\} = \nabla_{\hat{\boldsymbol{h}}^H} E\{e(n)e^*(n)\}$$

Via chain rule we obtain the given equation below.

$$\nabla_{\hat{\boldsymbol{h}}^H} E\{e(n)e^*(n)\} = 2E\{\nabla_{\hat{\boldsymbol{h}}^H}\big(e(n)\big)e^*(n)\}$$

$$\nabla_{\hat{\boldsymbol{h}}^H}\big(e(n)\big) = \nabla_{\hat{\boldsymbol{h}}^H}\left(d(n) - \hat{\boldsymbol{h}}^H \cdot \boldsymbol{x}(n)\right) = -\boldsymbol{x}(n)$$

The gradient and steepest ascent is given below.

$$\nabla_{\widehat{\boldsymbol{h}}^H} C(n) = -2E\{\boldsymbol{x}(n)e^*(n)\}$$

Consider Figure X, the height of the surface can be regarded as the value of the cost function, the mean squared error, and the x-y plane is in this case the 2 filter weights. The gradient vector field is shown on the bottom plane, showing us the direction of the steepest ascent at each point, given by $\nabla C(n)$. To minimise the cost function we will take a step in the opposite direction so will negate the gradient vector. The update algorithm is then as follows, where $\frac{\mu}{2}$ is the step constant.

$$\widehat{\boldsymbol{h}}(n+1) = \widehat{\boldsymbol{h}}(n) - \frac{\mu}{2}\nabla C(n) = \widehat{\boldsymbol{h}}(n) + \mu E\{\boldsymbol{x}(n)e^*(n)\}$$

The expectation is not computed as the LMS is run online, an instantaneous estimate is used so the update algorithm finally becomes as below.

$$\widehat{\boldsymbol{h}}(n+1) = \widehat{\boldsymbol{h}}(n) + \mu\boldsymbol{x}(n)e^*(n)$$

Convergence

As the LMS algorithm is online the weights never reach their exact optimal weights however convergence in mean is possible whereby the weights fluctuate by small amounts around the optimal. The step parameter μ is important so that the convergence is meaningful and that these fluctuations are minimal as possible, while also reaching, in the mean, these optimal weights as quickly as possible, so the step size must not be too large as to cause large deviations around the optimal and not too small as to be very slow to converge to the optimal weights mean. A step constant μ for convergence is loosely bound $0 < \mu < \frac{2}{LP_x}$, where L is the number of filter taps, and P_x is the power of the input signal.

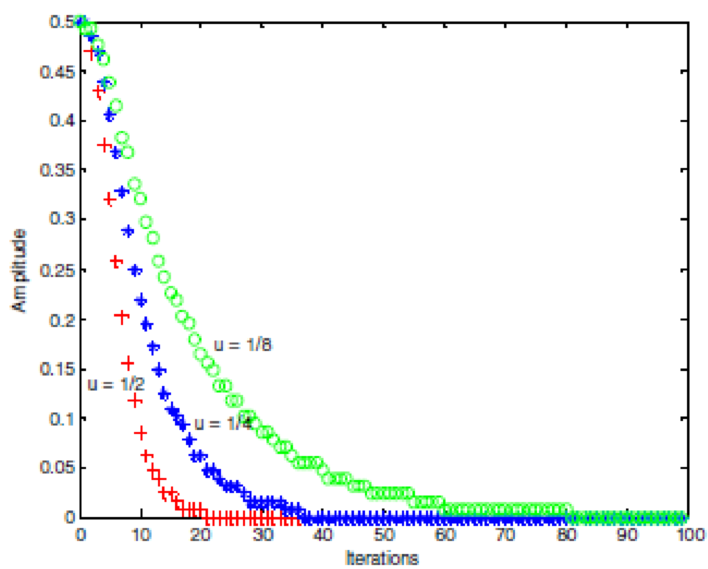


Figure 7 Absolute Error vs LMS iterations for 3 different step parameters from [30]

2.3.2 Multi-Channel Adaptive Filters

We consider a multiple input, multiple output (MIMO) system where there are P inputs and Q outputs. Each of the P inputs go through Q filters, there are PQ filters, each with a tap length of L, and each of the Q outputs sum the contributions from their respective filters. This is shown in Figure 8 below.

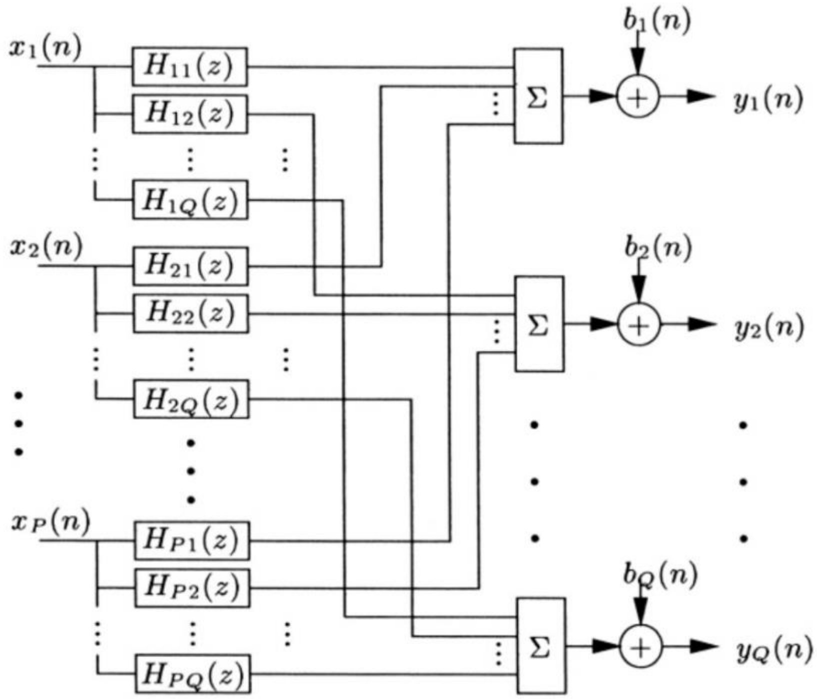


Figure 8 from [31]

The pth input is defined as $\boldsymbol{x}_p(n) = [x_p(n), x_p(n-1), \dots, x_p(n-L+1)]^T$, where $p = 1,2,\dots,P$. Each of the PQ filters are described as follows $\boldsymbol{h}_{pq}(n) = [h_{pq,0}(n), h_{pq,1}(n), \dots, h_{pq,L-1}(n)]^T$. The qth output is defined below as the sum of the P filter outputs plus noise.

$$y_q(n) = \sum_{p=1}^{P} \boldsymbol{h}_{pq}^T(\boldsymbol{n})\, \boldsymbol{x}_p(n) + b_q(n)$$

System identification is conducted on this MIMO system using a multichannel adaptive filter. We define the estimated adaptive filter weights as $\hat{\boldsymbol{h}}_{pq}(n) = [\hat{h}_{pq,0}(n), \hat{h}_{pq,1}(n), \dots, \hat{h}_{pq,L-1}(n)]^T$.

The estimated output and the qth error signal of the adaptive filter is shown below.

$$\hat{y}_q(n) = \sum_{p=1}^{P} \hat{\boldsymbol{h}}_{pq}^T(\boldsymbol{n})\, \boldsymbol{x}_p(n)$$

$$e_q(n) = y_q(n) - \hat{y}_q(n) = y_q(n) - \sum_{p=1}^{P} \hat{\boldsymbol{h}}_{pq}^T(\boldsymbol{n})\, \boldsymbol{x}_p(n)$$

Multi-channel adaptive filters have a wide range of applications, the foremost being multi-channel acoustic echo cancellation [citation] (MCAEC). In echo cancellation [31], an originally transmitted signal that reappears with some delay in the transmitted or received signal is recognised and is removed by subtracting it from the transmitted or received signal, for example voice in a telephone circuit voice output from the speaker, which is echoed back into the telephone microphone, this echo path would be estimated and cut out. In multi-channel acoustic echo cancellation there are multiple inputs and multiple outputs, for example a conference call, with P microphones and Q speakers, there would be PQ echo paths which need to be estimated. Other applications of multi-channel adaptive filtering include, in medicine, the removal of CPR artefacts from ECG signals [32].

2.3.3 Multi-Channel LMS

We define the qth error signal and cost function as below.

$$e_q(n) = y_q(n) - \sum_{p=1}^{P} \hat{\boldsymbol{h}}_{pq}^T(\boldsymbol{n})\, \boldsymbol{x}_p(n)$$

$$C_q(n) = E\left\{\left(y_q(n) - \sum_{p=1}^{P} \hat{\boldsymbol{h}}_{pq}^T(\boldsymbol{n})\, \boldsymbol{x}_p(n)\right)^2\right\}$$

Minimising the mean squared error cost function using stochastic gradient descent with instantaneous expectation estimate gives the update algorithm as below.

$$\hat{\boldsymbol{h}}_{pq}(n+1) = \hat{\boldsymbol{h}}_{pq}(n) + \mu \boldsymbol{x}_{\boldsymbol{p}}(n) e_q(n)$$

2.3.4 Quaternions

The quaternion number system is an extension of the complex numbers. Quaternions have various applications to a broad range of fields from robotics [citation] to computer graphics [citation]. Quaternions are represented in the form below.

$$q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}, q \in \mathbb{H}, a, b, c, d \in \mathbb{R}$$

$\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are the basic quaternions and follow the multiplication rule $\boldsymbol{i^2} = \boldsymbol{j^2} = \boldsymbol{k^2} = \boldsymbol{ijk} = \boldsymbol{-1}$.

The quaternion conjugate is given by $q^* = a - b\boldsymbol{i} - c\boldsymbol{j} - d\boldsymbol{k}$.

An involution is a self-inverse mapping, along with the quaternion conjugate there are three orthogonal quaternion involutions.

$$q^i = a + b\boldsymbol{i} - c\boldsymbol{j} - d\boldsymbol{k}$$
$$q^j = a - b\boldsymbol{i} + c\boldsymbol{j} - d\boldsymbol{k}$$
$$q^k = a - b\boldsymbol{i} - c\boldsymbol{j} + d\boldsymbol{k}$$

The relationship between the conjugate and involutions are as follows.

$$q^* = \frac{1}{2}\left(q^i + q^j + q^k - q\right)$$

$\mathbb{HR}$-Calculus provides a framework for differentiation of functions of quaternion variables. The i-gradient [33] provides a quaternion gradient based on involutions. These are used in the derivation of the QLMS.

2.3.5 QLMS

A 4-channel signal can be represented as a quaternion, shown below (a bold function represents a quaternion).

$$\boldsymbol{x}(n) = x_a(n) + x_b(n)\boldsymbol{i} + x_c(n)\boldsymbol{j} + x_d(n)\boldsymbol{k}$$

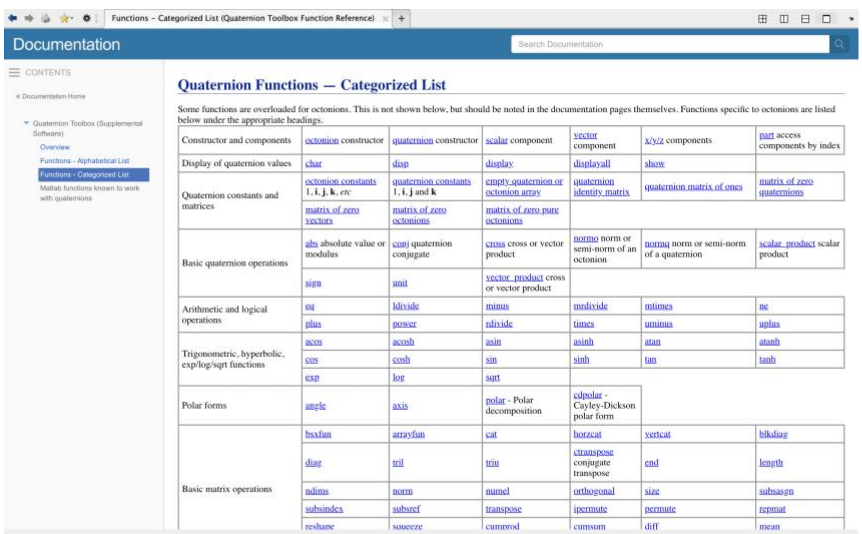
The cost function is minimised using stochastic gradient descent employing the i-gradient and $\mathbb{HR}$-Calculus [33], we get the iQLMS update algorithm as follows.

$$\boldsymbol{h}(n+1) = \boldsymbol{h}(n) + \frac{3}{4}\mu\boldsymbol{e}(n)\boldsymbol{x}^*(\boldsymbol{n})$$

It is clear from this update function that the QLMS is more elegant and efficient, only requiring the calculation of a quaternion conjugate and quaternion multiplication and addition versus the multi-channel LMS case where PQL filter weights must be updated individually, which is $8L$ filter weights for a 4D signal versus L quaternion weights.

2.3.6 MATLAB Quaternion toolbox QTFM

The quaternion toolbox for MATLAB allows quaternions to be created and allows various operations on these objects. Quaternions can be separated into their real and vector parts, signals can be assigned to each component of the quaternion, and conjugation operations may be applied allowing implementation of the QLMS algorithm.


Functions – Categorized List (Quaternion Toolbox Function Reference)

Documentation

Search Documentation

CONTENTS

« Documentation Home

Quaternion Toolbox (Supplemental Software)

Overview

Functions - Alphabetical List

Functions - Categorized List

Matlab functions known to work with quaternions

Quaternion Functions — Categorized List

Some functions are overloaded for octonions. This is not shown below, but should be noted in the documentation pages themselves. Functions specific to octonions are listed below under the appropriate headings.

Constructor and components	octonion constructor	quaternion constructor	scalar component	vector component	x/y/z components	part access components by index
Display of quaternion values	char	disp	display	displayall	show	
Quaternion constants and matrices	octonion constants 1, **i**, **j**, **k**, *etc*	quaternion constants 1, **i**, **j** and **k**	empty quaternion or octonion array	quaternion identity matrix	quaternion matrix of ones	matrix of zero quaternions
	matrix of zero vectors	matrix of zero octonions	matrix of zero pure octonions			
Basic quaternion operations	abs absolute value or modulus	conj quaternion conjugate	cross cross or vector product	normo norm or semi-norm of an octonion	normq norm or semi-norm of a quaternion	scalar_product scalar product
	sign	unit	vector_product cross or vector product			
Arithmetic and logical operations	eq	ldivide	minus	mrdivide	mtimes	ne
	plus	power	rdivide	times	uminus	uplus
Trigonometric, hyperbolic, exp/log/sqrt functions	acos	acosh	asin	asinh	atan	atanh
	cos	cosh	sin	sinh	tan	tanh
	exp	log	sqrt			
Polar forms	angle	axis	polar - Polar decomposition	cdpolar - Cayley-Dickson polar form		
Basic matrix operations	bsxfun	arrayfun	cat	horzcat	vertcat	blkdiag
	diag	tril	triu	ctranspose conjugate transpose	end	length
	ndims	norm	numel	orthogonal	size	subsasgn
	subsindex	subsref	transpose	ipermute	permute	repmat
	reshape	squeeze	cumprod	cumsum	diff	mean

Figure 9 QTFM Documentation

2.3.7 Adapting Least Mean Squares Algorithms for Prediction

In order to adapt each of these algorithms for prediction the desired signal must be altered to be a forwarded version of the input signal, this is the same for all algorithms, single, multi-channel and quaternion. The desired signal $d(n) = x(n + k)$ where k is the number of samples ahead to predict. The prediction adaptive filter diagram is shown in Figure X below.

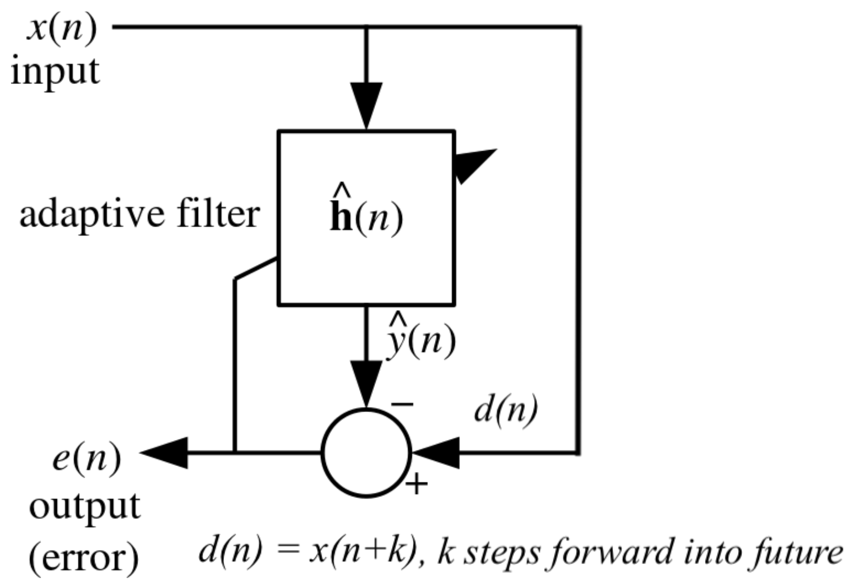


Figure 10 adapted from [28]

3. Methodology

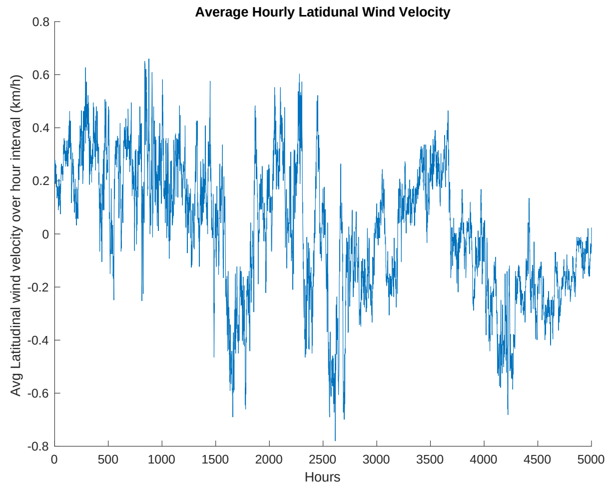
3.1 Datasets

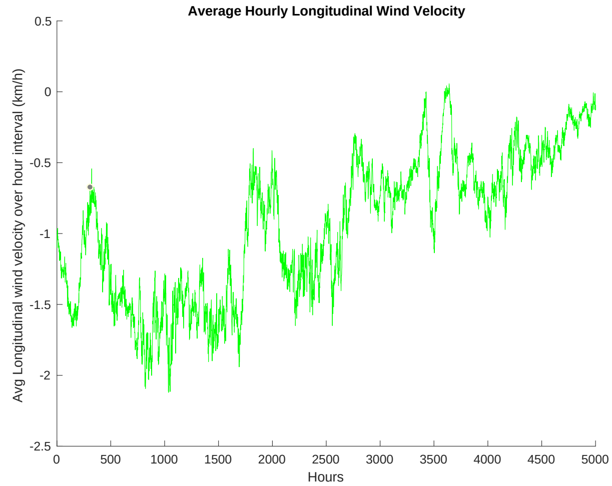
Signal processing will be undertaken on two multi-channel datasets, wind data which includes average hourly latitudinal, longitudinal, and altitudinal components of wind velocity along with the average temperature, these components consist of the 4 channels of the multi-channel signal. A portion of the wind data is show below in Table 1.

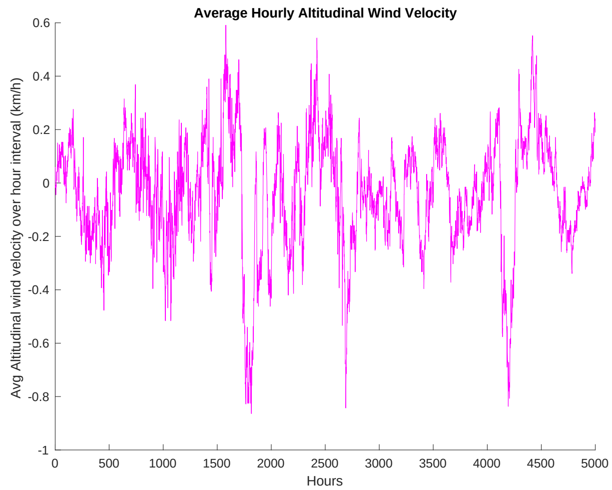
0.2430	-1.1010	0.0600	29.9350
0.2790	-1.0110	-0.0060	29.9700
0.2730	-0.9870	-0.0180	29.9600
0.1980	-0.9810	-0.0120	29.9600
0.2310	-0.9810	-0.0150	29.9250
0.1890	-0.9900	-0.0450	29.9350
0.2130	-0.9630	-0.0330	29.9600
0.2610	-1.0380	-0.0120	29.9400
0.2640	-1.0170	-0.0060	29.9200
0.2580	-1.0380	-0.0150	29.9300
0.2490	-1.0470	-0.0120	29.9150
0.2580	-1.0410	0.0240	29.9350
0.1920	-1.0560	0.0420	29.9600
0.1980	-1.0800	0.0420	29.9200
0.1770	-1.0980	0.0210	29.9850
0.2250	-1.0650	0.0090	29.9700
0.2250	-1.1010	0.0420	30.0100
0.2250	-1.1010	0.0390	30.0150
0.2070	-1.1070	0.0420	30.0050
0.1890	-1.0860	0.0690	29.9900
0.1770	-1.1310	0.0420	29.9600
0.1410	-1.1130	0.0840	29.9950

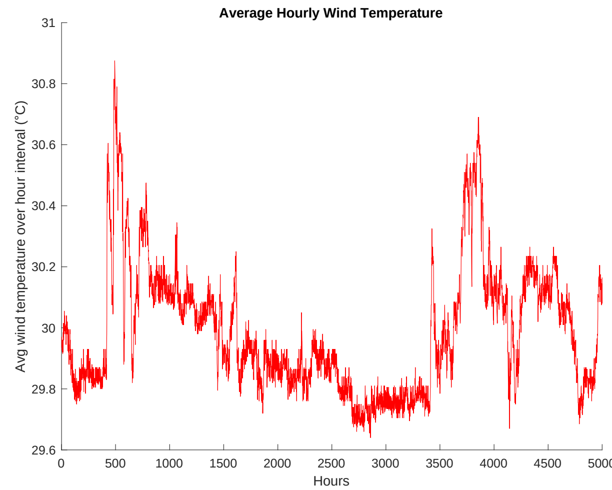
Table 1

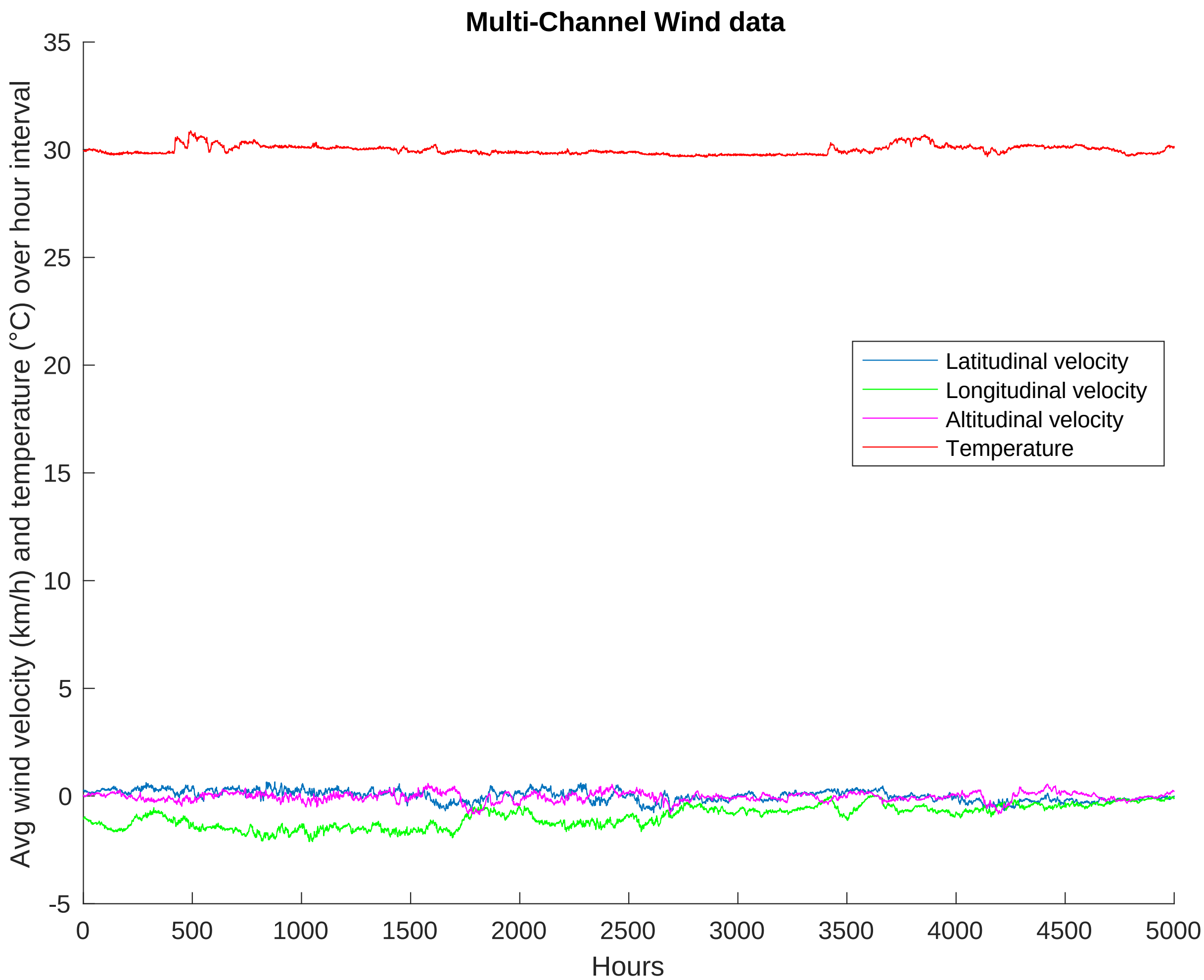
Each of the individual wind channels and their combination are plotted in the figures 11-15 respectively below. Datasets are imported using MATLAB $csvread$ or preloaded as a matrix.


Average Hourly Latidunal Wind Velocity
Avg Latitudinal wind velocity over hour interval (km/h)
Hours


Average Hourly Longitudinal Wind Velocity
Avg Longitudinal wind velocity over hour interval (km/h)
Hours


Average Hourly Altitudinal Wind Velocity
Avg Altitudinal wind velocity over hour interval (km/h)
Hours


Average Hourly Wind Temperature
Avg wind temperature over hour interval (°C)
Hours


Multi-Channel Wind data
Avg wind velocity (km/h) and temperature (°C) over hour interval
Hours
35
30
25
20
15
10
5
0
-5
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
Latitudinal velocity
Longitudinal velocity
Altitudinal velocity
Temperature

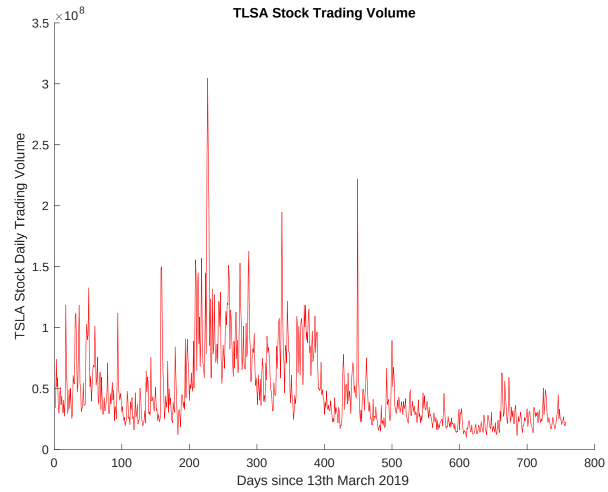
We now consider the other set of multichannel data, TSLA multi-channel stock data which consists of channels which include daily open price (in USD), close price, high price and trading volume, this data was collected from March 2019-March-2022. A portion of the TSLA stock data is shown in Table 2 below.

56.7800	57.7920	58.3980	34223500
58.4900	57.9920	59.0780	35517000
56.7020	55.0860	56.7440	73927500
55.2000	53.8980	55.6100	51405000
53.5000	53.4940	54.6600	59003000
53.9380	54.7200	54.9940	34541000
54.5200	54.8040	55.2900	29735500
54.5160	52.9060	54.5600	43728000
51.9420	52.0840	52.6360	51075000
52.8880	53.5540	54.0520	36754500
53.7500	54.9660	55.0740	43896000
55.4320	55.7240	56.0660	33870500

Table 2

Each of the individual wind channels and their combination are plotted in the figures 16-20 respectively below. A


TLSA Stock High Price
TSLA Stock High Price USD
Days since 13th March 2019


TLSA Stock Trading Volume
TSLA Stock Daily Trading Volume
Days since 13th March 2019

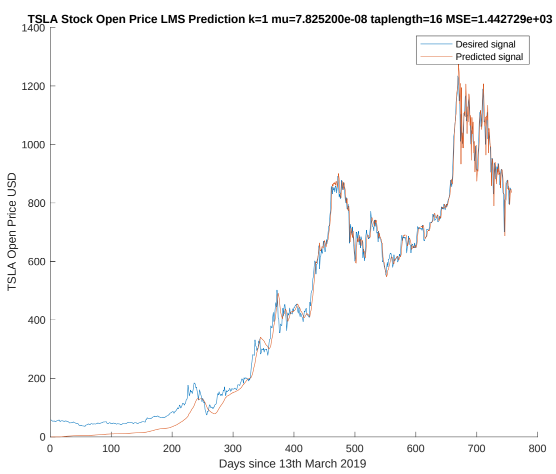

TLSA Stock
Open
Close
High
Trading Volume per 100k units
TSLA Stock Daily Open, Close, and High Price (USD) and Daily Trading Volume per 100k units
Days since 13th March 2019

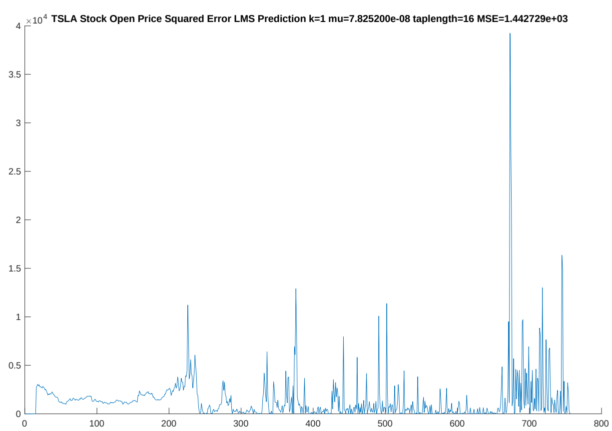
4. Results and Discussion

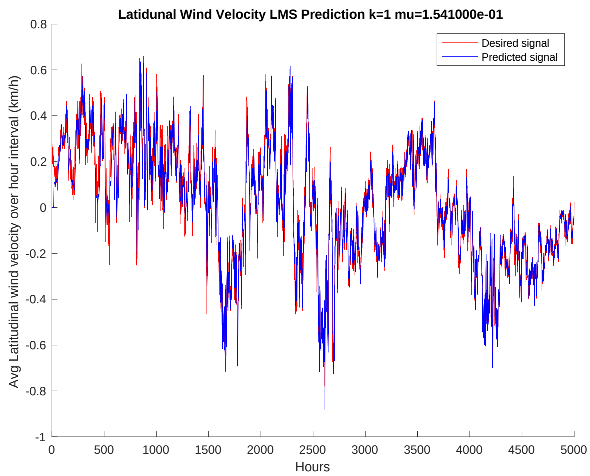
4.0 Introduction

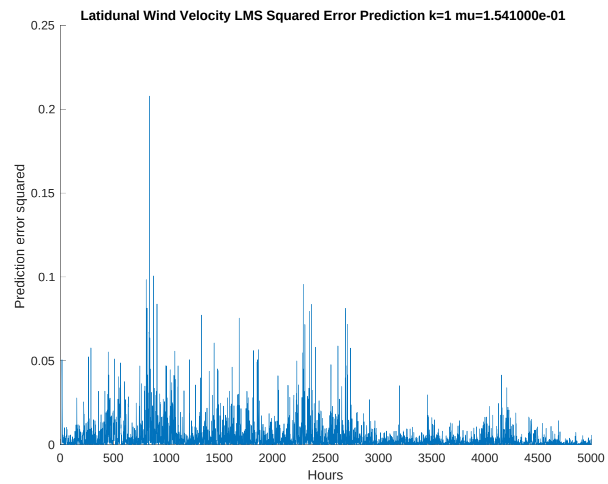
The three implementations of the LMS algorithms used for prediction all with a filter tap length $L = 16$, the single channel LMS, Multi-Channel LMS and QLMS (see Appendix F), are first to be explored and their various constant parameters, such as filter tap length and number of samples ahead to predict, adjusted to study the different filter's behaviour, and then the filter algorithms are to be compared with each other.

As an introduction to see the single channel prediction algorithm in action the plots below Figures 21-24 respectively show the results of the single channel LMS prediction algorithm on latitudinal wind velocity data and TSLA open price stock data. There is a plot for showing how well the prediction algorithm tracks the signal as well as an error plot.







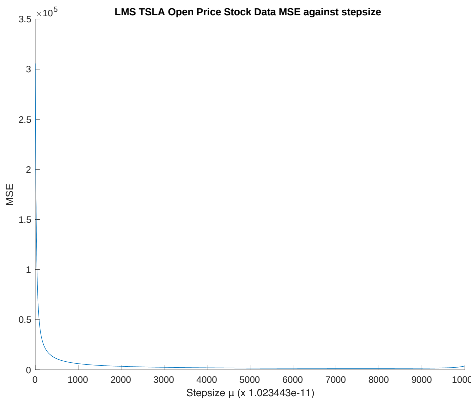


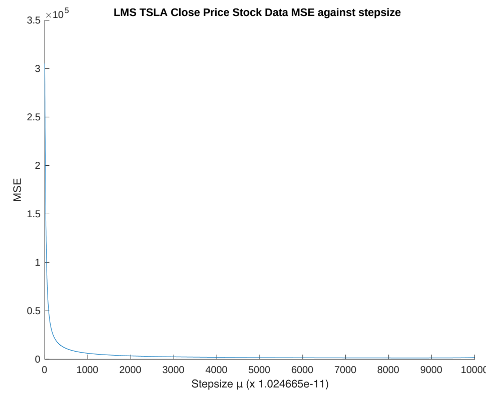
As you can see in the error plots the algorithm converges to its general mean level prediction of prediction inaccuracy and in the other plots you can see the predicted signal follow the desired future shifted signal.

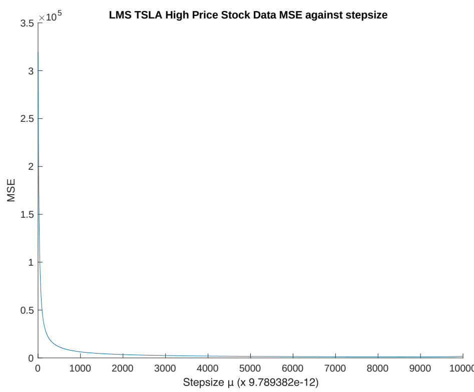
4.1 Step-size Optimisation

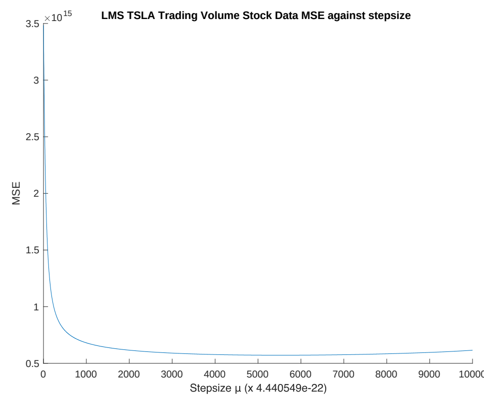
For each algorithm and data set, the step-size μ is optimised via a simple brute force approach by iterating over the algorithm with 10,000 different step-size values, maximally bounded between $0 < \mu < \frac{2}{LP_x}$, where L is the number of filter taps, and P_x is the power of the input signal and finding the minimum value of mean of the squared error. For all simulations $L = 16$ filter taps will be used and the LMS filter will predict $k = 1$ step forward, altering tap length and prediction constant will be explored separately to step-size.

First the step size will be optimised on single LMS on each of the 4 channels of each dataset. The plots below Figures 25-32 show mean squared error against step size on each channel for stock data and wind data. Table 3 and 4 below shows the minimum mean squared error and optimal step-size for each channel.



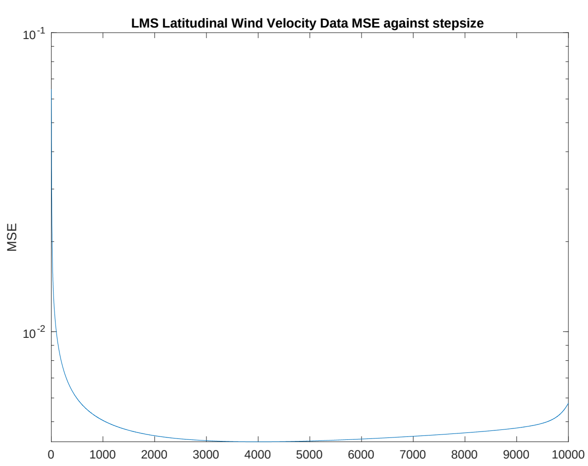


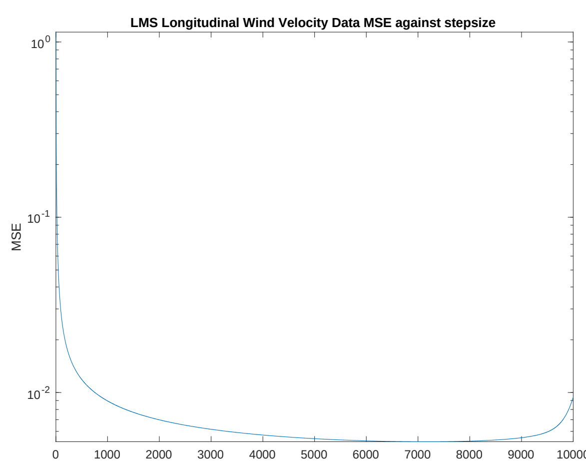


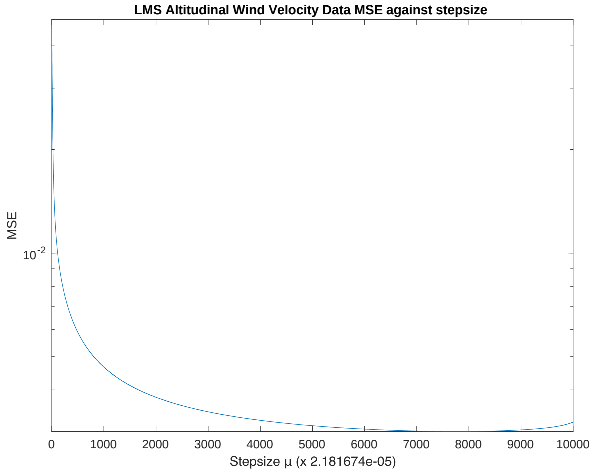


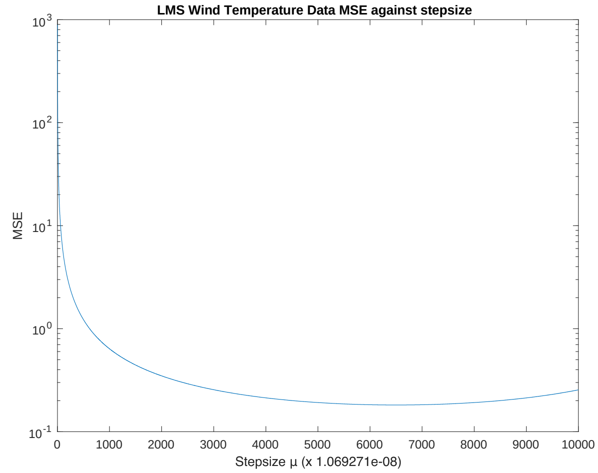
Stock Data Channel	Open Price	Close Price	High Price	Trading Volume
Minimum MSE	1.4427e+03	1.2678e+03	1.1999e+03	5.7160e+14
Optimum step-size μ	7.8252e-08	8.7394e-08	8.3738e-08	2.4499e-18

Table 3





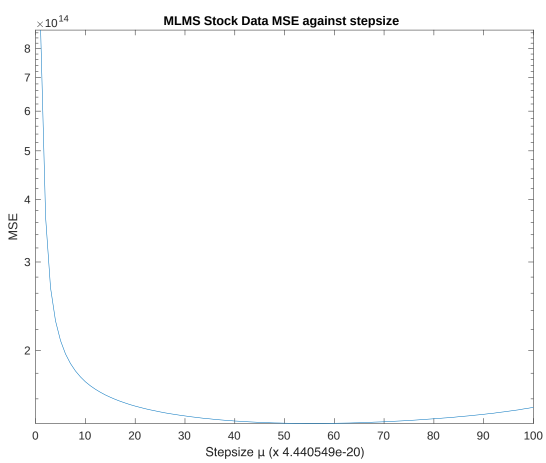


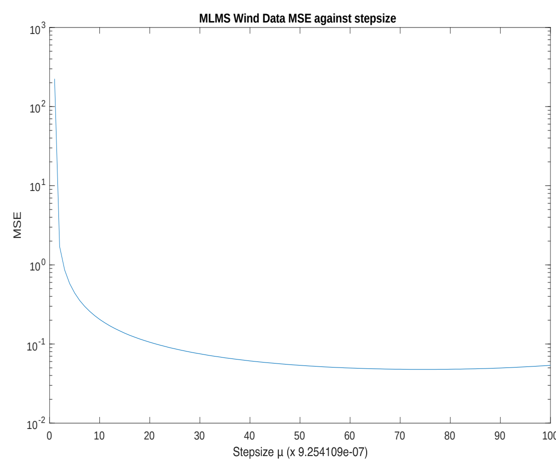


Wind Data Channel	Latitudinal	Longitudinal	Altitudinal	Temperature
Minimum MSE	0.0043	0.0052	0.0030	0.1817
Optimum step-size μ	0.1541	0.0245	0.1695	6.9524e-05

Table 4

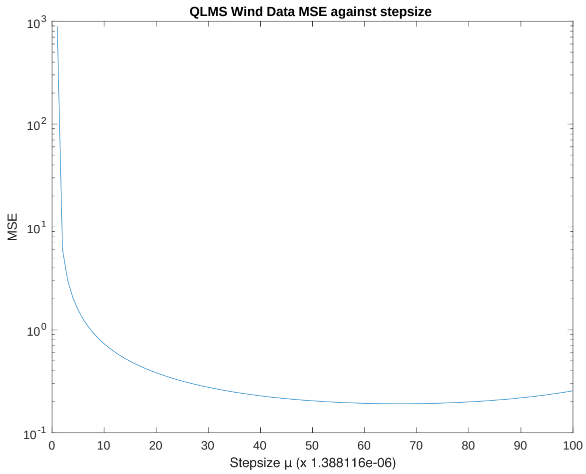
Step-size optimisation for QLMS and Multi-Channel LMS is conducted with 100 different step size μ values, with the range of convergence [citation] of $0 < \mu < \frac{2}{L\sum_{p=1}^{P} \mathrm{P}_{x_p}}$, where P_{x_p} is the power of the pth input signal and L is tap length. The mean of all the channels mean square error is minimized for MLMS and the mean of the squared magnitude of the quaternion for QLMS, so this is the best step size for minimising error across all the channels. Plots and data are shown in the tables 5 and 6 and Figures 33-36 below. The optimal step size for MLMS and QLMS for wind and stock data are shown below.

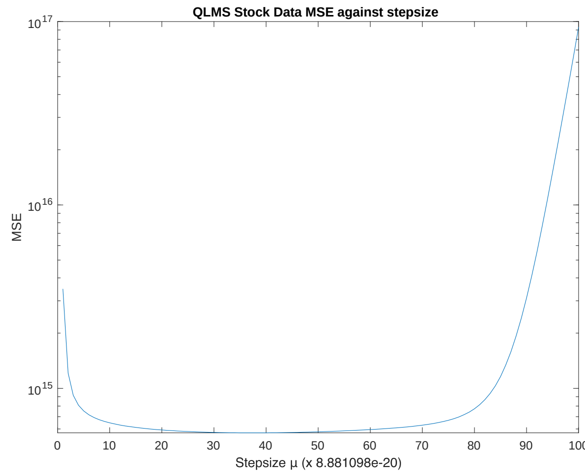




	Wind Data MLMS	Stock Data MLMS
Minimum MSE	0.0477	1.4290e+14
Optimum step-size μ	6.9406e-05	2.4867e-18

Table 5





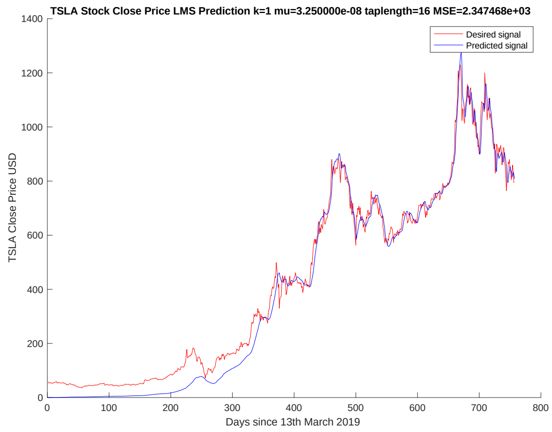
	Wind Data QLMS	Stock Data QLMS
Minimum MSE	0.1916	5.7161e+14
Optimum step-size μ	9.3004e-05	3.2860e-18

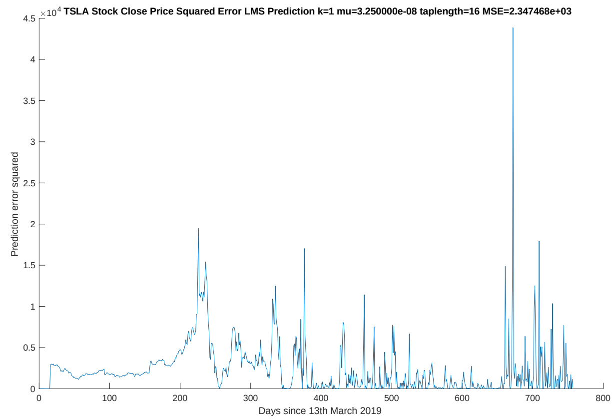
Table 6

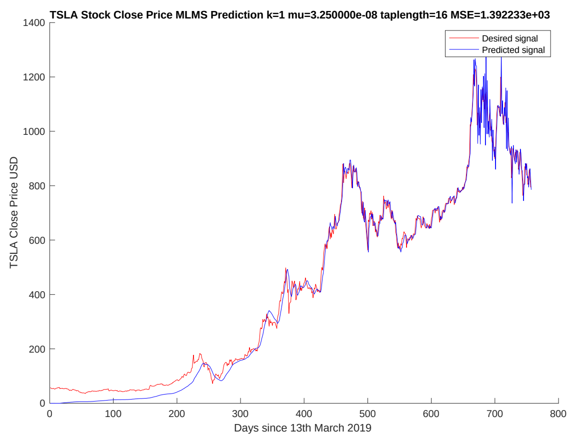
4.3 Analysis of different LMS algorithms

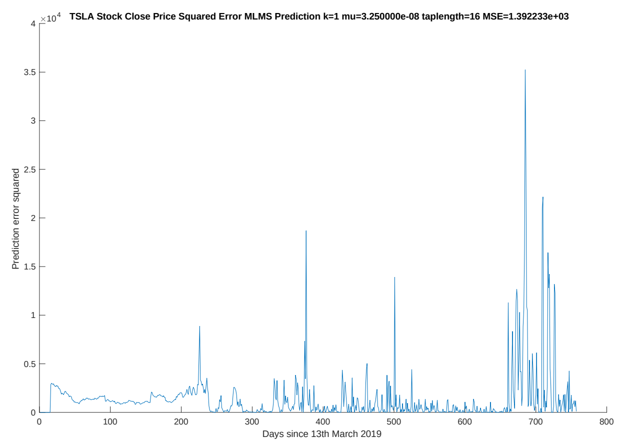
Multiple channels give more information to the signals, and when all the channels' data are used in the algorithm for prediction of a single, the prediction should be more accurate versus a prediction only using a single channel to predict a single channel. This hypothesis will be tested on each data set and the two multichannel algorithms will also be compared.

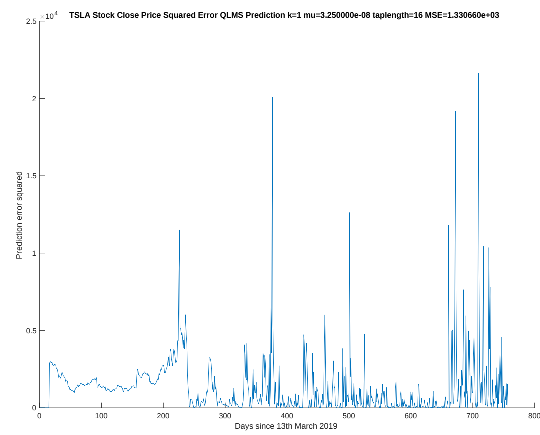
The daily closing TSLA stock price data is analysed. First using single channel LMS which will then be compared against multi-channel LMS and Quaternion LMS on the 4D signal where the other 3 channels, open price, high price, and trading volume, are also used to hypothetically improve prediction. The trading volume is normalised to between 0 and 1. The plots of each algorithm's desired and predicted signal and error plots are shown below in Figures 37-42.



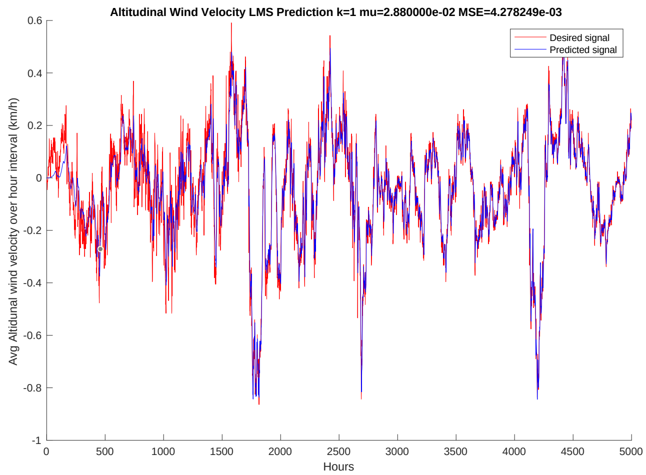


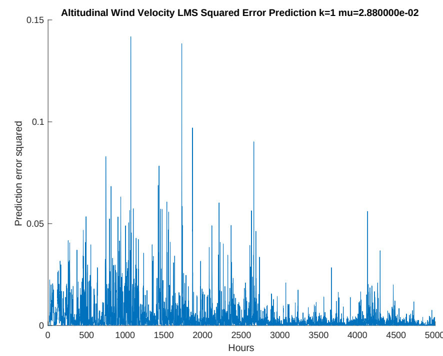


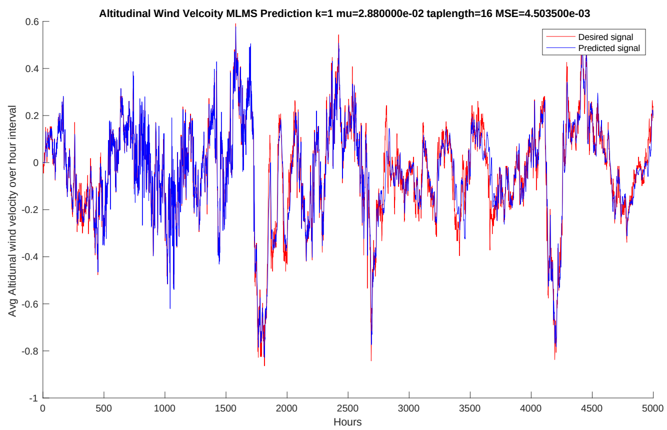


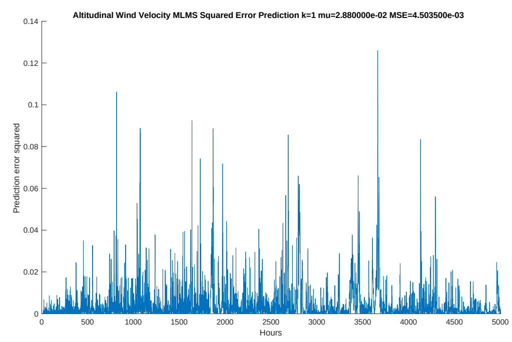


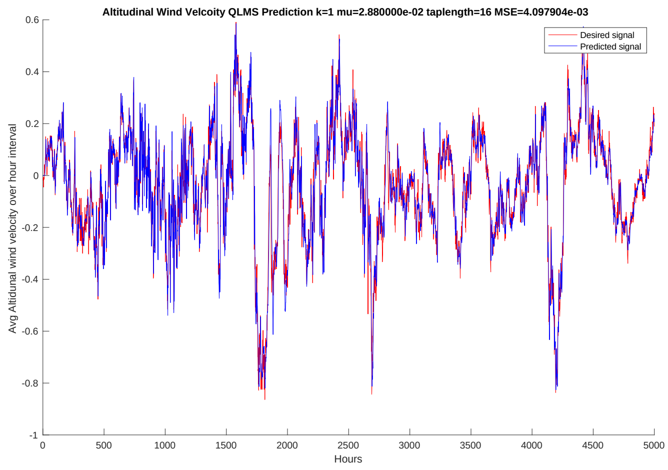
The hourly average altitudinal wind velocity is now analysed, in the same way as before to see if the 3 other channels in the 4D signal, longitudinal wind velocity, latitudinal wind velocity and wind temperature can improve prediction. Wind temperature is centred around its median to make data more like other channels. Plots are shown below in Figures 43-48.

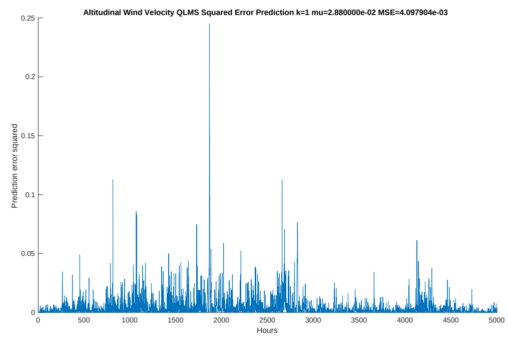










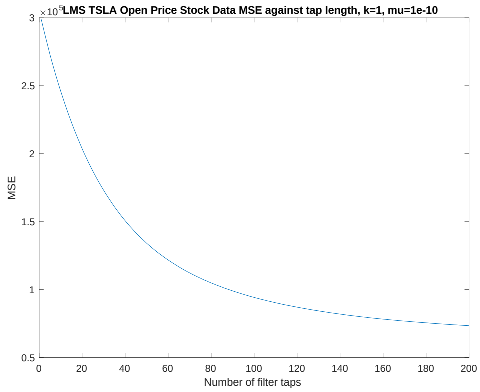


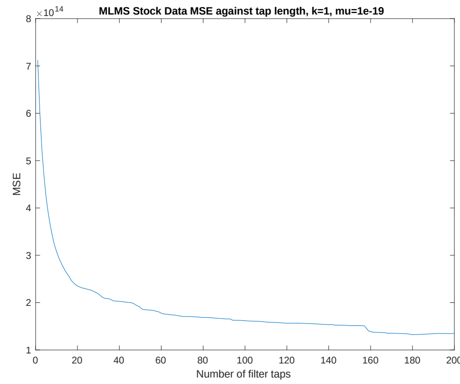
For the TSLA close price analysis, the mean square of the QLMS algorithm is the lowest, with the MLMS mean squared error being slightly higher followed by the single channel LMS algorithm, showing that multi-channel data does give a more accurate prediction as there is more information to make predictions from. With the wind data however the single channel LMS has a slightly lower mean squared error than the multi-channel LMS but higher than the QLMS, this result is anomalous however, but the QLMS still shows the superiority of multiple channels for prediction.

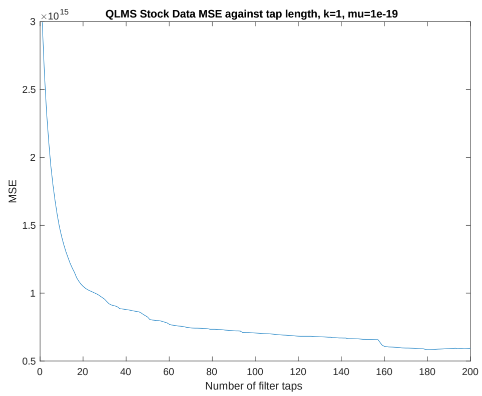
4.4 Exploring filter tap lengths

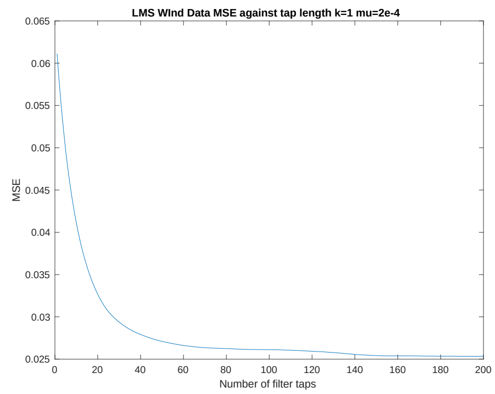
The number of filter weights $\boldsymbol{L}$ in a digital filter is known as its tap length. In the case of prediction using the LMS and algorithms, increasing tap length should lead to better information about the system as there are more parameters to optimise.

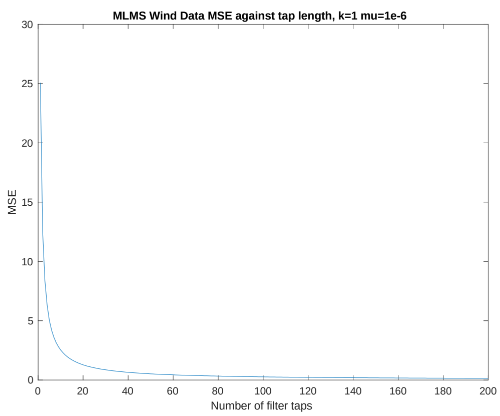
TSLA stock open price data and Latitudinal wind velocity data is predicted using the LMS algorithm, at each iteration on this data the tap length is varied $1 \leq L \leq 200$ and means squared error recorded. This also done using the MLMS and QLMS algorithms. This is shown in the plots below in Figures 49-54.

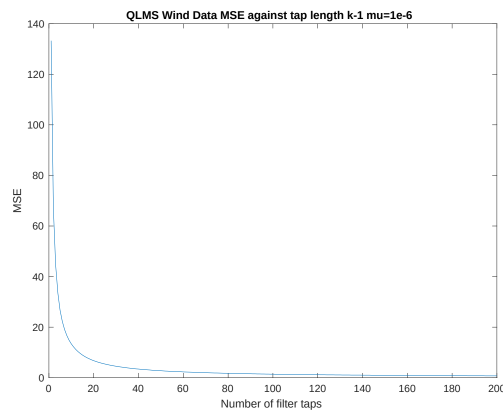








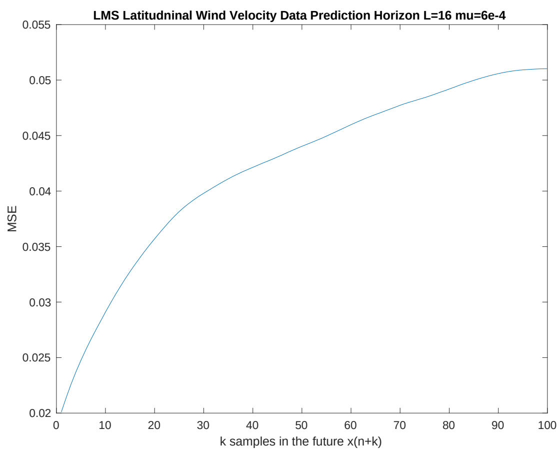


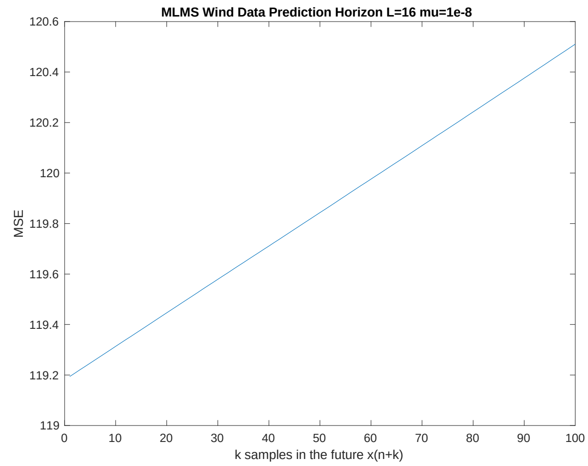


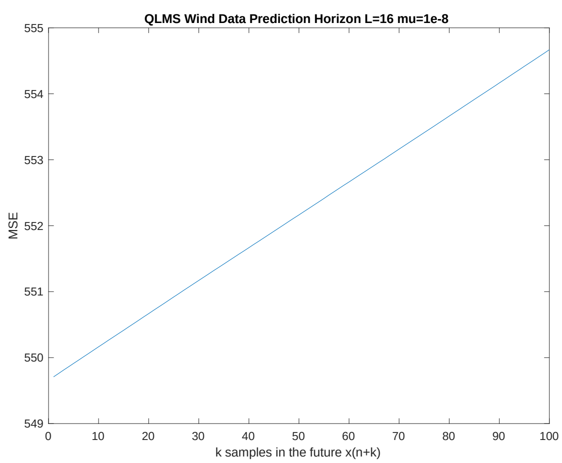
The shapes of the plots show increasing the number of weights causes a decrease in mean square error, however this relationship appears non-linear with asymptotic nature indicating diminishing returns in mean square error in increasing $\boldsymbol{L}$.

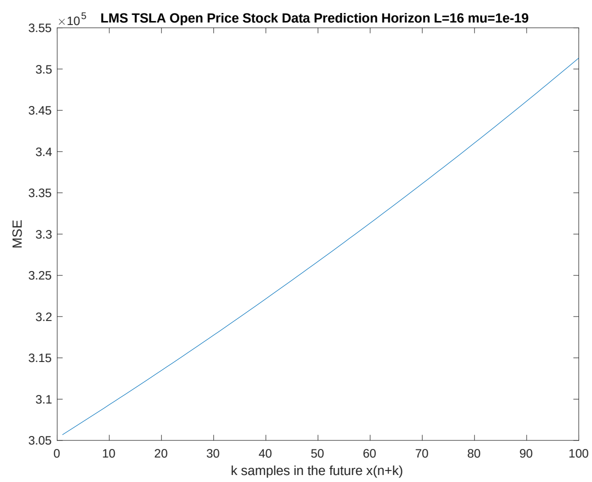
4.5 Prediction horizon

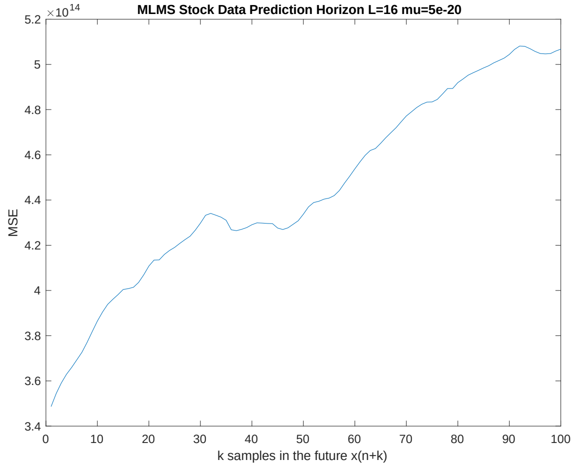
The number of samples predicted forward in the future k will be altered and the results analysed, it is expected that increasing k will lead to more inaccurate predictions, as data further in the past will have less of an influence than more recent data in determining future samples. The plots below Figures 55-60 show the mean squared error against variable k with bounds $1 \leq k \leq 100$, and with $L = 16$ on each data set with the different algorithms.

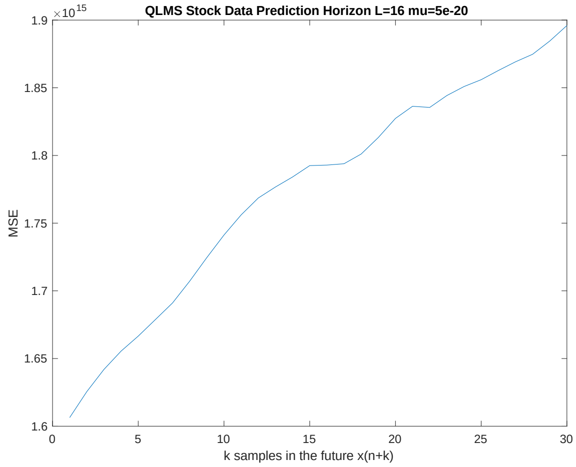












We see a general monotonically increasing relation between mean squared error and k, showing that prediction further forward in the future is more inaccurate due to greater extrapolation.

5. Conclusion

Least Mean Squares adaptive filtering applied to the problem of signal prediction depends on multiple parameters such as step size, filter tap length and number of samples predicted forward in the future. Through analysis we were able to establish general relationships for filter tap length, where an increase in the number of filter weights causes a decrease in error as hypothesised due to their being more parameters to alter, but with there being diminishing returns. A general increasing relationship for mean square error and number of samples predicted forward in the future was also noticed, confirming the hypothesis that further extrapolation from samples would cause more error. In the comparison of the algorithms on the 4D (multichannel) signal of single channel LMS, Multi-channel LMS and Quaternion LMS, we see that prediction accuracy is generally enhanced with addition of related data in other channels, with the QLMS algorithm giving the best accuracy, with MLMS giving slightly less accuracy and sometimes giving higher accuracy than the single channel LMS which does not have any input from the other channels. We can conclude that there is benefit using related information as other channels to enhance prediction. The QLMS algorithm shows its superiority in its accuracy as well as its elegance.

References

[1] S. Reddy Beeram and S. Kuchibhotla, "A Survey on state-of-the-art Financial Time Series Prediction Models," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 596-604, doi: 10.1109/ICCMC51019.2021.9418313.

[2] C. R. Madhuri, M. Chinta and V. V. N. V. P. Kumar, "Stock Market Prediction for Time-series Forecasting using Prophet upon ARIMA," 2020 7th International Conference on Smart Structures and Systems (ICSSS), 2020, pp. 1-5, doi: 10.1109/ICSSS49621.2020.9202042.

[3] Čepulionis Paulius, Lukoševičiūtė Kristina Electrocardiogram time series forecasting and optimization using ant colony optimization algorithm. Mathematical Models in Engineering, Vol. 2, Issue 1, 2016, p. 69-77.

[4] F. Shen, J. Liu and K. Wu, "Multivariate Time Series Forecasting Based on Elastic Net and High-Order Fuzzy Cognitive Maps: A Case Study on Human Action Prediction Through EEG Signals," in IEEE Transactions on Fuzzy Systems, vol. 29, no. 8, pp. 2336-2348, Aug. 2021, doi: 10.1109/TFUZZ.2020.2998513.

[5] Runtian Zhang and Qian Zou 2018 J. Phys.: Conf. Ser. 1061 012012

[6] U. Barman, R. D. Choudhury, A. Ekbal Hussain, M. J. Dahal, P. Barman and M. Hazarika, "Comparative Assessment of AR, MA and ARMA for the Time Series Forecasting of Assam and Meghalaya Rainfall Division," 2020 International Conference on Computational Performance Evaluation (ComPE), 2020, pp. 507-511, doi: 10.1109/ComPE49325.2020.9200014.

[7] N. Stamatis, D. Parthimos and T. M. Griffith, "Forecasting chaotic cardiovascular time series with an adaptive slope multilayer perceptron neural network," in IEEE Transactions on Biomedical Engineering, vol. 46, no. 12, pp. 1441-1453, Dec. 1999, doi: 10.1109/10.804572.

[8] A. Tokgöz and G. Ünal, "A RNN based time series approach for forecasting turkish electricity load," 2018 26th Signal Processing and Communications Applications Conference (SIU), 2018, pp. 1-4, doi: 10.1109/SIU.2018.8404313.

[9] F. Hamami and I. A. Dahlan, "Univariate Time Series Data Forecasting of Air Pollution using LSTM Neural Network," 2020 International Conference on Advancement in Data Science, E-learning and Information Systems (ICADEIS), 2020, pp. 1-5, doi: 10.1109/ICADEIS49811.2020.9277393.

[10] S. Kwong and K. F. Man, "A speech coding algorithm based on predictive coding," Proceedings DCC '95 Data Compression Conference, 1995, pp. 455-, doi: 10.1109/DCC.1995.515565.

[11] L. -. Wang and J. M. Mendel, "An RLS fuzzy adaptive filter, with application to nonlinear channel equalization," [Proceedings 1993] Second IEEE International Conference on Fuzzy Systems, 1993, pp. 895-900 vol.2, doi: 10.1109/FUZZY.1993.327396.

[12] E. Szopos, M. Topa and N. Toma, "System identification with adaptive algorithms," 2005 IEEE 7th CAS Symposium on Emerging Technologies: Circuits and Systems for 4G Mobile Wireless Communications, 2005, pp. 64-67, doi: 10.1109/EMRTW.2005.195681.

[13] A. Ükte and A. Kızılkaya, "Adaptive noise cancellation with a multirate normalized least mean squares filter," 2015 23nd Signal Processing and Communications Applications Conference (SIU), 2015, pp. 1417-1420, doi: 10.1109/SIU.2015.7130108.

[14] S. Ohta, Y. Kajikawa and Y. Nomura, "Acoustic Echo Cancellation using Sub-Adaptive Filter," 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07, 2007, pp. I-85-I-88, doi: 10.1109/ICASSP.2007.366622.

[15] P. Prandoni and M. Vetterli, "An FIR cascade structure for adaptive linear prediction," in IEEE Transactions on Signal Processing, vol. 46, no. 9, pp. 2566-2571, Sept. 1998, doi: 10.1109/78.709548.

[16] J. E. Wesen, V. VV. Vermehren, and H. M. de Oliveira, Adaptive Filter Design for Stock Market Prediction Using a Correlation-based Criterion. arXiv, 2015. doi: 10.48550/ARXIV.1501.07504.

[17] Felipe Tobar, Iván Castro, Jorge Silva, Marcos Orchard, Improving battery voltage prediction in an electric bicycle using altitude measurements and kernel adaptive filters, Pattern Recognition Letters, Volume 105, 2018, Pages 200-206, ISSN 0167-8655, https://doi.org/10.1016/j.patrec.2017.09.009.

[18] Zhu Li and Chao Li, "LMS and RLS algorithms comparative study in system identification," 2011 International Conference on Multimedia Technology, 2011, pp. 5428-5430, doi: 10.1109/ICMT.2011.6002287.

[19] Mannan, Abdul & Habib, Aamir. (2017). Adaptive processing of image using DWT and FFT OFDM in AWGN and Rayleigh channel. 10.1109/C-CODE.2017.7918955.

[20] D. Satpathy and P. K. Sahoo, “Design of Adaptive Noise Canceller using LMS Algorithm,” International journal of engineering research and technology, vol. 3, 2018.

[21] Leng, S., Tan, R.S., Chai, K.T.C. et al. The electronic stethoscope. BioMed Eng OnLine 14, 66 (2015). https://doi.org/10.1186/s12938-015-0056-y

[22] Tian, Xiange & Wang, Tie & Chen, Zhi & Gu, Fengshou & Ball, Andrew. (2013). A Transient Enhancement Method for Two-Stage Helicopter Gearbox Fault Diagnosis Based on ALE. Journal of Signal and Information Processing. 04. 132-137. 10.4236/jsip.2013.43B023.

[23] Paul, Baby & Balakrishnan, Arun & K., Prasanth. (2013). Fetal ECG Extraction Using Adaptive Filters. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 2. 2320-3765.

[24] Chan, H., Tsai, Y., Meng, L. and Wu, T., 2010. The Removal of Ocular Artifacts from EEG Signals Using Adaptive Filters Based on Ocular Source Components. Annals of Biomedical Engineering, 38(11), pp.3489-3499.

[25] S Selvan & R Srinivasan (2000) Adaptive Filtering Techniques using Neural Networks, IETE Technical Review, 17:3, 111-118, DOI: 10.1080/02564602.2000.11416891

[26] Shen, Dinghan & Min, Martin & Li, Yitong & Carin, Lawrence. (2017). Adaptive Convolutional Filter Generation for Natural Language Understanding.

[27] Yunhui, Guo & Li, Yandong & Wang, Liqiang & Rosing, Tajana. (2020). AdaFilter: Adaptive Filter Fine-Tuning for Deep Transfer Learning. Proceedings of the AAAI Conference on Artificial Intelligence. 34. 4060-4066. 10.1609/aaai.v34i04.5824.

[28] En.wikipedia.org. 2022. Least mean squares filter - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Least_mean_squares_filter> [Accessed 1 May 2022].

[29] En.wikipedia.org. 2022. Gradient - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Gradient#/media/File:3d-gradient-cos.svg> [Accessed 8 May 2022].

[30] Mehendale, Ninad & khandhar, jay & vora, ruchik & Shah, Dhruvil. (2012). ADAPTIVE FILTER IMPLEMENTATION USING FPGA. Technofocus. 2. 133.

[31] Benesty, J., Gänsler, T., Huang, Y., Rupp, M. (2004). Adaptive Algorithms for Mimo Acoustic Echo Cancellation. In: Huang, Y., Benesty, J. (eds) Audio Signal Processing for Next-Generation Multimedia Communication Systems. Springer, Boston, MA. https://doi.org/10.1007/1-4020-7769-6_5

[32] C. C. Took, C. Jahanchahi and D. P. Mandic, "A unifying framework for the analysis of quaternion valued adaptive filters," 2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), 2011, pp. 1771-1774, doi: 10.1109/ACSSC.2011.6190325.

[33] Amann, A., Klotz, A., Niederklapfer, T. et al. Reduction of CPR artifacts in the ventricular fibrillation ECG by coherent line removal. BioMed Eng OnLine 9, 2 (2010). https://doi.org/10.1186/1475-925X-9-2

Bibliography

En.wikipedia.org. 2022. Stochastic gradient descent - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Stochastic_gradient_descent> [Accessed 8 May 2022].

J. G. Avalos, J. C. Sanchez, and J. Velazquez, "Applications of Adaptive Filtering", in Adaptive Filtering Applications. London, United Kingdom: IntechOpen, 2011 [Online]. Available: https://www.intechopen.com/chapters/16112 doi: 10.5772/16873

“Gradient descent - Wikipedia", En.wikipedia.org, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Gradient_descent. [Accessed: 08- May- 2022].

D. P. Mandic, C. Jahanchahi and C. C. Took, "A Quaternion Gradient Operator and Its Applications," in IEEE Signal Processing Letters, vol. 18, no. 1, pp. 47-50, Jan. 2011, doi: 10.1109/LSP.2010.2091126.

Appendices

Appendix A – Project Specification

EE3000 - Individual Project Specification Form – 2021/2022

Project Title: Signal Processing of 3D and 4D processes

Student Name: Tom Devereux

Supervisor(s): Clive Cheong Took

BRIEF DESCRIPTION AND AIMS OF THE PROJECT:

In this project wind and stock data will be processed as multi-channel discrete signals over time, the signals will be put through a Least Means Squares adaptive filter system and this LMS filter will be used to predict future samples in the signal. As the wind and stock data is multi-channel the use of quaternions will be explored, and the signals will be put through a quaternion LMS filter. The aims are as follows: To explore the LMS filter and how it can be used for prediction, to adapt the LMS filter to use quaternions and to explore the differences between using a system that takes in 4 real channels versus 1 quaternion.

PRIMARY GOALS:

1. Gain an understanding of the LMS filter and algorithm and how it can be used for prediction
2. Adapt the LMS filter and algorithm to be used with quaternions in order to analyse multi-channel signals
3. Investigate optimisations of the LMS Quaternion algorithm
4. Explore the difference of using a system that takes in 4 real channels versus 1 quaternion and how the algorithms differ in performance and other metrics.

SECONDARY GOALS:

1. Develop Multi-channel quaternion LMS filter system and compare to using larger multi-channel LMS systems.

2. An LMS Adaptive Filter is very similar to a single perceptron in a neural network, explore creating a small network of these LMS filters and how it affects the powers of prediction.

RESOURCES NEEDED:

MATLAB
Wind and stock data

HEALTH AND SAFETY ASSESSMENT:

This project involves no electronic work so there are no serious hazards and only involves programming and processing signal data on MATLAB. However proper health and safety precautions must be met. When working at the computer you must hold the proper posture to protect your back, and you must make sure to take breaks and not spend exorbitant amounts of time in front of the screen.

ETHICAL ISSUES:

This project only uses wind and stock data and does not use any personal data that may have any ethical impacts socially and on the wellbeing or people, neither does it have any environmental concerns. There is a potential for misuse if the systems developed prediction algorithms are used unwisely such as predicting the price of a future stock and making an unwise financial decision like purchasing a risky derivative instrument on said stock. Use of predictions made by the program is at a third-party's own discretion.

Supervisor(s) Signature: Clive Cheong Took

Date:08 May 2022

Student's Signature: Thomas Devereux

Date: 08 May 2022

Appendix B – Progress Forms

EE3000 - Individual Project Supervision and Progress Log

Date: 20th October 2021

Student Name: Tom Devereux

Project Title: Signal Processing of 3D and 4D processes

Supervisor: Clive Cheong Took

What tasks have you been assigned during the last month?	1.Initial research into LMS Algorithm 2.Initial research into adaptive filters 3.Implement LMS Algorithm 4.
Which of the above tasks have you completed?	1,2.
Any problems encountered? How do you intend to resolve them?	No
What tasks have you been assigned for the next month?	1.Implement LMS Algorithm for prediction 2. 3. 4.

Date: 20th November 2021

Student Name: Tom Devereux

Project Title: Signal Processing of 3D and 4D processes

Supervisor: Clive Cheong Took

What tasks have you been assigned during the last month?	1.Adapting Single LMS Algorithm for prediction
Which of the above tasks have you completed?	1
Any problems encountered? How do you intend to resolve them?	No
What tasks have you been assigned for the next month?	1. Testing Single Channel LMS algorithm with different parameters 2. Optimising for step size 3. Tap length and prediction horizon graphs 4. Research into QLMS filter algorithm

Date: 20th December 2021

Student Name: Tom Devereux

Project Title: Signal Processing of 3D and 4D processes

Supervisor: Clive Cheong Took

What tasks have you been assigned during the last month?	1. Testing Single Channel LMS algorithm with different parameters 2. Optimising for step size 3. Tap length and prediction horizon graphs 4. Research into QLMS filter algorithm
Which of the above tasks have you completed?	1, 2,3,4
Any problems encountered? How do you intend to resolve them?	For step size optimisation couldn't get fmbind to work so used brute force method.
What tasks have you been assigned for the next month?	1. Implementation of QLMS Algorithm

Date: 20th January 2022

Student Name: Tom Devereux

Project Title: Signal Processing of 3D and 4D processes

Supervisor: Clive Cheong Took

What tasks have you been assigned during the last month?	1.Implementing QLMS Algorithm for prediction
Which of the above tasks have you completed?	1
Any problems encountered? How do you intend to resolve them?	Couldn't get conjugation of quaternion error signal so used iQLMS update rather than HR-QLMS or original QLMS update.
What tasks have you been assigned for the next month?	Researching MultiLMS Algorithm and testing on wind and stock data

Date: 20th February 2022

Student Name: Tom Devereux

Project Title: Signal Processing of 3D and 4D processes

Supervisor: Clive Cheong Took

What tasks have you been assigned during the last month?	1. Researching Multi LMS Algorithm for prediction 2. Implementation and testing on wind and stock data
Which of the above tasks have you completed?	1,2
Any problems encountered? How do you intend to resolve them?	Initially I tried to implement algorithm with 3d shape matrix operations however I changed to an iterative approach that was easier to follow and implement.

What tasks have you been assigned for the next month?	1.Start Simulations explore tap length, step size, prediction forward samples, compare algorithms 2.Start Literature Review

Date: 20th March 2022

Student Name: Tom Devereux

Project Title: Signal Processing of 3D and 4D processes

Supervisor: Clive Cheong Took

What tasks have you been assigned during the last month?	1.Start Simulations explore tap length, step size, prediction forward samples, compare algorithms 2.Start Literature Review
Which of the above tasks have you completed?	1
Any problems encountered? How do you intend to resolve them?	Make more time for starting Literature review
What tasks have you been assigned for the next month?	1.Literature Review, 2. Results and 3. Project First Draft

Date: 20th April 2022

Student Name: Tom Devereux

Project Title: Signal Processing of 3D and 4D processes

Supervisor: Clive Cheong Took

What tasks have you been assigned during the last month?	1.Literature Review, 2. Results 3. Project First Draft
Which of the above tasks have you completed?	2

Any problems encountered? How do you intend to resolve them?	Time management
What tasks have you been assigned for the next month?	3. Project First Draft

Date: 8th May 2022

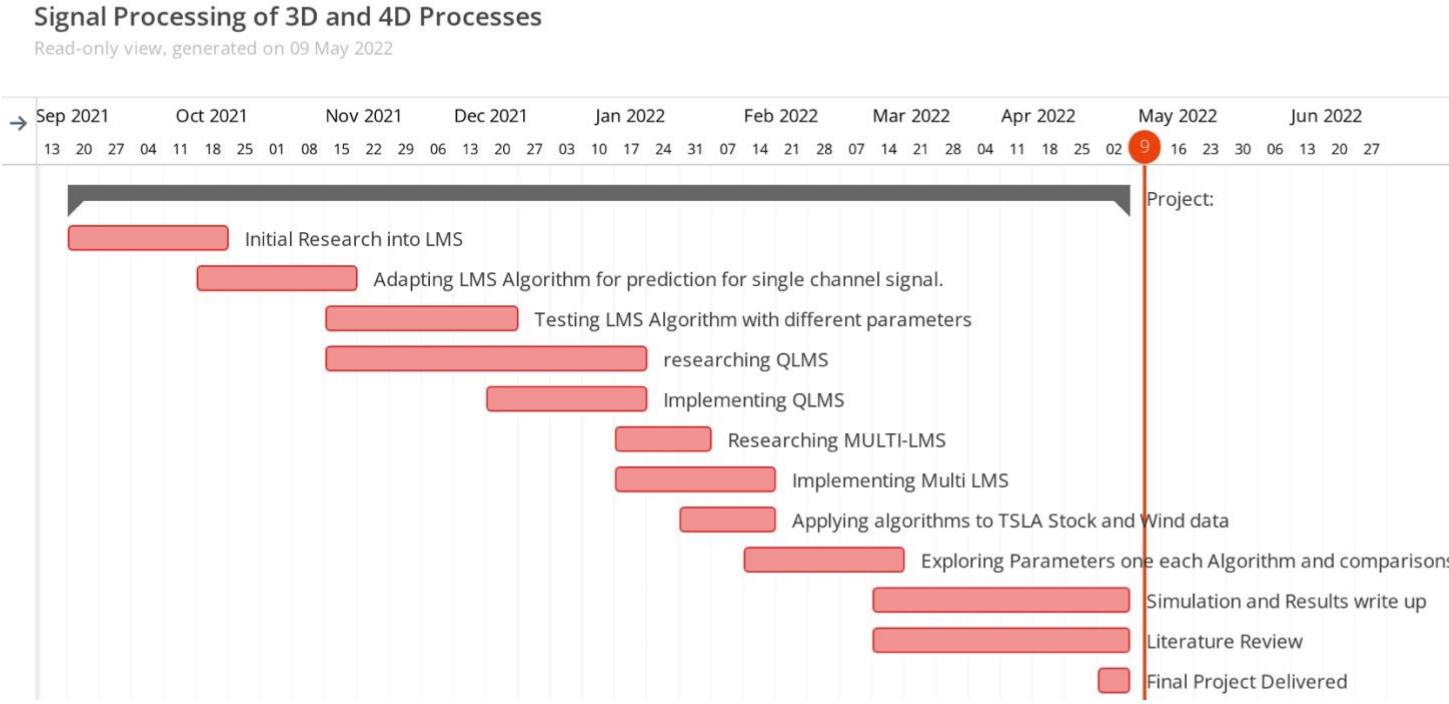
Student Name: Tom Devereux

Project Title: Signal Processing of 3D and 4D processes

Supervisor: Clive Cheong Took

What tasks have you been assigned during the last month?	1. Project First Draft 2. Project Second Draft
Which of the above tasks have you completed?	1,2
Any problems encountered? How do you intend to resolve them?	
What tasks have you been assigned for the next month?	

Appendix C – Gantt Chart



Appendix D – Ethics Form

EE3000 – Individual Project Ethics Form

2021-2022

	Please answer the following
1	(a) Are you using personal data human specimen and/or animals? (e.g., personal information that allows you to identify individuals or company confidential information.) No, only wind and stock data will be used (b) If yes, are <u>you</u> collecting this data? No
2	Who might be adversely affected by your project and why? In this project, predictions are made on the future considering past wind and stock data, if these predictions are used unwisely, for example using these

	predictions and making risky financial decisions or using predicted wind data also for unwise decisions.
3	Are there any environmental, societal, legal or sustainability issues? If yes, how will you address these potential risks? No
4	Could your project and its results be misused by third parties in any way? If yes, explain what measures you have taken to counteract this. This project gives future predictions on wind and stock data, these predictions may be used by any third parties at their own risk.

Note: If the answer to 1 (b) above is 'yes' then the following must also be considered:

Further ethical considerations:

- Informed consent
- Voluntary participation
- Confidentiality
- Anonymity
- Beneficence

Appendix E – Health and Safety Form

ROYAL HOLLOWAY UNIVERSITY OF LONDON

GENERAL RISK ASSESSMENT FORM

Name of Person Undertaking Assessment	Date Conducted	Department / Area/ Topic (including description of what is being assessed)
Tom Devereux	Wednesday 10th Nov	*Signal Processing of 3D and 4D processes* In this project wind and stock data will be processed as multi channel discrete signals over time, the signals will be put through a Least Means Squares adaptive filter system and this LMS filter will be used to predict future samples in the signal. As the wind and stock data is multi channel the use of quaternions will be explored and the signals will be put through a quaternion LMS filter. The aims are as follows: To explore the LMS filter and how it can be used for prediction, to adapt the LMS filter to use quaternions and to explore the differences between using a system that takes in 4 real channels versus 1 quaternion.

Ref No	Hazard under review	No & Description of Staff/Students/ Others Involved	Existing Controls	Assessed Level of Risk			Further Action Required	By (Date) + Review Date
				L	M	H		
	Potential back injury/pain from working at the computer too long	None only me	Take frequent intermittent breaks from working at computer, maintain correct posture when working	☒	☐	☐	None	
	Potential computer vision syndrome from prolonged computer use	None only me	Take frequent intermittent breaks from working at computer, option to wear computer glasses to reduce eyestrain	☒	☐	☐	None	
	COVID-19 exposure from working in the lab with other students	Anybody in lab at the time	Socially distance, wear face masks	☐	☒	☐	None	
				☐	☐	☐		
				☐	☐	☐		

V2 Nov 2021 Supervisor checked ✓ TOM checked ✓

Appendix F – Programs

Single Channel LMS Algorithm Function

```

% x is input signal
% y is predicted output signal
% k is number of samples forward predicted
% mu is step size
% sysorder is filter tap length
% w is filter weights
% e is error signal
% d is desired output (shifted forward input signal)
% mse is mean squared error


function [y,w,e,d,mse] = LMSFunc(x,k,mu,sysorder)

    N = length(x);
    w = zeros(sysorder,1); % weights array full of sysorder number of zeroes
    y = zeros(N-k,1);
    e = zeros(N-k,1);
    d = zeros(N-k,1);

    d(1:sysorder) = x(1+k:sysorder+k);

    for n = sysorder : N-k % n = increasing element of array [sysorder,...,N-k]
        u = x(n:-1:n-sysorder+1) ; % Sysorder length Sub-array created starting from n going backwards by 1, sysorder number of
        y(n)= dot(w, u) ; % dot product of weights and sysorder length of sample inputs
        d(n) = x(n+k);
        e(n) = d(n) - y(n); % desired output - lms

        w = w + mu * e(n) * u; % weights updated by adding mu * sub array * error
    end

    mse = mean(e.^2);
end
```

Zoom: 50% UTF-8 LF LMSFunc Ln 38 Col 1

Multi-Channel LMS Algorithm Function

```matlab

% x is input multichannel data, 4XN
% k is prediction forward
% mu is step size
% P is inputs P=4
% Q is outputs Q=4
% sysorder is filter tap length = 16

function [y,W,e,d,mse] = MLMSFunc(x,k,mu,P,Q,sysorder)
    %x is P rows by N columns double
    % Wpq or Ĥpq is P by Q WEIGHT MATRIX of sysorder length vectors
    % for adaptive LMS filter
    % Sysorder number of prev inputs
    % filter with sysorder weights for each input to output connection
    W = zeros(P,Q,sysorder);
    N=length(x(1,:));
    d = zeros(Q,N-k);
    y = zeros(Q,N-k);
    e = zeros(Q,N-k);

    d(:,1:sysorder) = x(:,1+k:sysorder+k); % make sure desired output is properly initialised

    for q = 1:Q % loop through outputs
        for n = sysorder : N-k % iterate over signal
%             mu=1e-2;
            hx = 0; % estimated qth output initialised at zero
            for p = 1:P % loop through inputs
                u = transpose(x(p,n:-1:n-sysorder+1)); % take pth input subarray of sysorder length
                Wpq = squeeze(W(p,q,:)); % take pqth filter and squeeze to become 1d array of filter weights
                hx = hx + Wpq.' * u; % dot weights with subarray
            end
            y(q,n) = hx; % estimated qth output
            d(q,n) = x(q,n+k); % desired signal is input shifted k forward
            e(q,n) = d(q,n) - y(q,n); % qth error signal
            for p = 1:P  % loop through inputs again
                u = transpose(x(p,n:-1:n-sysorder+1)); % take pth input subarray of sysorder length
                Wpq = squeeze(W(p,q,:)); % take pqth filter and squeeze to become 1d array of filter weights
                W(p,q,:) = Wpq + mu * e(q,n) * u; % take pqth fitler and update weights by incrementing by mu * error * subarray
            end
        end
    end

    mse = [mean(e(1,:).^2), mean(e(2,:).^2), mean(e(3,:).^2), mean(e(4,:).^2)]; % array of mean squared errors for each channel
end


```

Zoom: 50% UTF-8 LF MLMSFunc

QLMS Algorithm Function

```matlab

% x is quaternion input, each part of quaternion is a signal channel
% y is quaternion estimated output
% k is number of samples predicted forward in future
% mu is step size
% sysorder is filter tap length
% w is quaternion weights, sysorder length on each part
% e is error quaternion
% d is desired ouptut quaternion, forward shifted input
% mse is mean of error quaternion squared

function [y,w,e,d,mse] = QLMSFunc(x,k,mu,sysorder)
    % x is expected to be a 1 row by N columns of quaternion
    N = length(x);
    w = quaternion(zeros(1,sysorder),zeros(1,sysorder),zeros(1,sysorder),zeros(1,sysorder)) ; % weights quaternion full of sysorder number of zeroes in each part of quaternion
    w=w.';
    y = quaternion(zeros(1,N-k),zeros(1,N-k),zeros(1,N-k),zeros(1,N-k)) ; %  estimated ouptut quaternion signal initialised to zero
    y=y.';
    e = quaternion(zeros(1,N-k),zeros(1,N-k),zeros(1,N-k),zeros(1,N-k)); % error quaternion initialised to zero
    e=e.';

%     d = quaternion(zeros(1,N-k),zeros(1,N-k),zeros(1,N-k),zeros(1,N-k));
%     d=d.';

    x1 = x.w;
    x2 = x.x;
    x3 = x.y;
    x4 = x.z;

    d_ = zeros(4,N-k);

    d_(1,1:sysorder) = x1(1+k:sysorder+k);
    d_(2,1:sysorder) = x2(1+k:sysorder+k);
    d_(3,1:sysorder) = x3(1+k:sysorder+k);
    d_(4,1:sysorder) = x4(1+k:sysorder+k);

    % ensures desired signal is properly shifted.

    d = quaternion(d_(1,:),d_(2,:),d_(3,:),d_(4,:));
    d=d.';

    %Ensure signal lengths are correct, N-k

    for n = sysorder : N-k
        u = x(n:-1:n-sysorder+1);

%         disp(norm(u));

        y(n) = w.'*u; % Quaternion multiplication to get estimated output
        d(n) = x(n+k);
        e(n) = d(n) - y(n);
        esquared(n) = scalar(e(n)*conj(e(n)));

        %mu=1e-2;
        w = w + 3/4*mu * e(n) * conj(u); %iQLMS update
        %w = w + 1/2*mu( e(n)*conj(u) - 1/2*u*conj(e(n)) ); %HR-QLMS update
        %w = w + 1/2*mu(e(n)*conj(u) - 1/2*conj(u)*conj(e(n))); %QLMS update
    end


%       mse = [mean((e.w).^2), mean((e.x).^2), mean((e.y).^2), mean((e.z).^2)];

    mse = mean(esquared);

end
```

Zoom: 50% UTF-8 LF QLMSFunc

Single LMS

```
[y,w,e,d,mse] = LMSFunc(x,k,mu,sysorder);
```

MLMS run

```
[y,W,e,d,mse] = MLMSFunc(TSLA.',k,mu,P,Q,sysorder);
```

QLMS Input initialization and run

```
QSTOCK = quaternion(TSLA(:,1),TSLA(:,2),TSLA(:,3),TSLA(:,4));
[y,w,e,d,mse] = QLMSFunc(QSTOCK,k,mu,sysorder);
```

The following have same code for each data set and algorithm, MLMSFunc is switched out for either QLMSFunc or LMSFunc,, and TSLA data is swapped out for WIND data:

Prediction horizon

```
% ----- MSE/Prediction ahead

for k = 1:100
    [~,~,~,~,mse] = MLMSFunc(TSLA.',k,mu,P,Q,sysorder);
    if isnan(mse)
        MSEP(k)=0
    else
        MSEP(k) = mean(mse);
    end
end

plot(MSEP)
title('MLMS Stock Data Prediction Horizon');
xlabel('k samples in the future x(n+k)');
ylabel('MSE');

[best,index] = min(MSEP);
```

Tap length

```
for taps = 1:200
    [~,~,~,~,mse] = MLMSFunc(TSLA.',1,mu,P,Q,taps);
    if isnan(mse)
        MSET(taps)=0
    else
        MSET(taps) = mean(mse);
    end
end

plot(MSET)
title('MLMS Stock Data MSE against tap length');
xlabel('Number of filter taps');
ylabel('MSE');

[best,index] = min(MSET)
```

Step size optimisation

```
% ------ MSE/step size

i = 0;

mumax = 2/(sysorder*sum(bandpower(TSLA)));
for t = linspace(0,mumax,100)
    i = i + 1;
    [~,~,~,~,mse] = MLMSFunc(TSLA.',1,t,P,Q,sysorder);
    if isnan(mse)
        MSE(i)=0;
    else
        MSE(i) = mean(mse);
%       mse
    end
%     MSE(i) = QLMS(t);
end

% MSE(isnan(MSE))=[];

[m,idx]=min(MSE(MSE>0))

semilogy(MSE);
title('MLMS Stock Data MSE against stepsize');
xlabel(sprintf('Stepsize μ (x %e)',mumax/100));
% ylabel('MSE');
```

Data Set importing

TSLA Stock Data CSV

TSLA.csv

	A	B	C	D	E	F	G
				TSLA			
	Date	Open	High	Low	Close	AdjClose	Volume
	Datetime	Number	Number	Number	Number	Number	Number
1	Date	Open	High	Low	Close	Adj Close	Volume
2	2019-03...	56.779999	58.397999	56.540001	57.792000	57.792000	34223500
3	2019-03...	58.490002	59.077999	57.658001	57.992001	57.992001	35517000
4	2019-03...	56.702000	56.743999	54.880001	55.085999	55.085999	73927500
5	2019-03...	55.200001	55.610001	53.459999	53.897999	53.897999	51405000
6	2019-03...	53.500000	54.660000	52.692001	53.493999	53.493999	59003000
7	2019-03...	53.938000	54.993999	53.259998	54.720001	54.720001	34541000
8	2019-03...	54.520000	55.290001	53.689999	54.804001	54.804001	29735500
9	2019-03...	54.515999	54.560001	52.799999	52.905998	52.905998	43728000

Importing

```
STOCK = csvread("TSLA.csv",1,1);

TSLA=zeros(758,4);

TSLA(:,1)=STOCK(:,1);
TSLA(:,2)=STOCK(:,4);
TSLA(:,3)=STOCK(:,2);
TSLA(:,4)=STOCK(:,6);
```

Wind data Matrix

WindFs50

5000x4 double

	1	2	3	4
1	0.2430	-1.1010	0.0600	29.9350
2	0.2790	-1.0110	-0.0060	29.9700
3	0.2730	-0.9870	-0.0180	29.9600
4	0.1980	-0.9810	-0.0120	29.9600
5	0.2310	-0.9810	-0.0150	29.9250
6	0.1890	-0.9900	-0.0450	29.9350
7	0.2130	-0.9630	-0.0330	29.9600
8	0.2610	-1.0380	-0.0120	29.9400
9	0.2640	-1.0170	-0.0060	29.9200
10	0.2580	-1.0380	-0.0150	29.9300
11	0.2490	-1.0470	-0.0120	29.9150
12	0.2580	-1.0410	0.0240	29.9350
13	0.1920	-1.0560	0.0420	29.9600

Importing

```
load("WindRHUL.mat");
WIND = WindFs50.';
```