Stock Market Price Prediction Based on Cubature Kalman Filter

M.Tech. Thesis

By

Richa Kumari
(2102102008)

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE

JUNE 2023

Stock Market Price Prediction Based on Cubature Kalman Filter

A THESIS

Submitted in partial fulfillment of the requirements for the award of the degree

of

Master of Technology

by

RICHA KUMARI

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

JUNE 2023

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **STOCK MARKET PRICE PREDICTION BASED ON USING CUBATURE KALMAN FILTER** in the partial fulfillment of the requirements for the award of the degree of **MASTER OF TECHNOLOGY** and submitted in the **DEPARTMENT OF ELECTRICAL ENGINEERING, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from AUGUST 2021 to JUNE 2023 under the supervision of Dr.Ram Bilas Pachori, Professor, Indian Institute of Technology Indore.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Richa Kumari
09/06/2023

Signature of the student with date
(RICHA KUMARI)

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

09.06.2023

Signature of the Supervisor of
M.Tech. thesis (with date)
(Prof. Ram Bilas Pachori)

RICHA KUMARI has successfully given her M.Tech. Oral Examination held on **MAY 12th 2023.**

Signature(s) of Supervisor(s) of M.Tech. thesis
Date: 09.06.2023

Convener, DPGC
Date: 09/06/2023

Signature of PSPC Member #1
Date: 09-06-2023

09/06/23
Signature of PSPC Member #1
Date:

Acknowledgments

I would like to express my sincere gratitude to my thesis Supervisor Prof. Ram Bilas Pachori, for his constant support, encouragement, and guidance during my Master's study and thesis work.

Furthermore, I would like to express my deep gratitude to my PSPC members Dr. Vijay A.S. and Dr. Girish Verma for their valuable suggestions and feedback on my research.

I would also like to thank Prof. Abhinoy Kumar Singh for giving me valuable comments regarding the problem related to modeling and coding. Also, he helped me in critical technical discussions throughout my master's research work.

I am especially grateful for the discussion with a few Ph.D. Scholars who helped me throughout my work.

IIT Indore played a vital role in providing me with lab facilities, a teaching assistantship, hostel accommodation, and library resources that added to my knowledge and my project.

All this was made possible by the incredible support and blessings of my family members and friends who blessed me all the time, it would not have been possible without their love and support.

Dedication

Dedicated to my family members, who have been my constant support and source of inspiration. Without their love and care, I would never have made it possible.

Abstract

Financial modelling especially in the field of stock market price prediction is a sophisticated process because of the variation and dependency on a few of the parameters and technical indices. This project Thesis examines the application of the Non-linear Kalman Filter in financial modeling.

The Kalman filter is a mathematical algorithm widely used in various fields that use state estimation and prediction. This filter is commonly used in most engineering applications to filter out noise from a measured signal. However, the Kalman filter has also been applied to estimate the value of financial assets, predict asset prices, and manage portfolios in the financial market.

The Kalman filter has been employed for financial modeling with the aim of improving the accuracy of prediction about stock prices, market trends, and other financial metrics. In return Stock markets provide insights to traders to gain high profits.

The financial data of certain companies listed on the National Stock Exchange (NSE) India for the year 2012 from the Yahoo Finance dataset of NSE are used. The prices of the stock for the past days along with a few technical indices were used for the prediction of the price the next day. In finance, the Kalman filter can be used for financial modeling by integrating historical data and making predictions about future stock prices or market trends. To provide more accurate predictions of the next day, the filter takes into account not only the current state of the market but also the uncertainty in the measurements and the dynamics of the market. In this project report, the basics of the Kalman filter have been reviewed, along with its application in the financial market including portfolio optimization, risk management, and algorithmic trading.

The study begins by providing a brief overview of the Kalman filter and its mathematical formulation. The Work then examines the application of the Kalman filter in finance, especially in asset pricing and portfolio management. A step-by-step mathematical derivation of stock market prediction using the Kalman Filter has been presented. The challenges and limitations of using the different models as well as the Kalman filter in financial modeling are also discussed and some recent advances are presented in this field. The results of the study show that the Kalman filter is a useful tool for financial modeling and can help investors and portfolio managers make more informed decisions.

Keywords: State-space model, sequential Bayesian estimation, linearity, time series, Finance, root mean square error, long short-term memory, dynamic mode decomposition, Gaussian noise, linear Kalman filter, and non-linear Kalman filter, stock market.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

In this project work, the propagation pattern of financial data such as stock market prediction using historical data is studied, from a mathematical model perspective. Billions of people participate in the stock market, and each one has unique knowledge and opinions about the value of a company depending on whether they purchase or sell it. Most of it depends on intuition, with a small amount of prediction supported by technology. This is due to the fact that the majority of prediction models now in use are not trustworthy enough for humans to base decisions purely on model predictions. Therefore, there is a need for more effective algorithms to predict stock market prices so that even the average person with little financial knowledge may become independent enough to invest in it with confidence and with the least amount of unanticipated risk and maximum profit.

The world of finance has always been about data. One could even argue that finance professionals utilized data even before the advent of data science, machine learning, and artificial intelligence.

Here are the use cases where data science has helped finance professionals and financial institutions to be more effective and efficient.

1. Fraud Detection and Prevention [64], [38], [57]: Machine learning algorithms can learn from historical data and identify unusual behavior, patterns, or transactions. e.g. identifying theft, or multiple accounts opened with similar KYC data.

2. Risk Assessment and Predictive Analysis [6], [70], [13]: It describes how likely is it that a particular borrower will depend on its future obligations, given its historical payment behavior and other characteristics, e.g. income, family size, address, etc.

3. Forecasting Through Time Series Analysis [9], [49], [55], [68]: By adding an explicit order de-

pendence between historical observations, a time series forecasting can attempt to predict the values of a variable at a given future point in time, either with or without any other dependent variables.

4. Customer Data Management and Analysis [2], [5], [69]: Financial institutions are inundated with humongous data volumes both structured and unstructured, with the latter being more challenging to manage, process and gain insights from.

Stock price prediction is a crucial aspect of finance that involves the development of mathematical models to analyze financial data to forecast future performance and make investment decisions. A variety of purposes, including asset pricing, portfolio management, risk management, and trading can be done using financial data modeling.

In this project, an attempt to predict the stock market using the Kalman filter (KF) algorithm has been made. The KF is a linear state space model [25], [7], [31] that acts recursively on noisy input data and produces a statistically optimum estimation of system data. The KF [75] is a very commonly used signal processing tool [65] that uses a recursive algorithm for estimating the state variables from noisy measurement and observation in linear and non-linear systems by eliminating inaccuracies obtained due to noisy data produced by high fluctuations. Being a time series data set with a high degree of dynamicity, the financial database is an ideal application for the nonlinear KF, which has real-time target tracking capabilities.

In recent years, The KF [75], [76], [10] has emerged as a useful tool for financial data modeling [41], particularly in the area of asset pricing and portfolio management [37]. It is a mathematical algorithm that is widely used in engineering applications to estimate the state of a dynamic system in the presence of noise. The KF has proven to be a powerful and useful tool in the stock market, allowing investors and portfolio managers to better estimate asset values, predict asset prices and manage portfolios. In financial market trading [18], KF are frequently employed to generate estimations of prices and correlation. Instead of using the most current price, they build a price estimate using a time frame of noisy prices that have been observed.

1.2 Motivations

The motivations behind the proposed work are as follows:

- We always have to work with big data which sometimes proves to be cumbersome while removing outliers using machine learning algorithms [63].
- There is no hard theory as to how exactly different factors like interest rates, and market shares vary with utmost precision. So, without any shadow of a doubt, we can say it is a time-consuming exercise that needs to be precise.
- Controlled experiments cannot be conducted, unlike the modeling of a physical system, using the already existing models, so there was a need for a better algorithm that can reduce the error of prediction.
- Collecting and aligning the assumptions that are to be used in the financial market [41] is also not easy while working with the already existing model.

1.3 Application of Kalman Filter in Finance

KF has various applications in the field of finance [77], especially in the field of quantitative finance [36], [61], [27]:

,

- Stock Market Prediction [27]: Non-linear KF can be used to predict stock market data using a historical database based on a state-space model using some mathematical equation for prediction and correction.
- Algorithmic Trading [48], [15]: KF can be easily applied in algorithmic trading [35] methods in order to estimate hidden state variables of financial data, like the true value of an asset of the market.
- Portfolio Optimization [26]: KF can help in balancing risk and return by optimizing the portfolios by estimating and correcting the expected return of an asset [77].
- State estimation in Financial Models [66], [39], [47]: Unobservable variables such as interest rate, volume, opening price, etc. in financial data, can be estimated using KF mathematical state and measurement equation.

1.4 Organization of the Thesis

The rest of this thesis is organized as follows: In chapter 2, the literature review of the previous work done and the problem statement is discussed. Further, in chapter 3, the theory of predictive-financial analysis, Gaussian estimation theory, stochastic process, differential equation, LSTM, Linear Regression, and DMD are carried out. Chapter 4 shows the performance analysis of Linear and Non-Linear Kalman filters using example problems. Next Chapter 5 includes the roadmap of the mathematical model for finance data using a Non-linear KF i.e. CKF. Finally, Chapter 6 includes the conclusion and scope of the future work of the research.

Chapter 2

Review of Past Work and Problem Formulation

2.1 Literature Review

Numerous research has been conducted to evaluate the algorithms that are now being utilized to forecast stock market prices. The overfitting issue plagued artificial neural networks (ANNs), which were frequently employed for stock prediction and financial modeling [40], [16], [51]. This was caused by the numerous parameters and variables that had to be used and fixed and the user's limited prior understanding of the relevance of the inputs in the problem being studied. By memorizing previous data, LSTM [63] was employed for numerous time series applications, and a similar method is used to forecast the trend for the stock. Other machine learning [74] methods are also applied to stock market forecasting [17], [34], [71]. There are no guidelines established by which the best option for the stock market prediction can be selected. The DMD [52], which transforms a dataset from a high-dimensional space to a low-dimensional space so that the low-dimensional representation retains some significant properties of the original information, is also used for stock market forecasting.

Financial signal processing [58], [24], [3], [42] is often used in quantitative analysis to make the best estimation of the movement of financial markets, such as stock prices, option prices, or other types of derivatives. A signal is any sequence of numerical data that varies with respect to a fundamental independent variable mostly time [42]. Techniques in signal processing, particularly stock market price prediction, have been more in demand in the finance sector as a result of the development of digital technology and associated improvements in data storage and processing speed..

KF [54], [53], [32] a linear state space model [31] that operates recursively on noisy input data and generates a statistically ideal assessment of the system state, was later introduced. The use of the KF can assist us in locating a statistically optimal estimate in such a system due to the extremely dynamic nature of stock markets that are also impacted by market noise. The two-stage KF algorithm [30] represents the real worth of the market data by using linear regression within the data. It achieves a balance between the original data and forecasts based on how much noise is there.

Let's say you need to measure the temperature of a fiery path in a rocket booster, no sensor is going to withstand that heat but you can take a measurement from the sensor a few inches hidden behind the heat shell and using that measurement you can pretty closely estimate what the temperature in the booster pathway is. KF does this in a way by minimizing the mean squared error. So, this is more like a prediction algorithm. Therefore, a step has been taken to use a KF for prediction in finance that is in the stock market.

To keep the testing environment consistent, the project analyses these algorithms using the same volume of test and training data. This project thesis work also discusses and compares the relevant parameters and widely utilized algorithms. Later in the study, a comparison of these algorithms based on *RMSE* is shown. Any predictive model's primary goal is to reduce error, and in the case of the stock market, that goal is to reduce risk. According to certain recent developments in this sector, the difficulties and restrictions associated with applying the KF to financial modeling [83], [79] are also covered. Here, the project's goal is to give a comprehensive overview of the KF's use in financial modeling.

2.2 Problem Statement

This project aims to dynamically improve the method of predicting financial distress [40] based on Kalman filtering [30] which is an important area in corporate finance. also, this project report will explain why and how the widely used discriminant models, currently used for financial distress, have deficiencies in dynamics. Financial distress as in to generate sufficient revenues in order to improve the return and reduce risk. The project also focuses on proposing advanced filtering algorithms using KF that can outperform the existing methods like Linear regression, LSTM, etc. Another important method, DMD has also been introduced for stock prediction which uses the dimensionality reduction technique [44]. Later on, all these methods are compared on the basis of Mean square error (MSE). At the end of the report, another important topic, the pros and cons of using the KF in financial analysis, especially in stock market prediction will be analyzed on the basis of simulation results.

Chapter 3

Theoretical Background

3.1 Predictive-Financial analysis

The predictive analysis circumscribes a variety of statistical methods and techniques from data mining [29], [50], predictive modeling, and machine learning that actually analyzes current and historical facts to make predictions of future events. In short, analytics helps in forecasting future performance and results [12].

Fundamentally, a prediction is a calculated guess based on some prior knowledge in the form of facts and pieces of evidence [12]. The stock market forecasting has been one of the buzz topics attracting equally the attention of both researchers and traders who want to make a profit recently.

Now, let's understand what is a financial model [41].

- It is a tool used for decision-making. Whenever there is a need to take a decision rather sensible decision, we need data or information to support them in the form of a model.

- It is a kind of activity of preparing any entity's future financial statements. These future financial statements are sometimes known as financial models.

Consequently, historical data are essentially used to create a classifier or predictive [12] model or a regressor that actually captures the important trend, and then the current data is used to predict what will happen next and which further helps to take optimal outcomes.

Below is the basic process steps flow diagram to be followed during the process of predictive financial modeling :

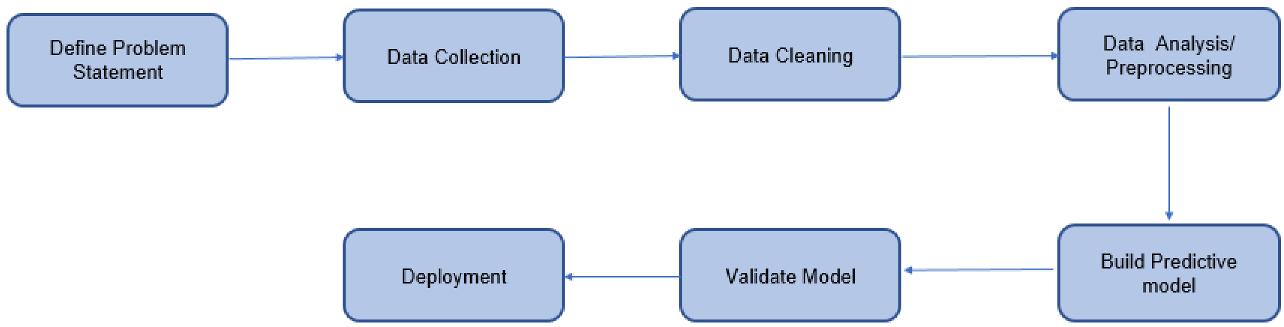

Figure 3.1: Predictive analysis process steps

Here, defining the problem statement and data collection are collectively called data exploration. The data needs to be cleaned in order to get rid of redundancies i.e. data cleaning. Then the modeling part is done which means after analyzing the data, a predictive model [59] is built followed by a performance analysis through model validation and deployment.

3.2 Bayesian Estimation

Bayesian estimation [84], [82], [8] uses prior data to estimate the value of an unknown parameter [82]. The main objective of Bayesian estimation is to estimate the underlying probability distribution of a random signal *X*(state) given noisy measurement data *Y*. This reduces the differences between the estimated value and the actual value of the parameter. Bayesian algorithm [62], [45] is a family of algorithms where all of them share a common principle i.e. every pair of features being classified is independent of each other.

Baye's theorem can be expressed as :

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{3.1}$$

where,

X: Random signal, *Y*: Noisy measurement data, *p(Y|X)* is the likelihood, *P(X)* is the prior distribution, *P(Y)* is the evidence.

Bayesian methods allow us to estimate the model parameter, construct a model forecast, and conduct model comparisons. In Bayesian signal processing [14], [80], the main area of interest is a stochastic process.

$$X_t = x(t)|t \in N \tag{3.2}$$

In these cases, generally, a Gauss-Markov chain [22] of 1^{st} order in the states is assumed, where the measurement is only dependent on the current state.So, A Gauss-Markov state-space model of the process.

$$x(t) = a[x(t-1)] + b[u(t-1)] + \omega(t-1) \tag{3.3}$$

and the measurement

$$y(t) = c[x(t)] + v(t) \tag{3.4}$$

where,

$\omega \sim N(0, R_{ww}(t-1))$, $v \sim N(0, R_{vv}(t-1))$ are the process and measurement noises, *u(t)* is the input and *a(.)*, *b(.)* and *c(.)* are some known functions.

1. *Analytic approach*: It is used to find the analytic solutions for the posterior and perform statistical inferences based on integrals.

$$E(f(x)) = \int f(x)P(X|Y)dx \tag{3.5}$$

2. *Monte carlo approach [11] [23]*: It is used to find an analytic or approximate solution for the posterior and perform statistical inferences based on sampling, which we going to use in our project.

$$E(f(x)) \approx \hat{f} = \frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{3.6}$$

The most common risk function used for Bayesian estimation is the *MSE* also called squared error risk. *MSE* is defined by,

$$MSE = E[(\hat{\theta}(x) - \theta)^2] \tag{3.7}$$

where the expectation is taken over the joint distribution of θ and x.

3.3 Stochastic Process

A stochastic process [78] [60] is a mathematical model that represents the evolution of a system or phenomenon over time in a probabilistic manner. It is an assemblage of random variables indexed by time or some other parameter. Stochastic processes are used in various fields, including mathematics, statistics, physics, finance, and engineering, to model and analyze systems with inherent uncertainty.

It can be written as,

$$\mu(B) = P(\{\omega \in \Omega : X(\omega) \in B\}). \tag{3.8}$$

The law of a stochastic process or a random variable(here ω) is also called the probability law, probability distribution, or distribution.

Formally, a stochastic process is defined as a collection of random variables $\{X(t) : t \in T\}$, where T represents the index set (usually time) and $X(t)$ represents the value of the random variable at time t. The index set can be discrete (e.g., integers or time steps) or continuous (e.g., real numbers). The random variables can be discrete (e.g., coin tosses) or continuous (e.g., stock prices)

Generally, there are three types of inferences that are of interest when considering state-space models [31]

1. Prediction: This includes forecasting subsequent values of the state.

2. Filtering: Estimating the current values of the state from the past and current observations is done in this step.

3. Smoothening: This step covers estimating the past values of the state given the observations.

Filtering and smoothening [56], [21] are comparable, but they are not the same. The best way to think of diversity is that with smoothening one can acknowledge what has happened with the states in the past given the present knowledge, whereas with filtering one can know what is the circumstances of that state right now.

A stochastic process, also known as a random process or random variable set, is a collection of random variables that are each organized using a built-in set of time-representative indices. It is used by traders to determine the performance of a portfolio of individual stocks using random probability distributions.

3.4 Long Short-Term Memory

LSTM [28] is a recurrent neural network (RNN) architecture commonly used for stock price prediction. LSTM networks are effective at capturing and learning from sequential data, making them suitable for time series forecasting tasks such as stock price prediction [74].

The application of LSTM in stock price prediction involves the following steps:

1. Data Preparation: Historical stock price data is preprocessed and formatted to suit the LSTM model's input requirements. This typically involves normalizing the data, partitioning it into input sequences (e.g., using a sliding window approach), and creating corresponding target sequences.

2. LSTM Model Architecture: The LSTM model is constructed using one or more LSTM layers followed by one or more fully connected (dense) layers. The LSTM layers capture temporal dependencies and patterns in the input sequences, while the dense layers map the learned features to the target variable, such as the future stock price.

3. Training: The LSTM model is trained using the prepared data. During training, the model adjusts its internal parameters through backpropagation and gradient descent optimization to minimize the difference between predicted and actual stock prices. MSE or a similar loss function is typically used.

4. Testing and Evaluation: The trained model is evaluated on a separate test dataset to assess its performance. Predicted stock prices are compared against the actual prices to calculate evaluation metrics like mean absolute error (MAE), RMSE, or directional accuracy.

5. Prediction: Once trained and evaluated, the LSTM model can be used to make future stock price predictions. Given a new input sequence, the model generates predictions for subsequent time steps.

To enhance LSTM's performance in stock price prediction, additional techniques can be employed. These include incorporating extra features (e.g., technical indicators or news sentiment) into the input data, tuning hyperparameters (e.g., number of LSTM layers, hidden units, learning rate), and using regularization techniques (e.g., dropout) to mitigate overfitting.

The schematic diagram of predicting the stock market using the LSTM model was taken from [63]:

It is important to note that while LSTM has shown promise in stock price prediction, accurate forecasting of stock prices remains challenging due to the inherent uncertainty and complexity of financial markets. Therefore, it is advisable to interpret predictions cautiously and consider them as probabilistic estimates rather than definitive outcomes.

Implementation result of stock price prediction model of NTPC using LSTM for which

RMSE is 3.6251

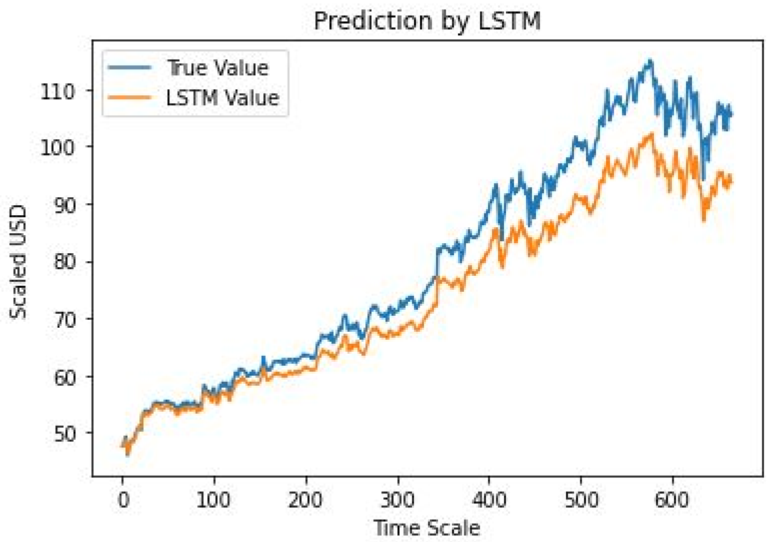

Figure 3.2: Stock price prediction model of NTPC using LSTM model

3.5 Linear regression

Predicting stock market prices using linear regression is a common approach in quantitative finance. While linear regression can provide a basic framework for modeling relationships between variables, it may not capture all the complexities and dynamics of the stock market. It is important to note that stock prices are influenced by numerous factors, including economic conditions, company performance, market sentiment, and geopolitical events, making accurate predictions challenging.

Let's assume we want to predict the closing price of a particular stock based on its historical data. Here's a step-by-step process using Python and the sci-kit-learn library:

1. Data Preparation: It includes gathering historical data for the stock, including the closing prices and any relevant features or indicators you want to consider as predictors. Split the data into training and testing sets.

2. Feature Selection: It includes choosing the features you believe might have a relationship with the stock price. For example, you could consider the opening price, trading volume, or the previous day's closing price.

3. Model Training: It includes fitting a linear regression model to the training data using the chosen features as predictors and the closing price as the target variable.

4. Model Evaluation: It includes evaluating the performance of the model using appropriate metrics such as MSE, RMSE, or R-squared value. Additionally, consider visualizing the predicted prices against the actual prices to gain insights

5. Prediction: It uses the trained model to make predictions on the testing data or new, unseen data.

It's important to note that linear regression assumes a linear relationship between the predictors and the target variable, which may not always hold in the stock market. Also, that stock price prediction is a complex task, and relying solely on linear regression may not yield highly accurate results.

The idea for the schematic diagram of predicting the stock market using a linear regression model was taken from the source (http://harry-nita.blogspot.com/2017/09/machine-learning-linear-regression.html)

Implementation result of stock price prediction model of NTPC using linear regression for which **RMSE is 5.0291**

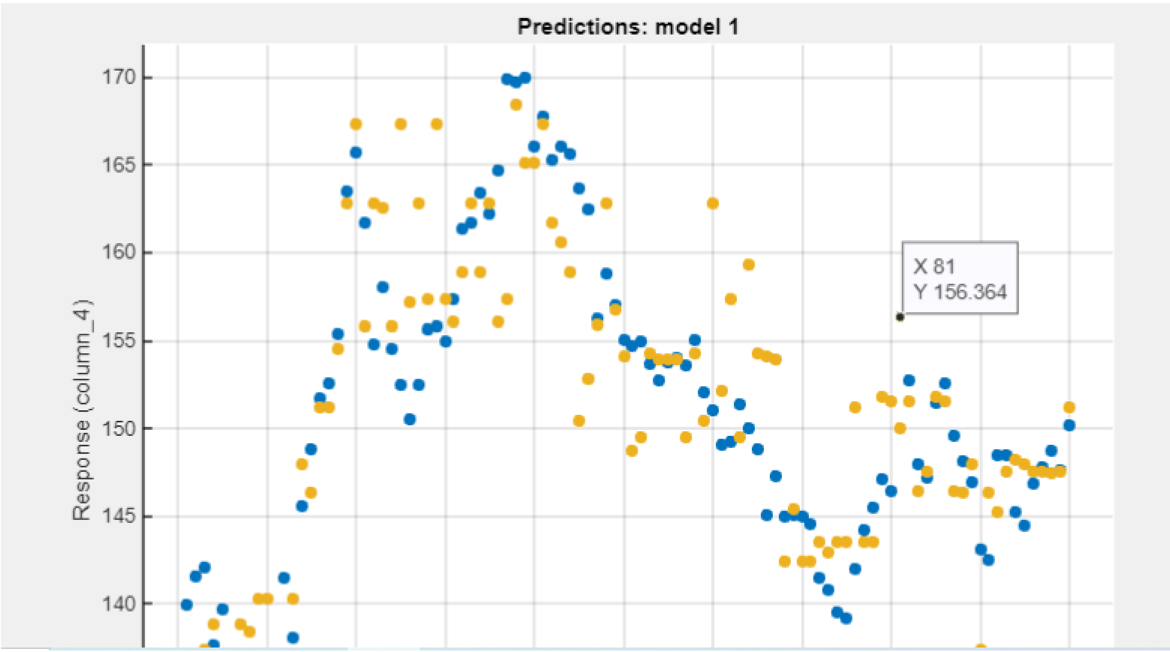

Figure 3.3: Stock price prediction model of NTPC using linear regression model

3.6 Dynamic Mode Decomposition

DMD is a data-driven technique used in stock market prediction to extract and analyze the underlying dynamics of a time series [52]. DMD is particularly useful for identifying dominant modes or patterns in the data and forecasting future behavior [43].

Here's an overview of the application of DMD in stock market price prediction:

1. Data preparation: Historical stock market data is preprocessed and organized into a suitable format for DMD analysis. This typically involves formatting the data as a matrix, where each row represents a snapshot in time and each column represents a variable (e.g. stock prices or market indices).

2. DMD: It decomposes the data matrix into a set of dynamic modes and associated temporal behaviors. The DMD algorithm computes the eigenvalues and eigenvectors of the data matrix and constructs a low-rank approximation of the system's dynamics. These dynamic modes capture the underlying oscillations or patterns present in the data.

3. Mode selection and analysis: From the DMD analysis, the dominant modes are identified based on the magnitude of their associated eigenvalues. These modes represent the most influential patterns in the stock market data. The associated eigenvectors provide insights into the structure and importance of these modes.

4. Forecasting: Once the dominant modes are identified, they can be used to forecast future behavior. By projecting the modes forward in time, future stock market trends or patterns can be predicted. This projection can be performed using the eigenvalues and eigenvectors obtained from the DMD analysis.

It's important to note that DMD is a purely data-driven technique and does not incorporate external factors or fundamental analysis. Therefore, it may not capture complex relationships or causal factors that impact stock market behavior. DMD-based predictions should be interpreted with caution and considered alongside other relevant information and analysis methods.

DMD has its advantages in providing a low-dimensional representation of the dynamics and its ability to capture non-linearity and transients. However, its effectiveness in stock market prediction may vary depending on the specific characteristics and dynamics of the stock market data being analyzed.

From measurements and computation of a given system in time, the DMD method provides a decomposition of data into a set of dynamic and robust modes that are derived. To illustrate the algorithm, we consider regularly spaced sampling in time.

The data collection process involves two parameters:

N = number of companies in a given portfolio, M = number of data snapshots taken

data collection times :

$$t_{m+1} = t_m + \Delta t \tag{3.9}$$

Data Snapshots :

$$X = \begin{bmatrix} x(t_1) & x(t_2) & x(t_3) & . & . & . & x(t_M) \end{bmatrix} \tag{3.10}$$

where, x represents the snapshots,and X is signal values at i^{th} time instant

For the purposes of the DMD method, the following matrix includes columns j through k of the original data matrix.

$$X_j^k = \begin{bmatrix} [x(t_j) & x(t_{j+1}) & . & . & . & x(t_k)] \end{bmatrix} \tag{3.11}$$

By splitting the data matrix into,

$$X_1^{M-1} = \begin{bmatrix} [x_1 & x_2 & x_3 & . & . & . & x_{M-1}] \end{bmatrix} \tag{3.12}$$

$$X_2^{M-1} = \begin{bmatrix} [x_1 & Ax_1 & A^2x_1 & . & . & . & A^{M-2}x_1] \end{bmatrix} \tag{3.13}$$

where X_2^{M-1} is one time slot shifted from X_1^{M-1}

In the DMD technique, the final data point X_M is represented

$$x_M = \sum_{m=1}^{M-1} b_m x_m + r \tag{3.14}$$

where r is the residual vector.The steps involved in DMD of the time series x_t is as following,

Computing SVD decomposition of X_1^{M-1}

$$X_1^{M-1} = U\Sigma V^* \tag{3.15}$$

constructing the matrix A ,Koopman operator

$$AX_1^{M-1} \approx X_2^M \tag{3.16}$$

$$X_2^M = X_1^{M-1}S + re_{M-1}^* \tag{3.17}$$

where S is the companion matrix. & e_{M-1} is $(m-1)^{th}$ unit vector. Hence,

$$AX_1 = X_2 \approx X_1 S \tag{3.18}$$

The Matrix S can be computed and it's eigen values and eigen vectors found, using these eigenvectors, the state of the system can be reconstructed in decomposed form.

3.6.1 Stock price prediction using DMD

A dynamic system is described using a governing set of differential equations:

$$\frac{\partial f}{\partial x} = F(x,t) \tag{3.19}$$

Where F is an unknown non-linear function. At each state, we can make different kinds of measurements of the observables. The measurement function can be denoted as:

$G(x,t_k) = 0$ Where, k = 0,1,2,......,M (measurement time)

The initial condition is stated as, $x(0) = x_0$.In the DMD procedure, approximate linear evolution of the system :

$$\frac{\partial \tilde{x}}{\partial t} = A\tilde{x} \tag{3.20}$$

So, the solution of the above equation :

$$\tilde{x} = \sum_{k=1}^{L} b_k \Psi_k \exp(\omega_k t) \tag{3.21}$$

where, Ψ_k and ω_k are eigenvectors and eigenvalues of matrix A. And $\tilde{x}(t)$ defines the state of the system at time t. Stock prices had been taken to represent the state of the system, so $x(t)$ gives the stock price at time t.

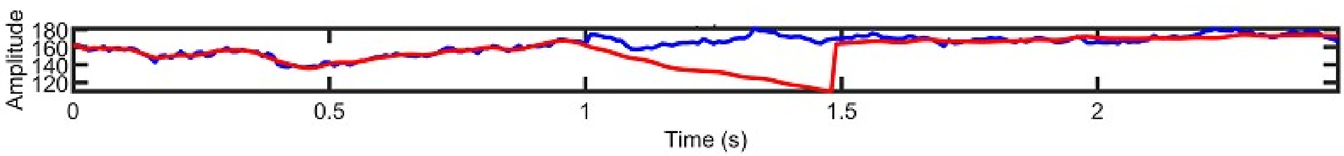

Figure 3.4: Stock price prediction of NTPC using DMD

Implementation result of Stock price prediction of NTPC using DMD using is shown for which **RMSE is 17.3625**, where the blue line shows the true value and the red line is the predicted value.

3.7 Kalman Filter

Kalman filter is a recursive mathematical algorithm used to estimate the state of a dynamic system from a series of noisy measurements. It is widely applied in various fields, including engineering, economics, robotics, and finance.

The Kalman filter operates by combining sensor measurements and mathematical model assumptions to get the best approximation of the system's true state. It is based on a probabilistic model of the dynamics of the system. It considers both the current measurement and the previously estimated state to update the state estimate.

Here's a high-level overview of the Kalman filter algorithm:

1. Initialization: The filter is initialized with an initial state estimate and covariance matrix. These represent the best knowledge or prior information about the system's state.

2. Prediction: The Kalman filter predicts the next state based on the dynamics model of the system. It uses the previously estimated state and covariance matrix, along with the known system dynamics equations, to make a prediction.

3. Update: The filter incorporates new measurements or observations into the state estimate. It compares the predicted measurement from the predicted state with the actual measurement from the sensor. The difference between the predicted and measured values, along with their associated uncertainties, is used to update the state estimate and covariance matrix.

4. Iteration: The prediction and update steps are repeated recursively as new measurements become available. Each iteration refines the state estimate based on the latest measurements, continually improving the accuracy and reliability of the estimated state.

The KF is designed to handle systems that exhibit linear dynamics and Gaussian noise characteristics. However, variations such as the extended Kalman filter (EKF), cubature Klaman filter (CKF), and unscented Kalman filter (UKF) [72] have been developed to handle non-linear dynamics and non-Gaussian noise scenarios.

In finance, the KF is commonly used for applications like portfolio optimization, asset allocation, and estimation of hidden factors or variables in financial models. It helps in extracting valuable information from noisy and incomplete market data to make more accurate predictions or decisions.

It's worth noting that the effectiveness of the KF in financial applications depends on the underlying assumptions and the quality of the available data. Careful consideration of model selection, parameter tuning, and understanding the limitations of the filter is essential for obtaining meaningful results in financial modeling and prediction.

The idea of the block diagram for the KF processing steps is necessary to generate the best state estimate from the observations, as well as the underlying models taken from [75].

Iteratively producing the best assessments of the state vector X_k at distinct time t_k where k = 1,2,..,n based on observations y_j at distant time t_j where j = 1,2,..,k is the underlying notion behind the KF. The Kalman filter's key benefit over other recurrence filtering techniques is that the guess it generates is optimal in the sense that they are linear, unbiased, and have the least amount of variance. In the follow-up, the derivation is guided by these three distinctive and varied properties of the Kalman filter.

1. Linear Estimator: The best appraisal of the state vector at time t_k is constructed as a linear combination of $\tilde{x}_k$ and y_k given the nearest prediction $\tilde{x}_k$ of x_k and the most recent observation at time t_k.

$$\tilde{x}_k = L_k\tilde{x}_k + K_k y_k \tag{3.22}$$

   where L_k and K_k are matrices that are properly specified. The well-known Kalman gain matrix defines everything that the matrix K_k is, as will be shown subsequently.

2. Unbiased estimator: The statistical supposition of the variable or parameter being estimated must be identical to its true value in order for an estimator to be considered impartial. Therefore, the following definition of a state error vector, $e_{k|j}$, is appropriate:

$$e_{k|j} = x_{k|j} - x_k \tag{3.23}$$

   The estimation error is denoted by the error vector $e_{k|k}$ = $\hat{x}_k$ when j = k, and when j = k-1, the error vector is $e_{k|k-1}$ = $\tilde{e}_k$ is assigned to as the one-step prediction error, often abbreviated to just forecasting errors. Using the linear figurer or estimator, the error $\hat{e}_k$ in the best rate and estimate $\hat{X}_k$ of the state vector x_k is deduced to be [75]

$$\hat{e}_k = \hat{x}_k - x_k = L_k\tilde{x}_k + K_k y_k - x_k = L_k(\tilde{e}_k + x_k) + K_k y_k - x_k \tag{3.24}$$

   The yields obtained after simplifying the result by substituting for the observation y_k from the equation :

$$\hat{e}_k = L_k(\tilde{e}_k + x_k) + K_k(H_k x_k + v_k) - x_k = L_k\tilde{e}_k + (L_k + K_k(H_k x_k - I)x_k + K_k v_k) \tag{3.25}$$

   where I is an $N * N$ identity matrix. Now, since the anticipated value of the observation noise v_k is considered to be zero, and since for an impartial estimator of the estimated and anticipated state errors ($\hat{e}_k$ and $\tilde{e}_k$ respectively) must have an expectation of zero, then the matrices L_k and K_k must be chosen so that

$$L_k + K_k H_k - I = 0. \tag{3.26}$$

Therefore, the matrix must be selected as follows for a linear unbiased estimator:

$$L_k = I - K_k \tag{3.27}$$

H_k, and after doing this, the estimator equations and estimation error equations become

Estimation error

$$\tilde{e}_k = (1 - K_k H_k)\tilde{e}_k + K_k v_k \tag{3.28}$$

Estimate

$$\tilde{x}_k = (1 - KkH_k)\tilde{x}_k + K_k y_k = \tilde{x}_k + K_k(y_k - H_k\tilde{k}_k) = \tilde{x}_k + K_k z_k \tag{3.29}$$

where, **Innovation**

$$z_k = y_k + H\tilde{x}_k \tag{3.30}$$

Innovation refers to the new information or particulars that results from a new observation, and the term z_k is used to describe this. [75] Standard usage for the KF is the label "z".

3. <u>Minimum variance estimator</u>: The estimation error $\hat{e}_k$ has a low variance because the Kalman gain matrix K_k at time instant t_k is selected accordingly. The prediction error and estimation error covariance matrices are defined as follows to demonstrate how this can be done:

Estimation error covariance

$$\hat{P}_k = E(\hat{e}_k \hat{e}_k^T) \tag{3.31}$$

Prediction error covariance

$$\tilde{P}_k = E(\tilde{e}_k \tilde{e}_k^T) \tag{3.32}$$

After substituting for the calculated error vector obtained in the preceding subsection, the estimation error covariance can be symbolized in terms of the prediction error covariance:

$$\hat{P}_k = E[\{(I - K_k H_k)\tilde{e}_k + K_k v_k\}\{(I - K_k H_k)\tilde{e}_k + K_k v_k\}^T] \tag{3.33}$$

$$\hat{P}_k = E[(I - K_k H_k)\tilde{e}_k \tilde{e}_k^T (I - K_k H_k)^T + K_k v_k v_k^T K_k^T] + E[(I - K_k H_k)\tilde{e}_k v_k^T K_k^T + K_k v_k \tilde{e}_k^T (I - K_k H_k)^T] \tag{3.34}$$

Firstly, consider the second assumption in order to simplify this formula. The forecasting error $\tilde{e}_k$ is based on received facts up to time t_{k-1}, whereas v_k is the inspection error associated with measurement and calculation made at time t_k. From the supposition that the state vector noise w_k and the observation vector noise v_k are independent, spectrally white, and zero means, the supposition of the product will be zero, and hence the second assumption in the last equation is zero. The first presumption can be expressed in terms of formerly expressed covariance matrices to give [75]

$$\hat{P}_k = E[\{(I - K_k H_k)\tilde{P}_k (I - K_k H_k)^T + K_k R_k K_k^T] \tag{3.35}$$

The variances of the various parts of the estimate error vector e_k must meet the specified minimal variance requirement. It will be sufficient to select the Kalman gain K_k to minimize the sum of these variances because variances are unavoidably non-negative. As a result, their sum is just the trace of the matrix P_k, which is represented as $T_r(\hat{P}_k)$. The components on the main diagonal of the presumption error covariance matrix P_k are now exactly the variances of the various components of the estimation error e_k.

4. Prediction error covariance: The state at time t_{k+1} can be best predicted from the state equation. is given by

$$\tilde{x}_{k+1} = \Phi_k \hat{x}_k \tag{3.36}$$

and the corresponding prediction error is indicated by

$$\tilde{e}_{k+1} = \Phi_k \hat{x}_k - x_{k+1} = \Phi_k (\hat{e}_k + x_k) - x_{k+1} \tag{3.37}$$

$$\tilde{e}_{k+1} = \Phi_k \hat{e}_k + \Phi_k x_k - x_{k+1} = \Phi_k \hat{e}_k - w_k \tag{3.38}$$

Consequently, the prediction error covariance may be determined using

$$\tilde{P}_{k+1} = E[\tilde{e}_{k+1}\tilde{e}_{k+1}^T] = E[(\Phi_k \hat{e}_k - w_k)(\Phi_k \hat{e}_k - w_k)^T] \tag{3.39}$$

$$\tilde{P}_{k+1} = \Phi_k E[\hat{e}_k \hat{e}_k^T]\Phi_k^T + E[w_k w_k^T] - \Phi_k E[\hat{e}_k w_k^T] - E[w_k \hat{e}_k^T]\Phi_k^T \tag{3.40}$$

$$\tilde{P}_{k+1} = \Phi_k \hat{P}_k \Phi_k^T + Q_k \tag{3.41}$$

In acquiring the last step, use was made of the actuality that $\hat{e}_k$ and w_k are not varying together i.e. uncorrelated. That this is so is easily revealed, since $\hat{e}_k$ is the presumption error for x_k at time

t_k whereas w_k is the noise associated with the state at the later time of t_{k+1}. The last equation provides a way to calculate $\tilde{P}_{k+1}$ recursively, and this adequately completes the set of equations required to calculate the Kalman gain repeatedly, and hence to process the observations y_k to produce appraisal $\hat{x}_k$ of the state vector x_k which are linear, unbiased and have least variance.

3.7.1 Linear Kalman Filter

3.7.1.1 Simple Kalman Filter

The SKF is a variant of the LKF algorithm that is specifically designed for systems with linear dynamics and Gaussian noise. It is a recursive and repeated estimator that combines predictions and forecasting from a linear system model with measurements to provide an optimal or ideal estimate of the true state of the system [75].

Here's an overview of the LKF algorithm:

1. Initialization: The filter is initialized with an initial estimate of the state vector and the corresponding covariance matrix. These represent the prior knowledge or best guess of the system's initial state.

2. Prediction: The filter predicts the next state of the system based on the linear system dynamics model. The prediction is made using the state transition matrix, which describes how the state evolves over time, and the control input, if applicable. The covariance matrix of the prediction is also updated using the process noise covariance matrix, which captures the uncertainty in the system dynamics.

3. Update: The filter incorporates measurements from sensors or observations to update the state estimate. It compares the predicted measurement, obtained by multiplying the state prediction by the measurement matrix, with the actual measured value. The measurement noise covariance matrix captures the uncertainty in the measurements. The Kalman gain, computed using the prediction and measurement covariance matrices, determines the weight given to the prediction and measurement in the update step. The state estimate and its covariance matrix are updated based on the Kalman gain and the measurement residual.

4. Iteration: The prediction and update steps are repeated recursively as new measurements become available. Each iteration refines the state estimate based on the latest measurements, leading to an improved estimate of the true state of the system.

Given the linear Gaussian assumptions, the LKF provides an optimal estimate of the state in terms of minimizing the mean squared error between the estimated state and the true state. It is widely used in various applications such as navigation, tracking, and control systems.

It's important to note that the LKF assumes linearity and Gaussian noise characteristics, which may not always hold in real-world systems. In cases where non-linearities or non-Gaussian noise exist, extensions such as the EKF or UKF [72] can be used to handle these scenarios.

Application of LKF [75]

Using multiple standard deviation sigma values for target acceleration, a computer simulation was carried out for tracking a target subjected to a known acceleration.

The following circumstances resulted in the generation of a one-dimensional target motion [75]:

- remaining still for 5 seconds between $t = 0$ and $t = 5$ seconds
- From $t = 5$ through $t = 15$, there is a steady acceleration of 10 m/s^2 for 10 seconds.
- From $t = 15$ to $t = 30$, the speed remains steady for a further 15 seconds.

At intervals of 10 ms, or T=0.01 s, simulated measurements of the target's position and speed were created. The normal distribution of measurement errors was presumed, in addition to independent samples and a mean of 0. We presumed that the standard error for location measurements was 100 m and the standard error for velocity was 4 m/s.

The measurements were passed on to a KF, which used the simulated measurements to estimate the target motion. The initially anticipated covariance matrix was somewhat at random set to be $X_0 = (0,0,0)T$, and the filter state was initialized to that value.

The findings from the simulation can be seen in the image using two distinct values for the desired acceleration's standard deviation, σ.

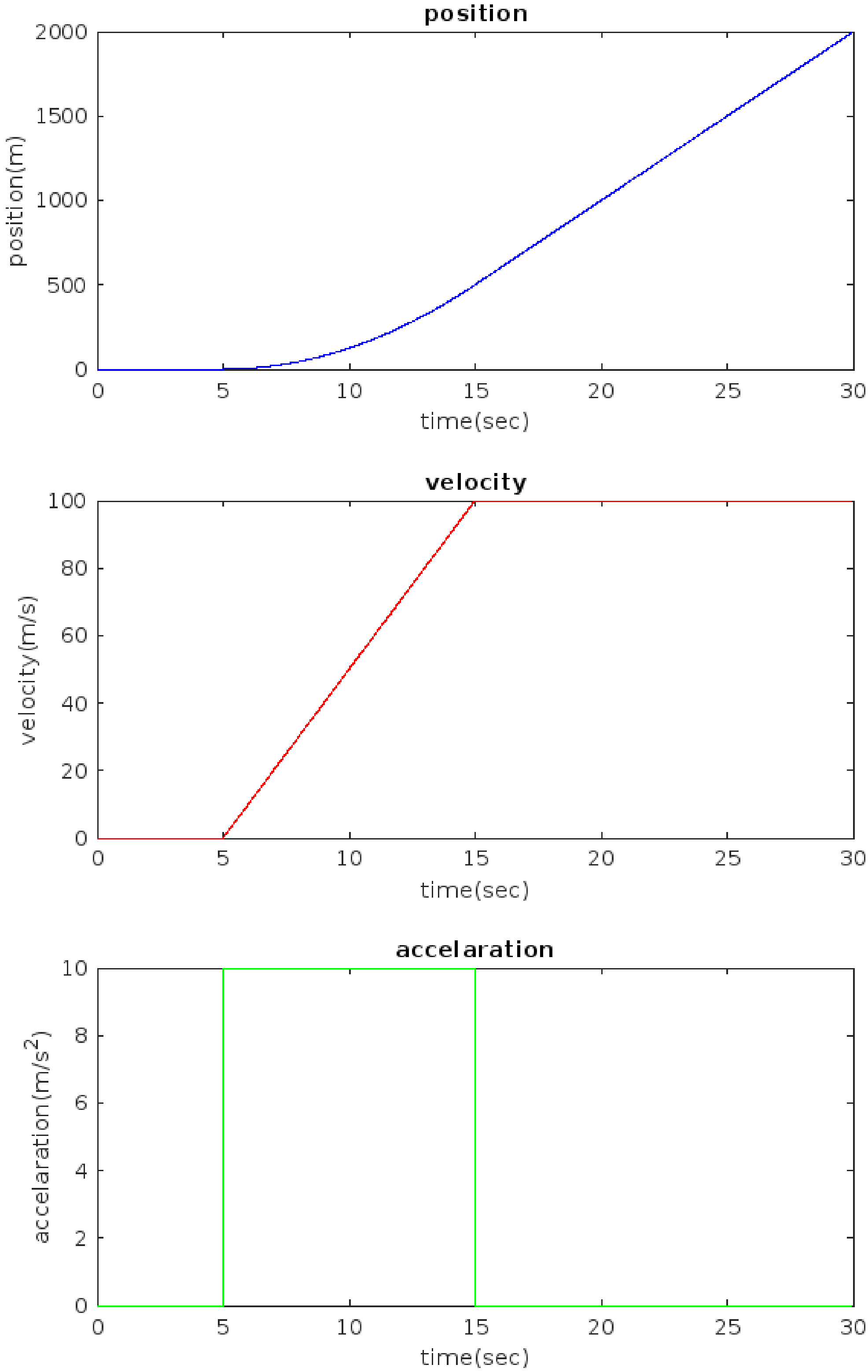

Figure 3.5: True value of position, velocity, and acceleration of target

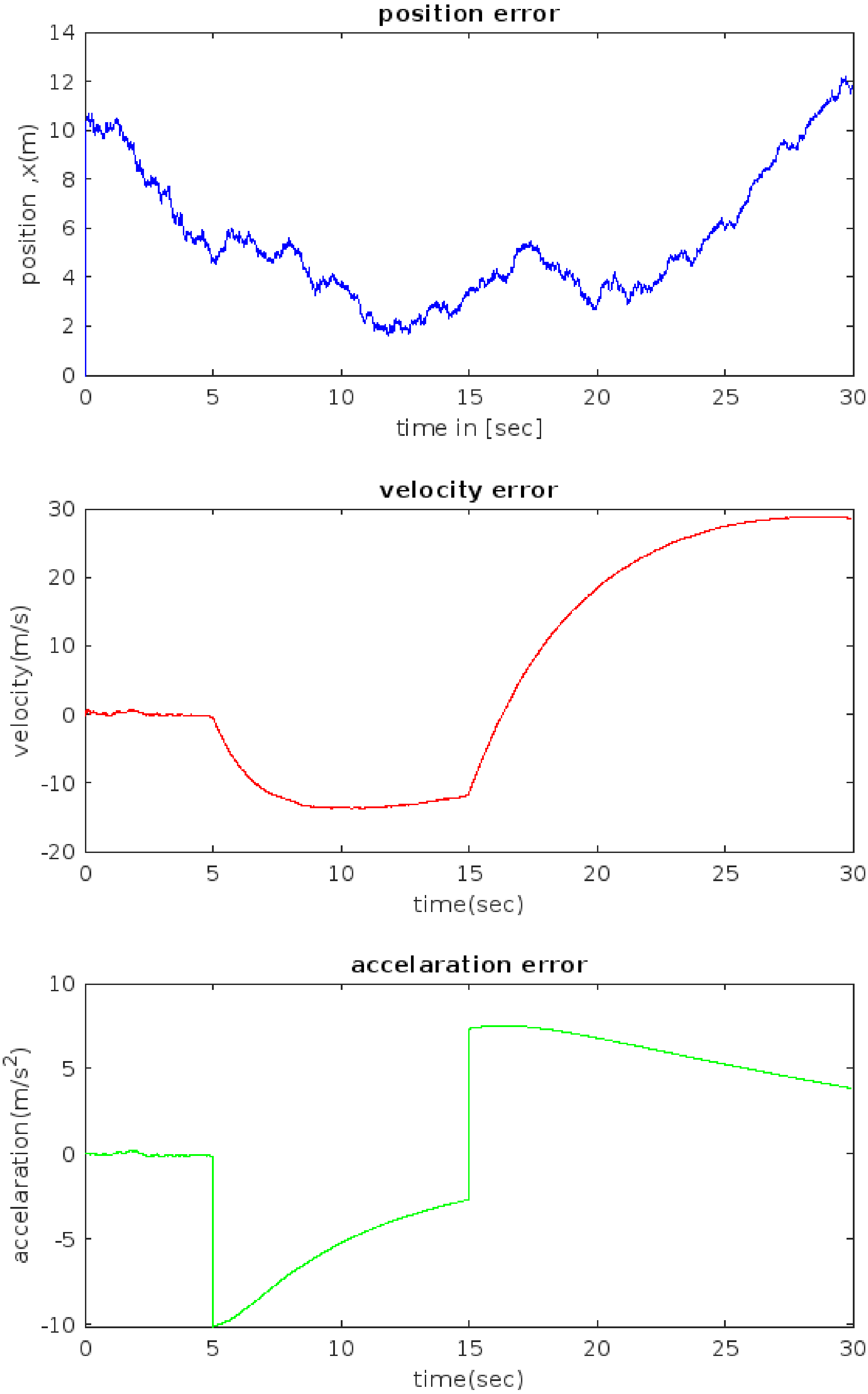

Figure 3.6: Predicted value of target using linear Kalman filter for $\sigma = 0.003\ m/s^2$

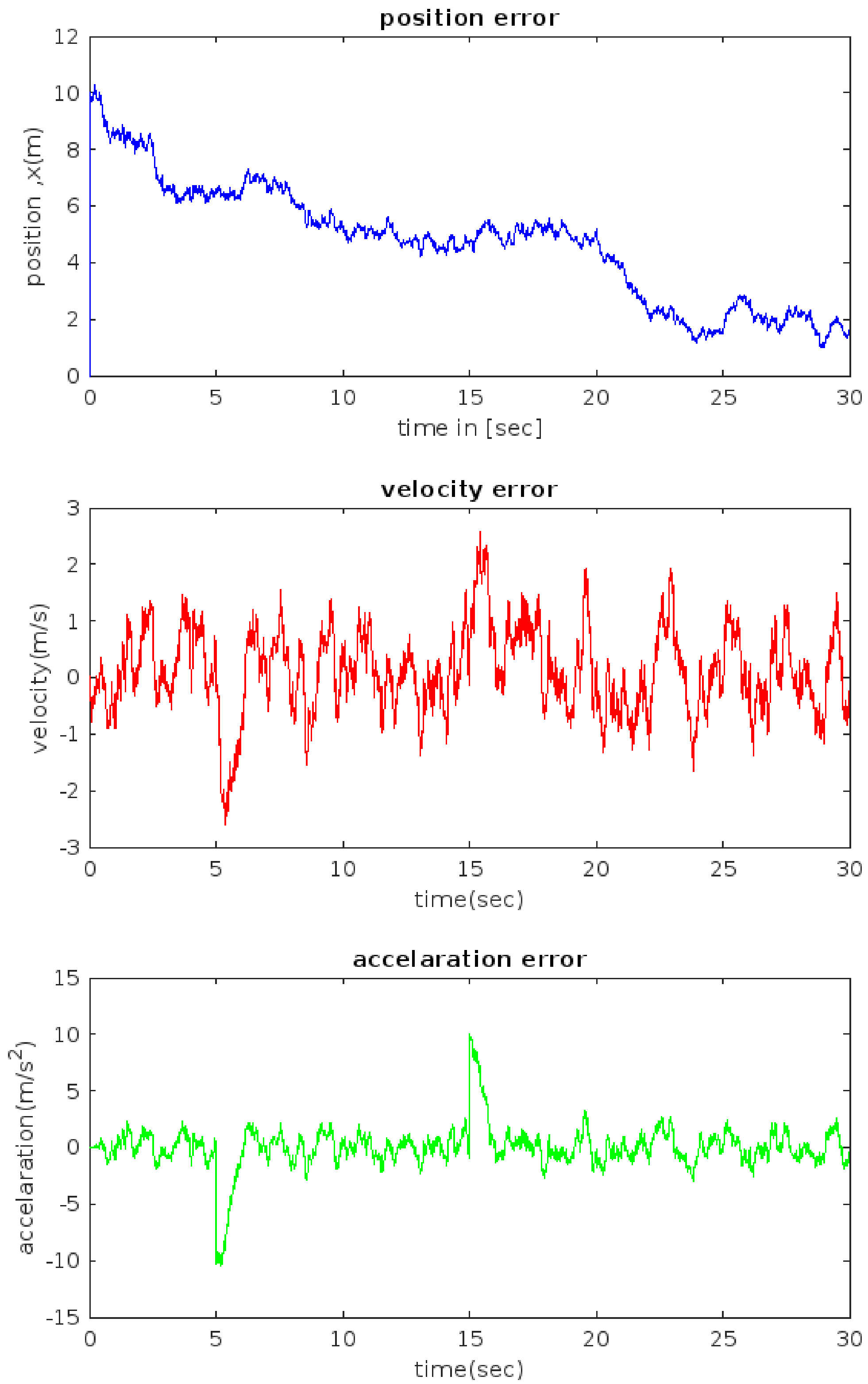

Figure 3.7: Predicted value of target using linear Kalman filter for $\sigma = 3\ m/s^2$

The target's simulated position, velocity, and acceleration are shown in the first figure. The amount of error in the KF estimate of these parameters is depicted in the second figure when the standard deviation of the target's acceleration is assumed to be sigma $(\sigma) = 0.03\ m/s^2$, and the same quantities are shown in the third figure if the value of sigma (σ) used is $3\ m/s^2$.

As can be seen, a lower sigma value produces an increased presumption error than a higher sigma value does. Furthermore, for the lower amount of sigma, the errors are smoother than they are for the higher value of sigma(σ).

The tracking error is significantly lower than the measurement error, despite the KF failing to correctly reflect the target's deterministic acceleration.

3.7.2 Non-Linear Kalman Filter

3.7.2.1 Extended Kalaman Filter

The EKF is an extension of the KF algorithm that is designed to handle systems with non-linear dynamics and non-Gaussian noise. It is a recursive and repetitive estimation algorithm that combines predictions from a non-linear system model with measurements to provide an optimal or ideal estimate of the true state of the system.

Here's an overview of the EKF algorithm:

1. Initialization: Similar to the KF, the EKF is initialized with an initial estimate of the state vector and the corresponding covariance matrix.

2. Prediction: The EKF predicts the next state of the system based on a non-linear system dynamics model. The prediction is made by applying a non-linear function to the previous state estimate and incorporating the process noise covariance matrix. The Jacobian matrix of the non-linear function is used to linearize the system dynamics around the current state estimate, allowing for a linear prediction step. The covariance matrix of the prediction is also updated using the linearized dynamics.

3. Update: The filter incorporates measurements from sensors or observations to update the state estimate. Similar to the KF, the EKF compares the predicted measurement, obtained by applying a non-linear function to the predicted state, with the actual measured value. The Jacobian matrix of the non-linear function is used to linearize the measurement function around the predicted state, enabling a linear update step. The measurement noise covariance matrix is also used in the update step. The Kalman gain is computed based on the linearized dynamics and measurement functions to determine the weight given to the prediction and measurement in the

update step. The state estimate and its covariance matrix are updated using the Kalman gain and the measurement residual.

4. Iteration: The prediction and update steps are repeated recursively as new measurements become available. Each iteration refines the state estimate based on the latest measurements, leading to an improved estimate of the true state of the system.

Summary of EKF [73]

Model and observation:

$$x_k = f(x_{k-1}) + w_{k-1} \tag{3.42}$$

$$z_k = h(x_k) + v_k \tag{3.43}$$

Initialization:

$x_0^a = \mu_0$ with error covariance P_0

Model forecast step/predictor:

$$x_k^f \approx f(x_{k-1}^a) \tag{3.44}$$

$$P_k^f = J_f(x_{k-1}^a) P_{k-1} J_f^T(x_{k-1}^a) + Q_{k-1} \tag{3.45}$$

Data assimilation step/corrector:

$$x_k^a \approx x_k^f + K_k(z_k - h(x_k^f)) \tag{3.46}$$

$$K_k = P_k^f J_h^T(x_k^f)(J_h(x_k^f) P_k^f J_h^T(x_k^f) + R_k)^{-1} \tag{3.47}$$

$$P_k = (1 - K_k J_h(x_k^f)) P_k^f \tag{3.48}$$

The EKF is a widely used technique for state estimation in non-linear systems. However, it has limitations, such as the accuracy of the linearization process and the assumption of Gaussian noise. If the non-linearities in the system are significant, or the noise is non-Gaussian, alternative approaches like the CKF, UKF, or particle filter may be more suitable.

It's important to note that the success of the EKF depends on the choice of the non-linear system model and the accuracy of the linearization process. Careful consideration of these factors and understanding the limitations of the EKF are crucial for obtaining reliable estimates in practical applications.

A complete picture of the operation of the extended KF can be understood from the source [73]

Application of Extended Kalman Filter

Below is the simulation result of Target tracking using the EKF. Here, is the Time history of estimation results for 1^{st} order non-linear dynamics of a vehicle. position error and velocity error of the true and estimated value is shown where the EKF proves to be successful in tracking the position of the target while the error for the velocity of the vehicle between the true and estimated value is increased for some portion of the graph due to high non-linearities.

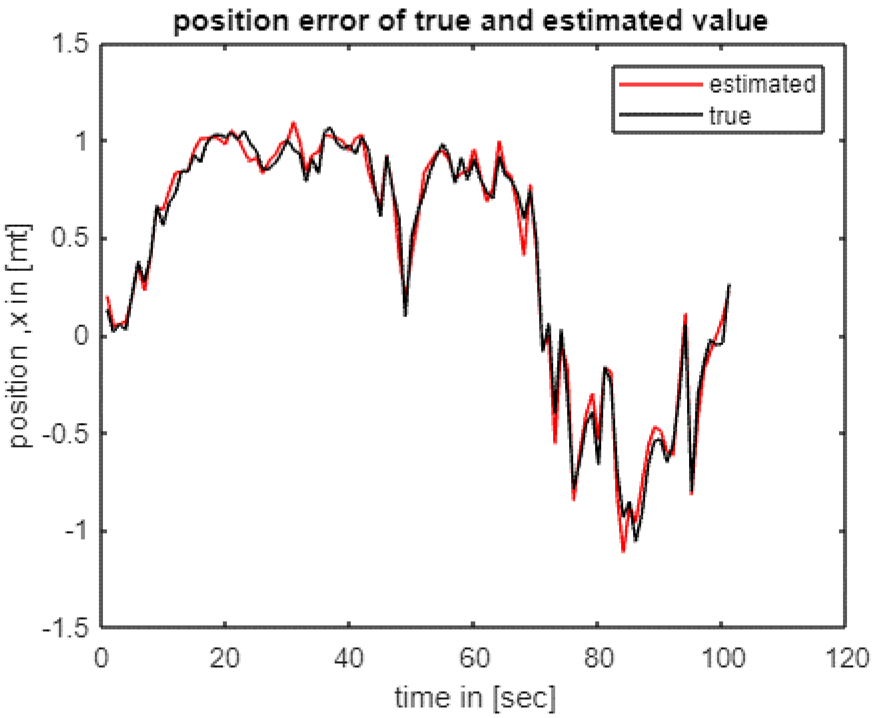

Figure 3.8: Position error of true and estimated value using EKF

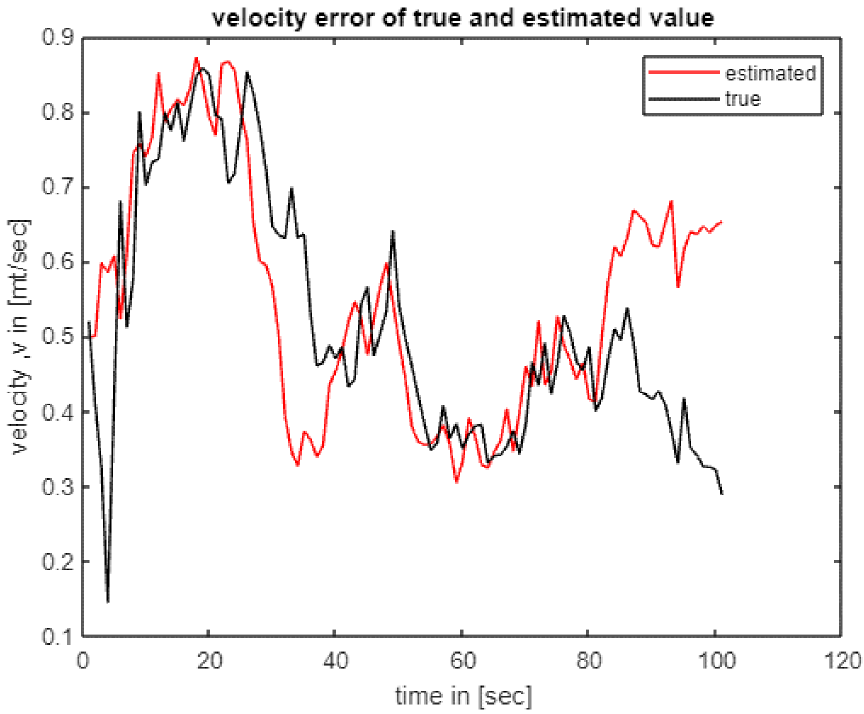

Figure 3.9: Velocity error of true and estimated value using EKF

3.7.2.2 Cubature Kalman Filter

The CKF is a variant of the KF algorithm that approximates the propagation of probability distributions in non-linear systems without explicitly linearizing the system dynamics. It is a recursive estimation algorithm that combines predictions from a non-linear system model with measurements to provide an optimal estimate of the true state of the system.

It is a nonlinear filter used for high-dimensional state estimation. A set of cubature points scaling linearly with the state-vector dimension is offered by this filter's third-degree spherical-radial cubature rule [4].

Process equation :

$$X_k = f(X_{k-1}, U_{k-1}) + V_{k-1} \tag{3.49}$$

Measurement equation :

$$Z_k = h(X_k, U_k) + W_k \tag{3.50}$$

where, X_k is the state of the dynamic system at a discrete time, f and h are some known functions, $\{V_{k-1}\}$ and W_k are independent processes and measurements of Gaussian noise [4].

Here's an overview of the CKF algorithm:

1. Initialization: Similar to other KF variants, the CKF is initialized with an initial estimate of the state vector and the corresponding covariance matrix.

2. Cubature Points Generation: The CKF generates a set of cubature points that capture the statistics of the current state estimate and covariance matrix. Cubature points are derived by considering a sigma point transformation of the Gaussian distribution associated with the current state estimate.

3. Prediction: The CKF propagates the cubature points through the non-linear system dynamics model to obtain predicted cubature points for the next time step. The prediction step is performed by applying the non-linear function to each cubature point and incorporating the process noise covariance matrix.

$$p(x_k|D_{k-1}) = \int_{R^{n_x}} p(x_k, x_{k-1}|D_{k-1})dx_{k-1} \tag{3.51}$$

$$p(x_k|D_{k-1}) = \int_{R^{n_x}} p(x_{k-1}|D_{k-1})p(x_k|x_{k-1}, u_{k-1})dx_{k-1} \tag{3.52}$$

4. Covariance Estimation: The CKF estimates the covariance of the predicted cubature points and computes the predicted state estimate and its covariance matrix based on the propagated points.

5. Measurement Update: The CKF compares the predicted cubature points, obtained from the predicted state estimate, with the actual measured value. The measurement update step is performed by applying the non-linear measurement function to each predicted cubature point. The measurement noise covariance matrix is also incorporated. The weighted sum of the transformed points provides an updated state estimate and covariance matrix.

$$p(x_k|D_k) = p(x_k|D_{k-1}, u_k, z_k) \tag{3.53}$$

$$p(x_k|D_k) = \frac{1}{c_k} p(x_k|D_{k-1}, u_k)p(z_k|x_k, u_k) \tag{3.54}$$

6. Iteration: The prediction and update steps are repeated recursively as new measurements become available. Each iteration refines the state estimate based on the latest measurements, leading to an improved estimate of the true state of the system.

$$p(u_k|D_{k-1},x_k) = p(u|D_{k-1}) \tag{3.55}$$

The CKF provides an approximation of the true probability distribution of the system state without explicitly linearizing the system dynamics, making it suitable for non-linear systems. It offers improved accuracy over linearization-based approaches like the EKF and can handle a wider range of non-linearities.

However, the CKF may require a higher computational cost compared to other KF variants, as it involves the propagation of multiple and distinct cubature points through the non-linear system model.

It's important to note that the success of the CKF depends on the choice of the non-linear system model and the accuracy of the cubature points generation process. Careful consideration of these factors and understanding the limitations of the CKF are crucial for obtaining reliable estimates in practical applications.

Application of CKF [4]:

An airplane performs maneuvering and steers turn in a horizontal plane at a steady and continuous but indeterminate turn rate. This is an example of a standard air traffic control structure Ω. A non-linear process equation is used to model the kinematics of the turning motion. Using the CKF where the state of the aircraft is given by the equation

$$x = \begin{bmatrix} \xi_k & \xi_k' & \eta_k & \eta_k' & \Omega_k \end{bmatrix} \tag{3.56}$$

where ξ_k and η_k denote positions, and ξ_k' and η_k' represent velocities in the x and y directions, respectively, T is the time interval between two sequential measurements , position and velocity estimates in both the x and y directions have been shown where the non-linear process equation is as follows :

$$x_k = \begin{pmatrix} 1 & \frac{sin\Omega T}{\Omega} & 0 & -(\frac{1-cos\Omega T}{\Omega}) & 0 \\ 0 & cos\Omega T & 0 & -sin\Omega T & 0 \\ 0 & (\frac{1-cos\Omega T}{\Omega}) & 1 & \frac{sin\Omega T}{\Omega} & 0 \\ 0 & sin\Omega T & 0 & cos\Omega T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} x_{k-1} + v_k \tag{3.57}$$

True initial state :

$$x_0 = \begin{bmatrix} 1000m & 300ms^{-1} & 1000m & 0ms^{-1} & -3^o s^{-1} \end{bmatrix}^T \tag{3.58}$$

and the associated covariance matrix:

$$P_{0/0} = \begin{bmatrix} 100m^2 & 10m^2s^{-2} & 1000m^2 & 10m^2s^{-2} & 100mrad^2s^{-2} \end{bmatrix}^T \tag{3.59}$$

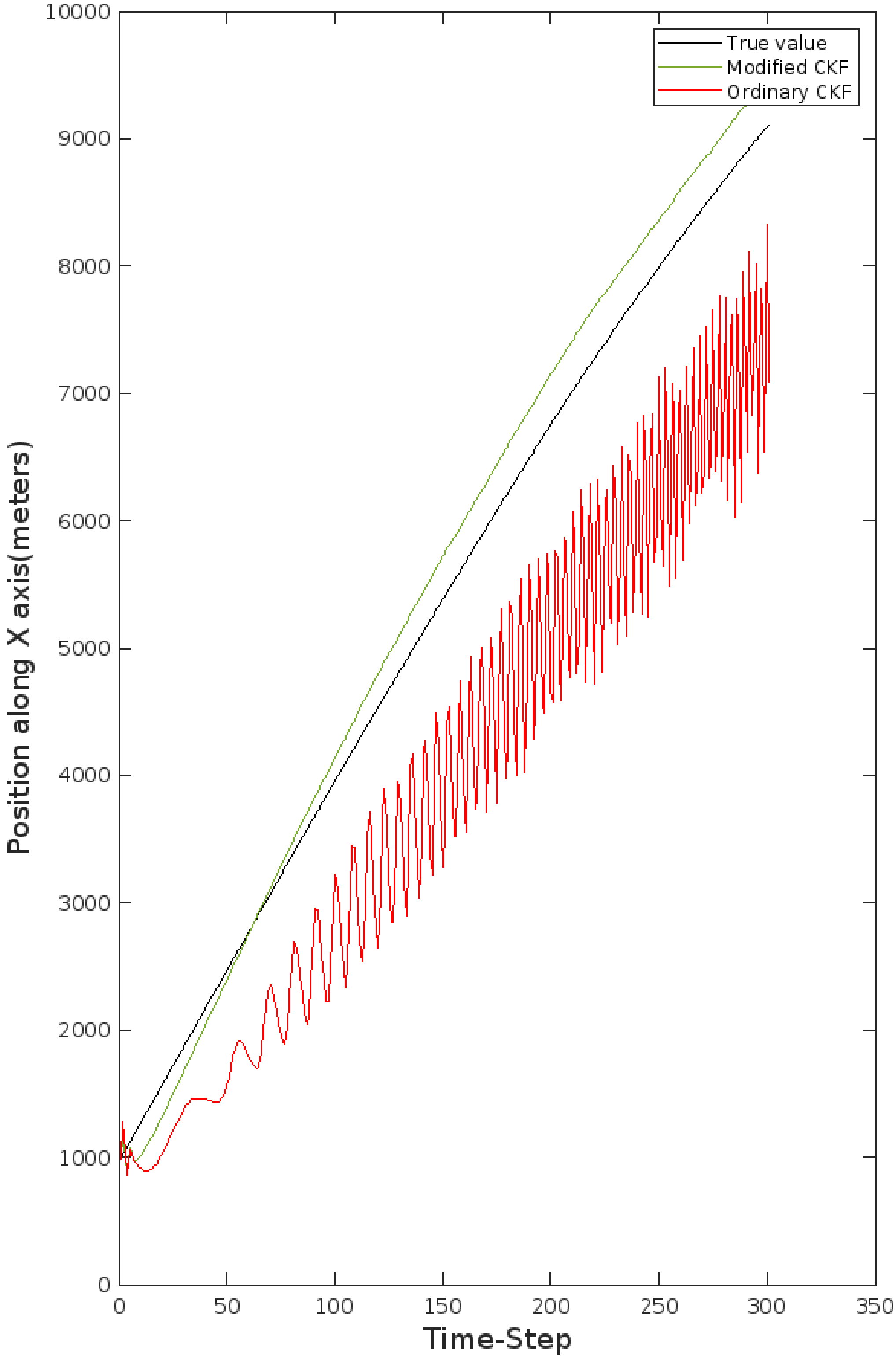

Figure 3.10: Position estimate along X axis using CKF

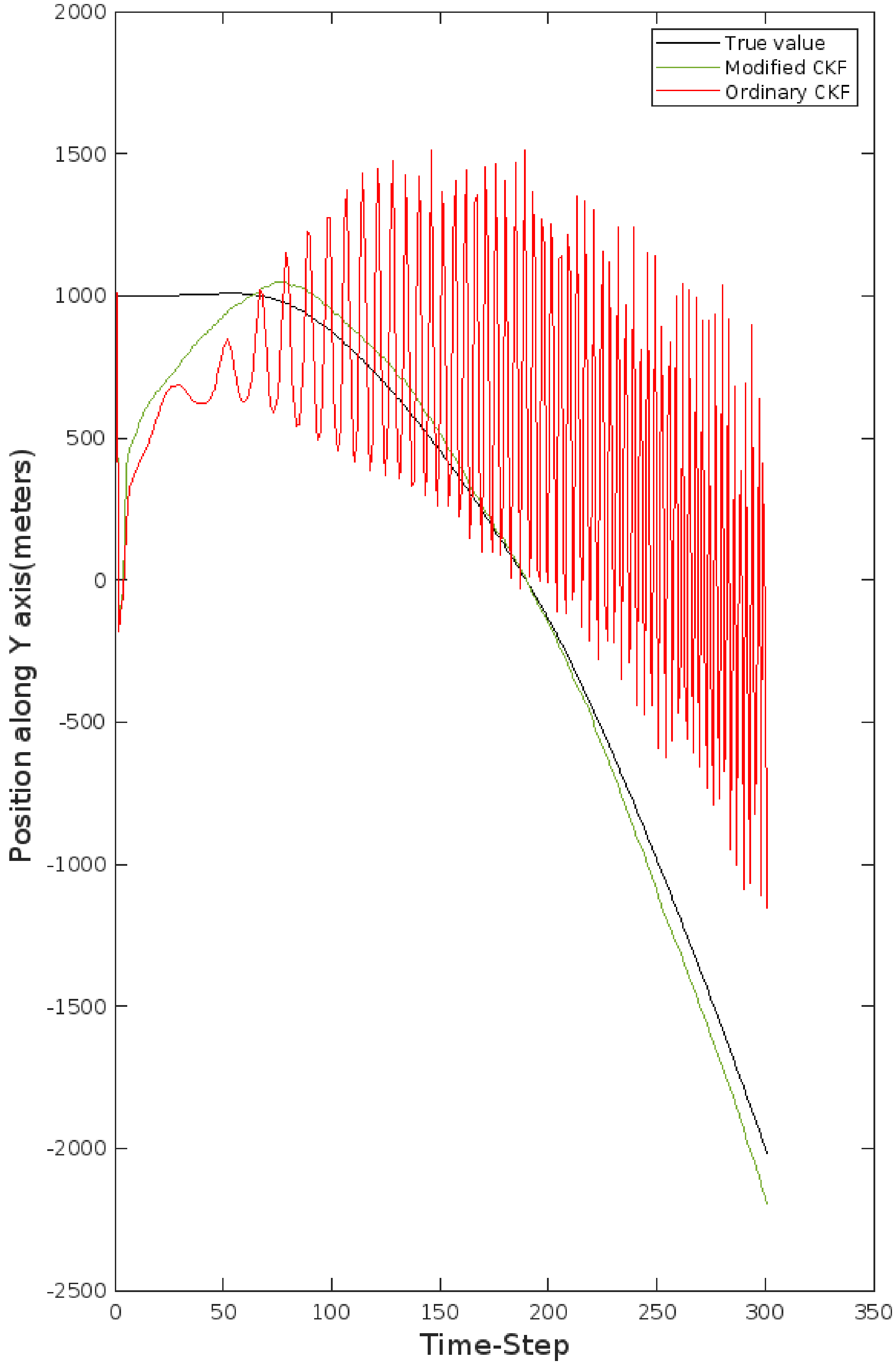

Figure 3.11: Position estimate along Y axis using CKF

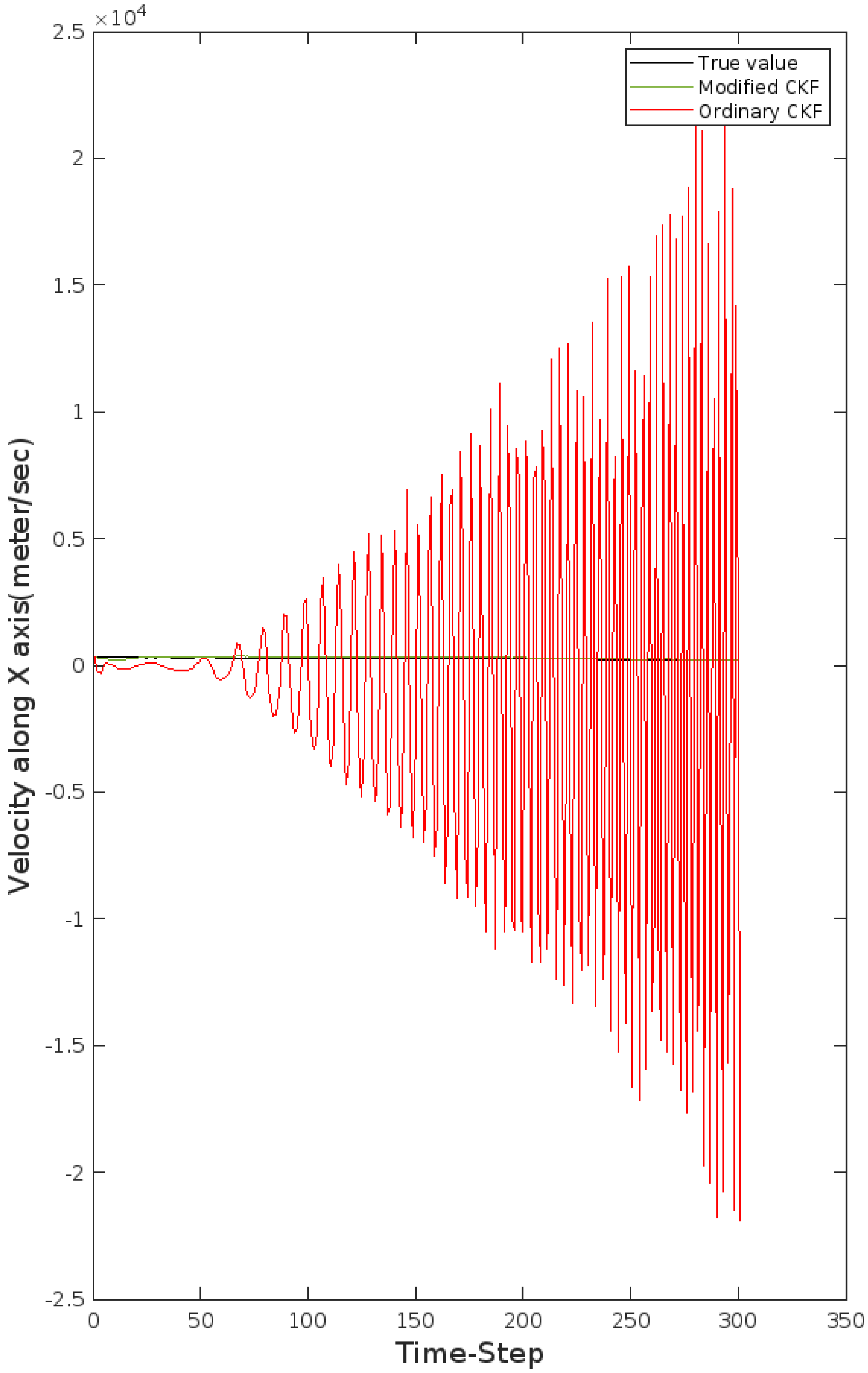

Figure 3.12: Velocity estimate along X axis using CKF

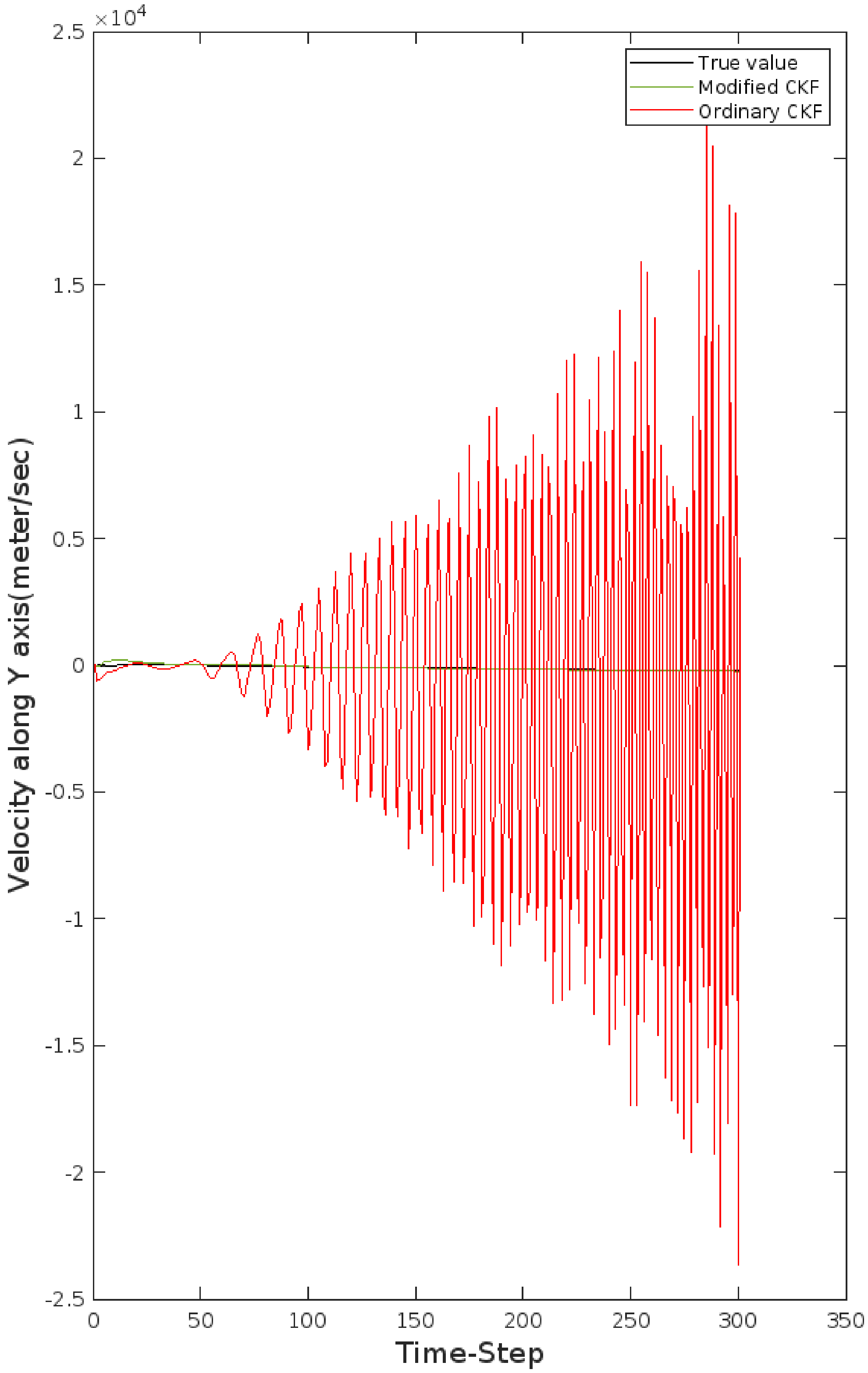

Figure 3.13: Velocity estimate along Y axis using CKF

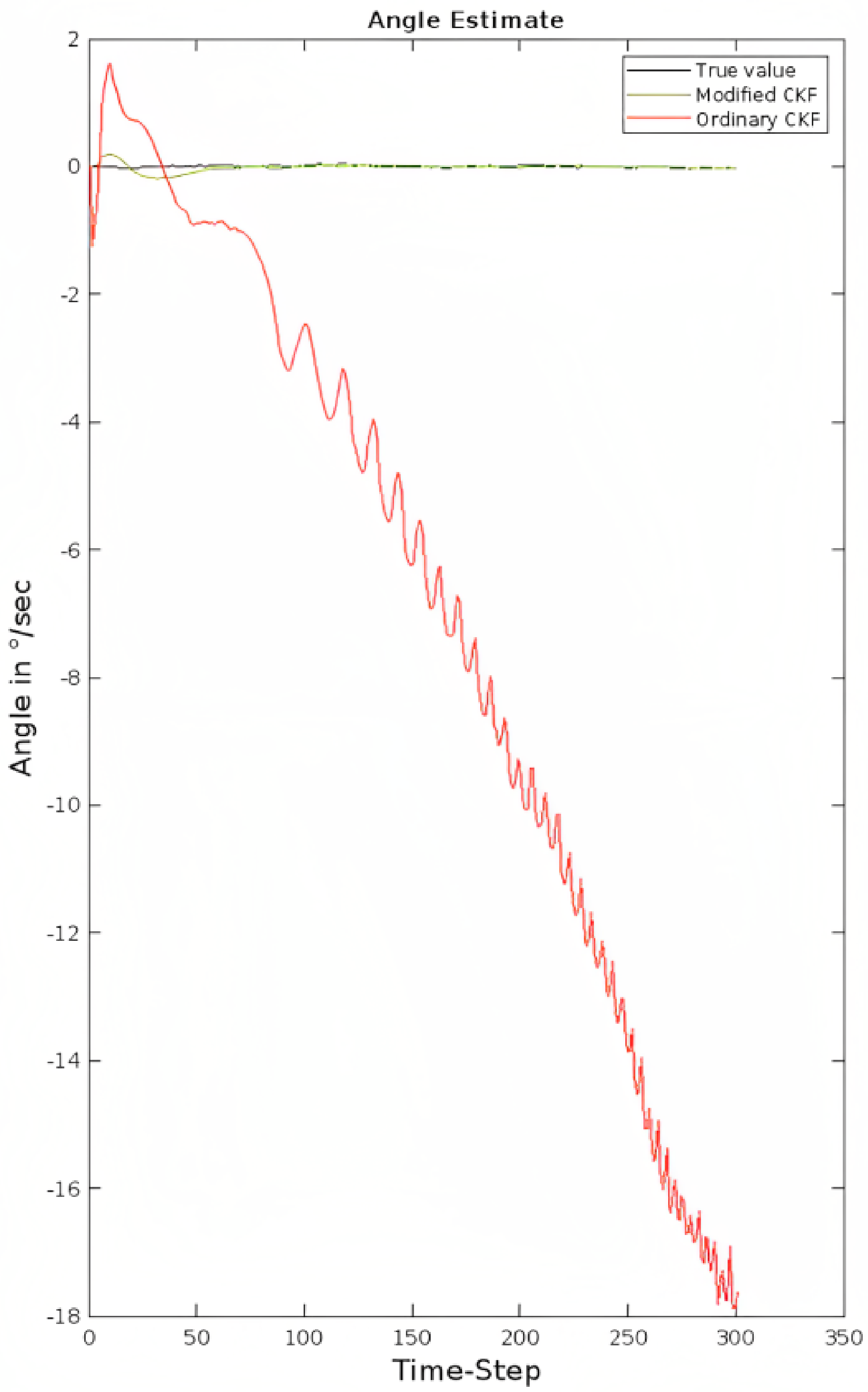

Figure 3.14: Angle estimate using CKF

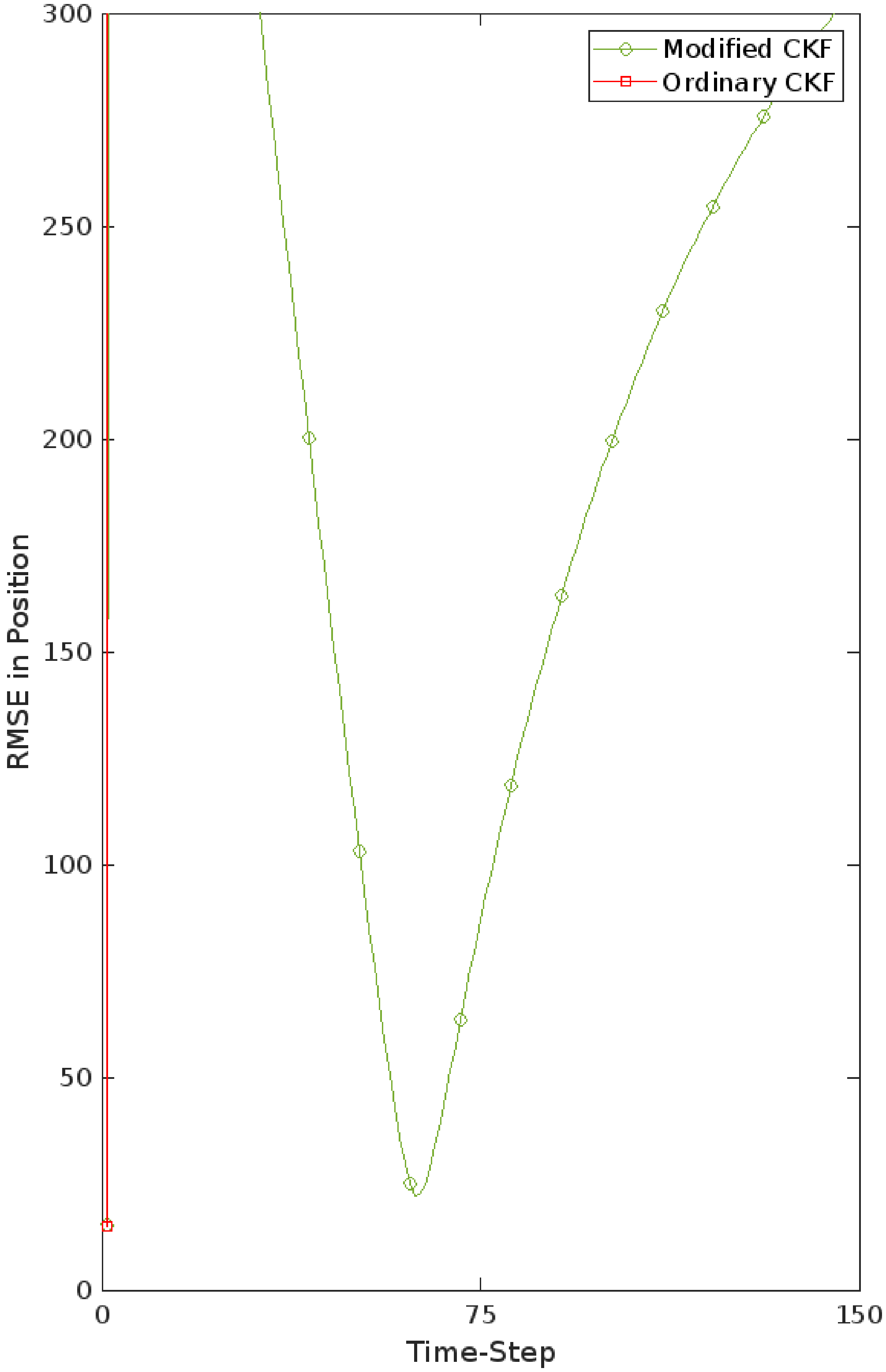

Figure 3.15: RMSE in position estimation using CKF

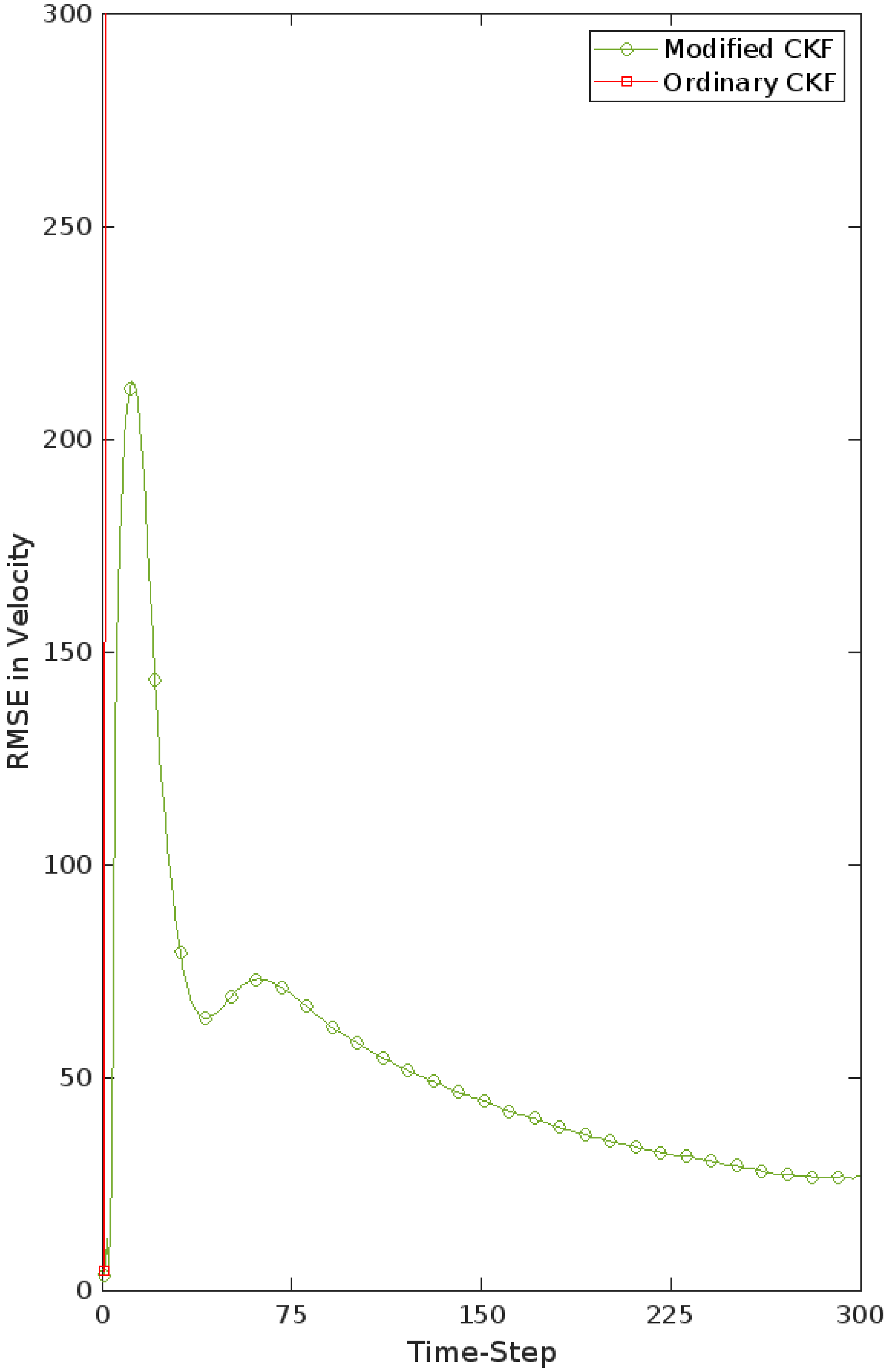

Figure 3.16: RMSE in velocity estimation using CKF

Chapter 4

Roadmap of Mathematical Model and Finance Dataset

This section includes dataset collection, model creation, and methodology using the KF process equation for stock market price prediction.

4.1 Dataset

For the purpose of getting historical datasets on the stock market, yahoo finance happens to be a good source due to its real-time nature [67]. Those datasets include data from trusted finance authorities. The financial dataset of a few companies listed on the National Stock Exchange (NSE) India for the year 2012 from yahoo finance.

The database is chosen so that it has a low standard deviation and should take usual behavior into consideration. It was crucial to choose a time period over which there are no abrupt spikes in the stock market's value (either positive or negative). A historical database is offered by yahoo finance and can be utilized for training.

The historical database of stock consists of the following important parameters :

Date: The date per day when the following values were collected.

OpenPrice: The opening value of the stock market

ClosePrice: The stock's closing price that day

HighPrice: The stock's highest price that day

LowPrice: The stock's lowest value that day

AdjClose: A stock's closing price before the market opens the following day, modified to compensate for corporate actions and distributions.

Volume: Total amount of shares traded on that day.

4.2 Technical Indices

Technical indices help us to decide when to buy any stock using historical data for forecasting the direction of prices.

1. Stock Momentum [46]: It measures the rate at which the price of a specific stock fluctuates i.e. we can analyze the speed of the fall or rise in the particular price. It is the average of the last N days' momentum of the stock where momentum is calculated as 1 if there is a price increase and -1 if there is a price decrease from the previous day.

2. Moving Average Convergence Divergence [1]: The *MACD* is a momentum trend-following indicator that depicts the association between two price moving averages. Increased market momentum, whether up or down, is indicated by a wider gap between the 12- and 26-period exponential moving averages.

$$MACD = EMA_{12} - EMA_{26} \tag{4.1}$$

   where, EMA_{12} and EMA_{26} respectively are the 12_{th} and 26_{th} day estimated moving averages.

3. Relative Strength Index [33]: The average price gains and losses over a specified time period are calculated by *RSI*. These two indicators are frequently combined to give analysts a more thorough technical view of the market, these two indicators are often combined.

$$RSI = 100 - \frac{100}{\frac{1+EMA_{up}^{N}}{EMA_{down}^{N}}} \tag{4.2}$$

4.3 Data Preprocessing

Data preprocessing is a crucial step in stock prediction as it helps ensure the quality and suitability of data for analysis. These are some common data preprocessing techniques used in stock prediction. The choice of techniques depends on the specific characteristics of the dataset and the requirements of the prediction model being used.

Here are some common techniques used in data preprocessing for stock prediction:

1. Data Cleaning: This involves handling missing values, outliers, and inconsistent data. Missing values can be imputed using methods such as mean imputation, forward filling, or backward filling. Outliers can be detected and treated using statistical techniques like z-score or percentile-based methods. Inconsistent data, such as conflicting stock prices, can be resolved by taking the average or applying data correction techniques.

2. Feature Selection: It is important to identify the relevant features that are most influential in stock prediction. Techniques like correlation analysis, information gain, or feature importance from machine learning models can help identify the most important features. Removing irrelevant or redundant features can improve model performance and reduce complexity.

3. Feature Scaling: Scaling the features to a similar range can help avoid bias towards variables with larger values. Common scaling techniques include standardization (0 mean and 1 as standard deviation) or normalization (scaling to a specific range, e.g., 0 to 1).

4. Time Series Resampling: Stock data is often sampled at irregular time intervals. resampling techniques, such as upsampling or downsampling, can be used to convert the data into a consistent frequency (e.g., daily, weekly, monthly) for easier analysis and modeling [81]. Common methods include interpolation, averaging, or taking the last value within the desired time frame.

5. Handling Imbalanced Data: In stock prediction, there may be a class imbalance between upward and downward movements in stock prices. Techniques like oversampling (e.g., SMOTE) or undersampling can be used to balance the classes and improve model performance.

6. Handling Sequential Data: Stock data is often sequential in nature, and the order of the data can be important. Techniques such as time lagging or windowing can be applied to create lagged features or sliding windows to capture temporal patterns in the data.

7. Splitting the Dataset: It is crucial to split the dataset into validation, training, and testing sets. The training dataset is used to train the model, the validation dataset is used for hyperparameter tuning, and the testing dataset is used for evaluating the final model's performance. The split can be done randomly or based on a specific time period to account for temporal dependencies in stock data.

4.4 Proposed Model

Here is a thorough explanation of how the Kalman filter works in a dynamic financial state that is stock market price prediction.

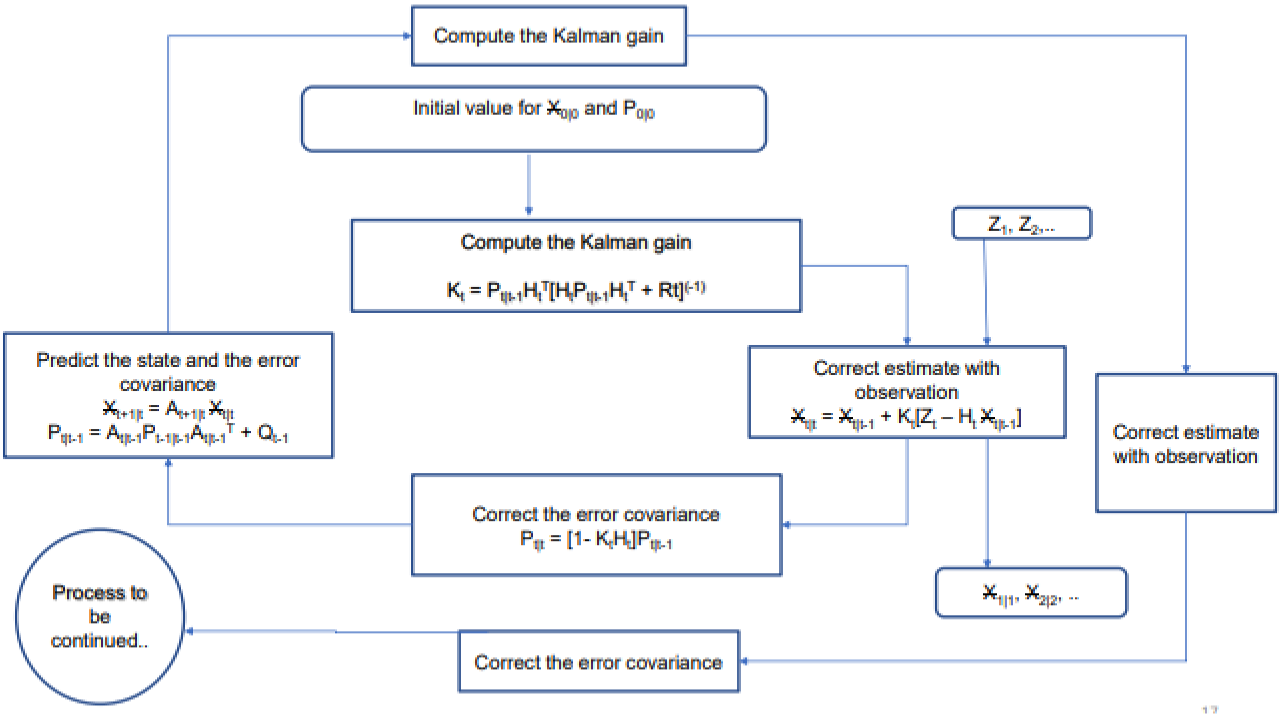

Figure 4.1: A complete picture of the working steps of the Kalman filter for stock market price prediction

We have considered that the highest price of the security (stock) i.e. m_{tk} on the next day is determined by certain factors like opening price, previous day's closing price, etc., and is linearly dependent on these factors.

Thus **the highest price of the stock on the next day** can be expressed as :

$$m_k = \sum_{i=0}^{N} a_{k-i} x_k(i) \tag{4.3}$$

where, a_k: market indices at time t_k, x_k: unknown parameter, N: number of previous days under consideration

The **dynamics of** x_k can be mathematically represented as :

$$x_k = I x_{k-1} + \eta_k \tag{4.4}$$

where, I: Identity matrix, η : mean zero Gaussian process with covariance Q_k

and the **measurement equation** is :

$$m_k = \sum_{i=0}^{N} a_{k-i} x_k(i) + v_k \tag{4.5}$$

where, v_k : measurement error and approximate as zero-mean Gaussian process with covariance R.

equations (4.4) and (4.5) can be represented by generalized state-space model equations that are :

$$x_k = F_k x_{k-1} + \omega_k \tag{4.6}$$

$$z_k = H_k x_k + \zeta_k \tag{4.7}$$

The measurement update for the $(k+1)^{th}$ day is done after the prediction of the highest price of the stock for the $(k+1)^{th}$ day i.e. $m_{k+1|k}$ based on x_k and availability of the true value of the highest price of the stock, $m_{k+1|k+1}$.

The Highest stock market price (HSMP) can be modeled as

$$\lambda_k^p = \alpha_k(0)\theta_k^{op} + \sum_{i=1}^{\delta_n} \alpha_k(d)\lambda_{k-d}^t + \alpha_k(\delta_n+1)\theta_k^{(MACD,N_1)} + \alpha_k(\delta_n+2)\theta_k^{(RSI,N_1)} + \alpha_k(\delta_n+3)\theta_k^{(SM,N_1)} \tag{4.8}$$

Where, λ_k^p, λ_k^t, and x_k is a vector representing the corresponding unknown linear coefficient, estimated on the k^{th} day with KF. And,

$$x_{k+1}(i) \in X_{k+1} = \begin{bmatrix} x_{k+1}(0) \\ x_{k+1}(1) \\ . \\ . \\ . \\ x_{k+1}(N+3) \end{bmatrix} \tag{4.9}$$

and X_{k+1} is a vector corresponding to the unknown linear coefficients that were estimated on the $(K+1)^{th}$ day, and N is the total number of days that have been taken into account.

4.5 Simulation Results

For the simulation of the proposed mathematical model, the algorithm is applied to the data of certain companies listed on the national stock exchange of the Indian stock market.

The parameters for a period of 70 samples are trained. i.e. 70 days with a period of 7 days for the previous measurement. N=7. After the prediction of the highest price of the stock on the next day's price for the next 30 days, those parameters are updated.

We assign the initial value of the parameters as:

a_0 = [0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1]

for the prediction and measurement equation following are the assumptions made :

- The state is the previous day's indices and measurement is the highest price of the current day.
- Given the opening price of today, the highest price of today is estimated.
- number of state variables considered in the state matrix is 8. While The number of measurement variables in the measurement matrix is 1.
- here, the highest value of the stock the next day is predicted as a linear function of F(past highest values, today's opening value, technical indices [19]), and the estimated moving average (EMA) of the previous day is being day is used as a parameter.

Below are the simulation results for the stock price prediction of a few companies:

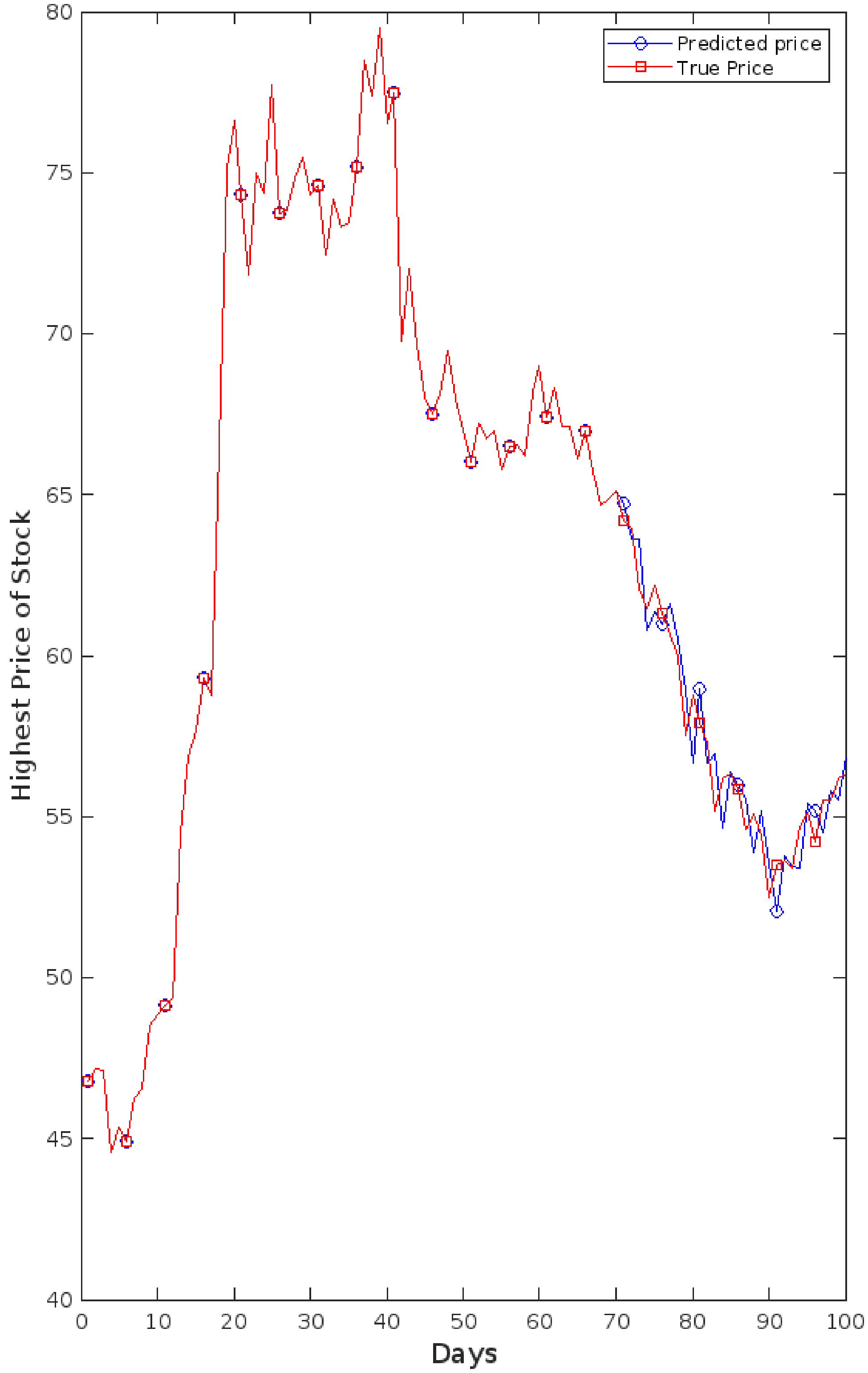

Figure 4.2: True and predicted stock price of BHEL

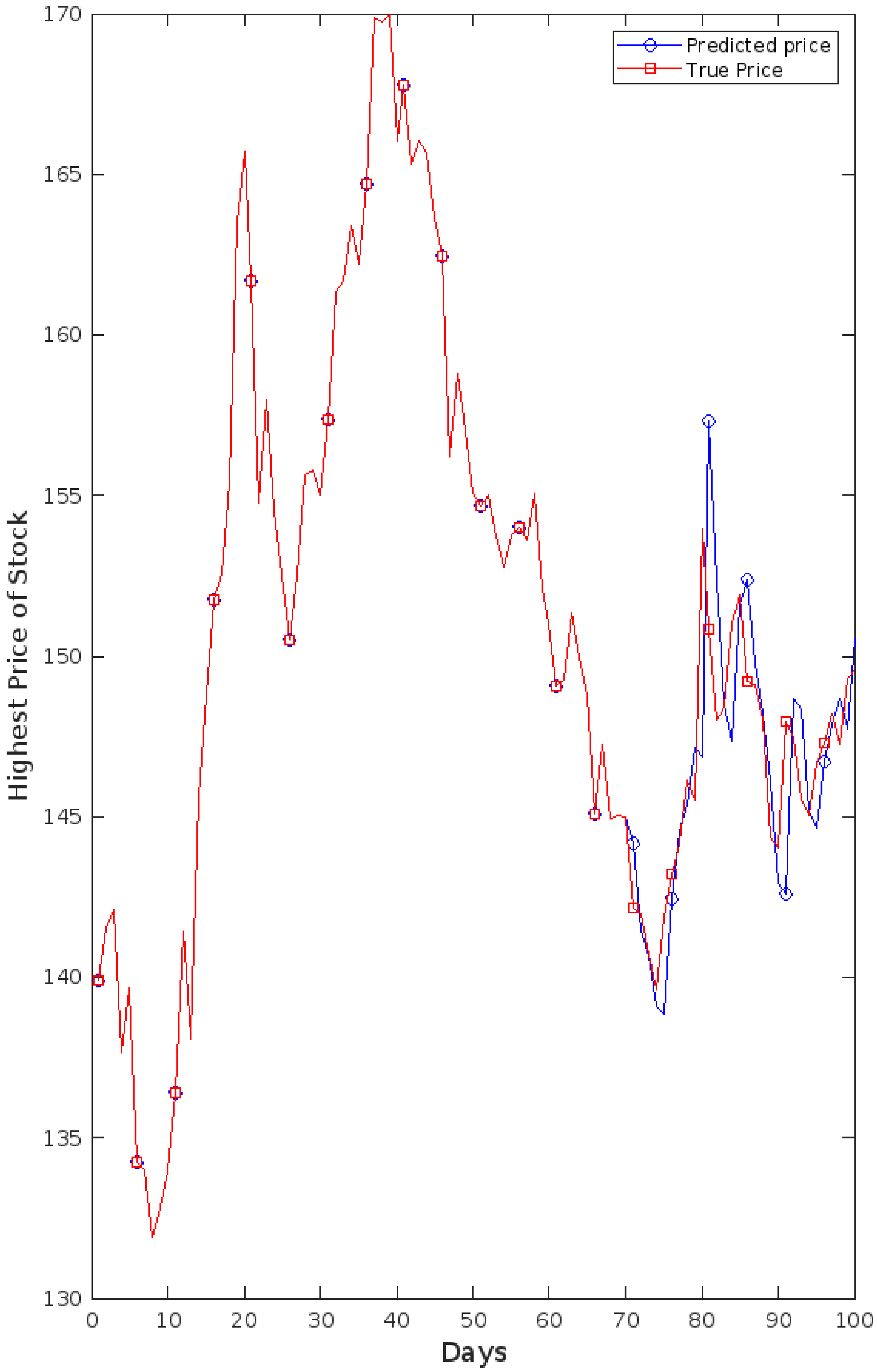

Figure 4.3: True and predicted stock price of GAIL

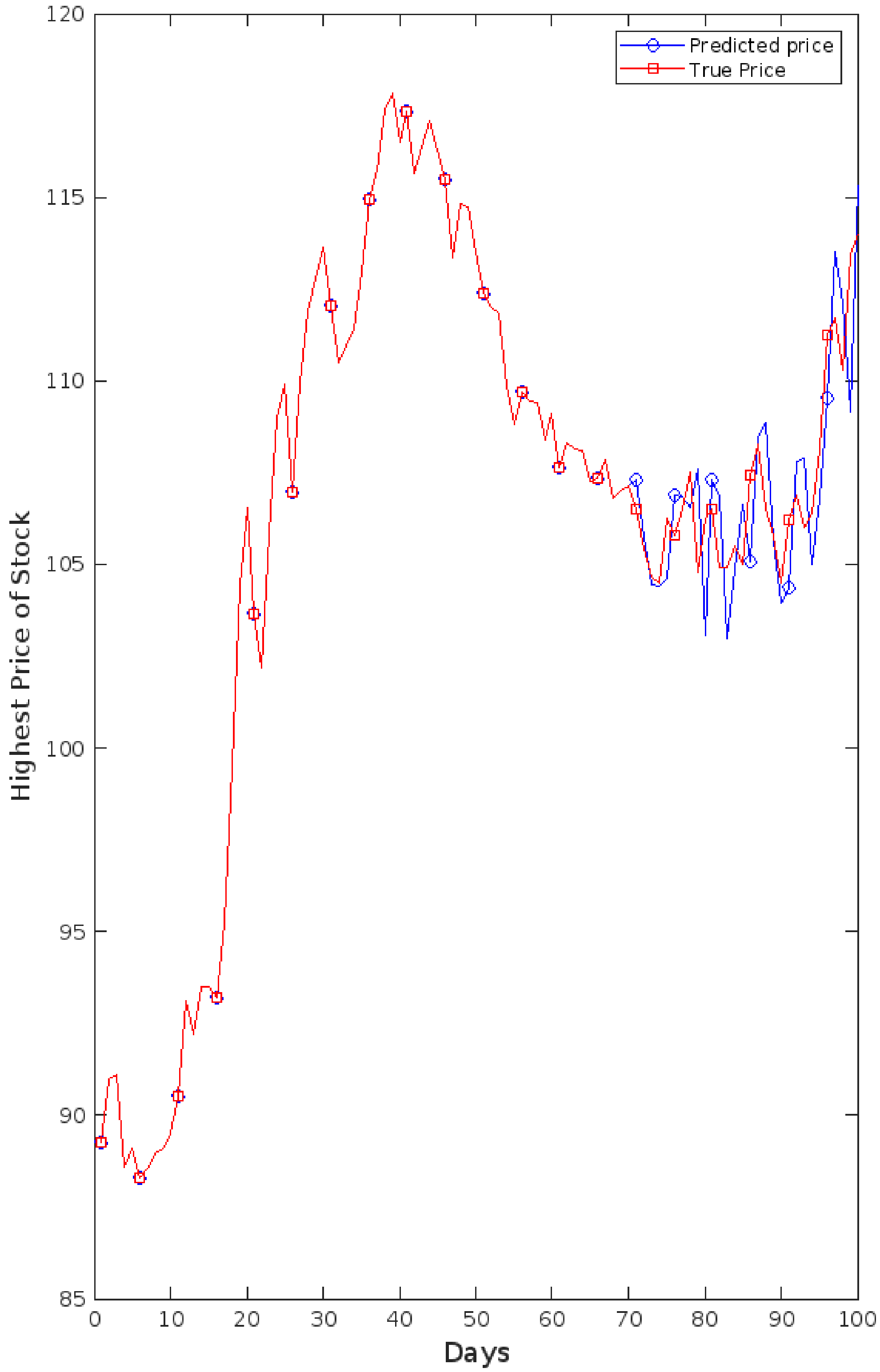

Figure 4.4: True and predicted stock price of IOCL

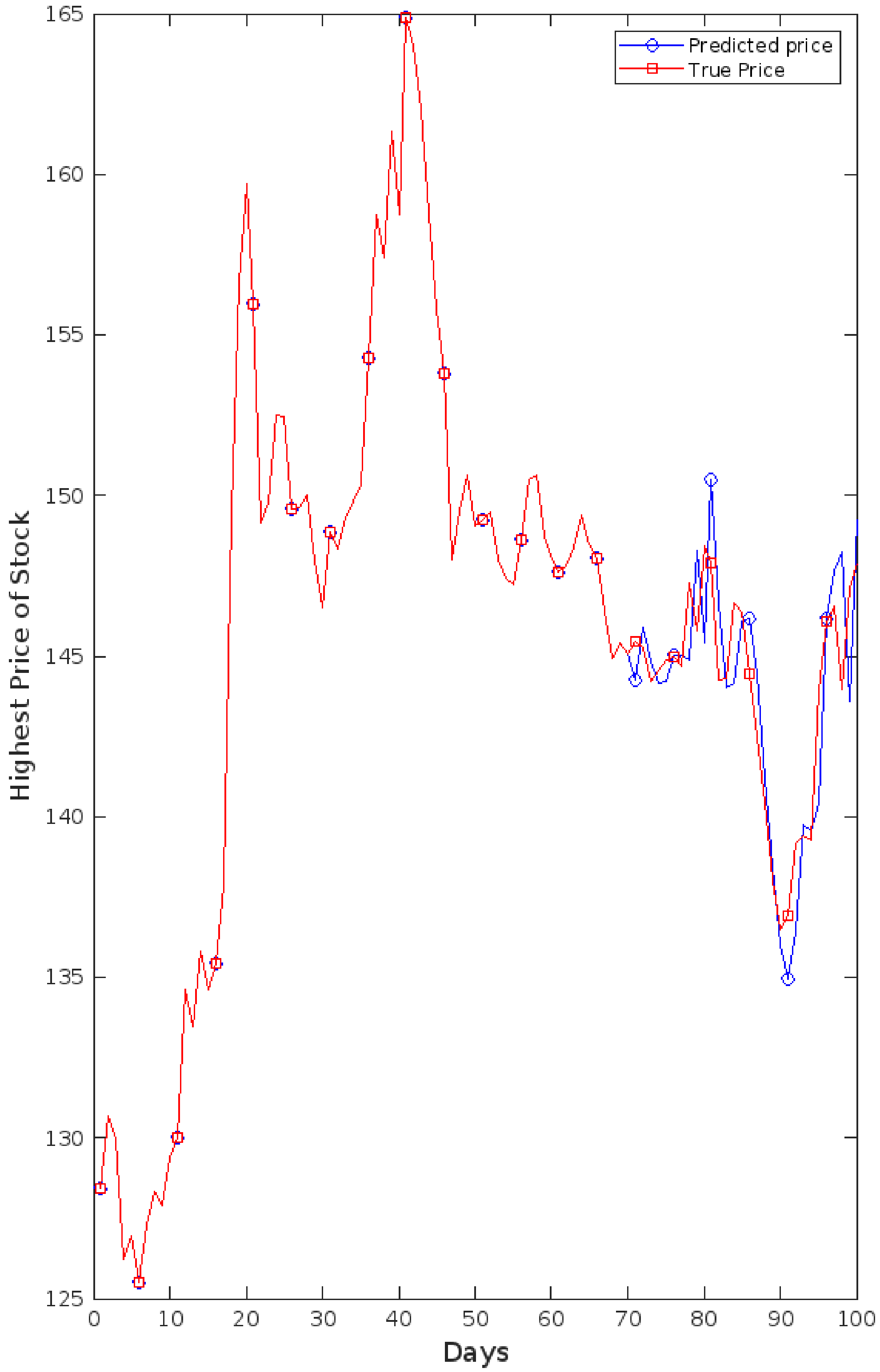

Figure 4.5: True and predicted stock price of COAL INDIA

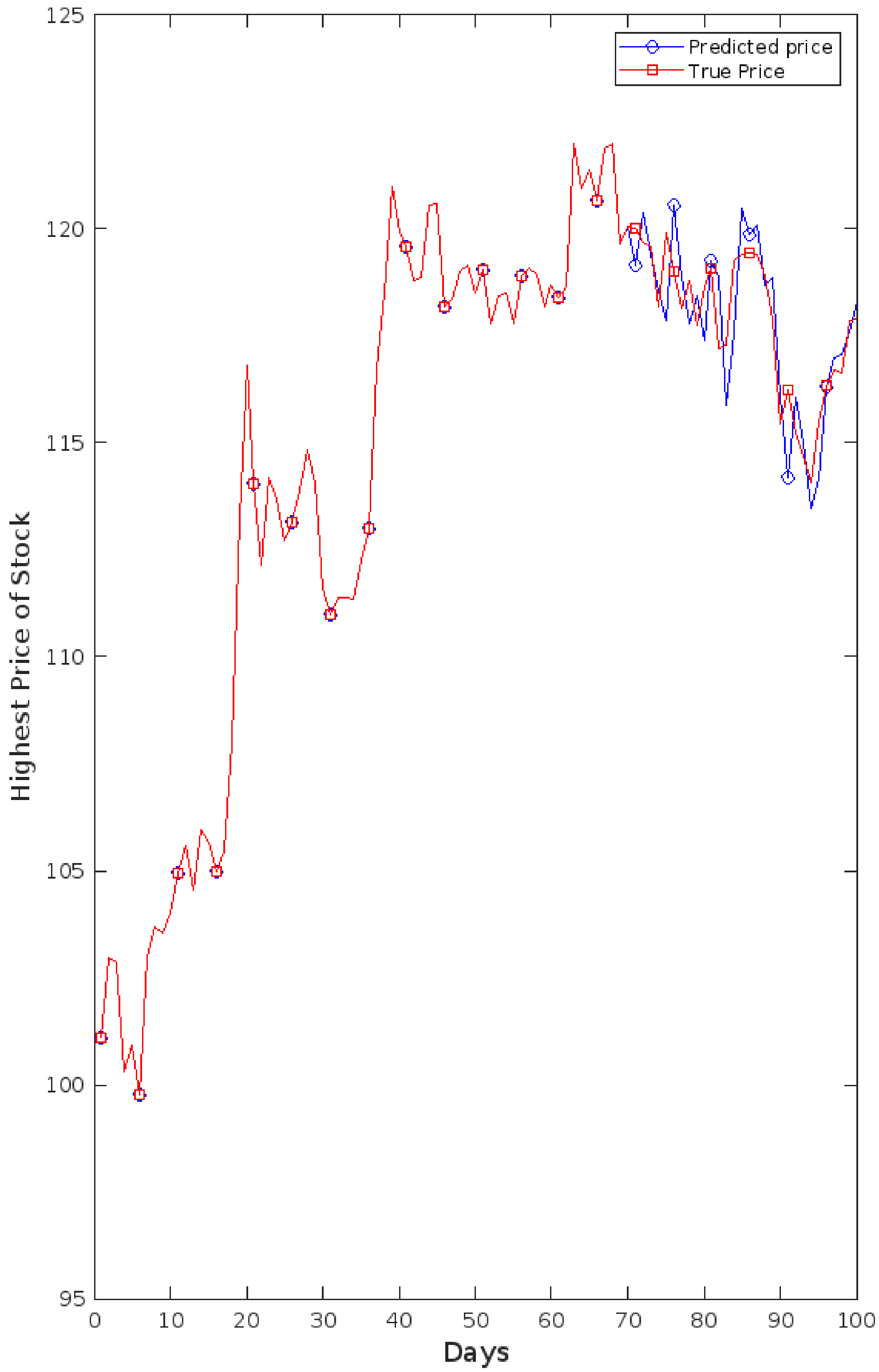

Figure 4.6: True and predicted stock price of NTPC

The simulation results of stock prices of 5 companies are shown in the above figures. It can be observed from the figures that The prediction on the test data, data after the 70^{th} day, is closely following the true price of the stocks with only a little deviation.

Here are the Bar graph plots for percentage accurate relative jump (PARJ) under 30 % error and percentage accurate share prediction (PASP) under 1 % error bound, for different companies:

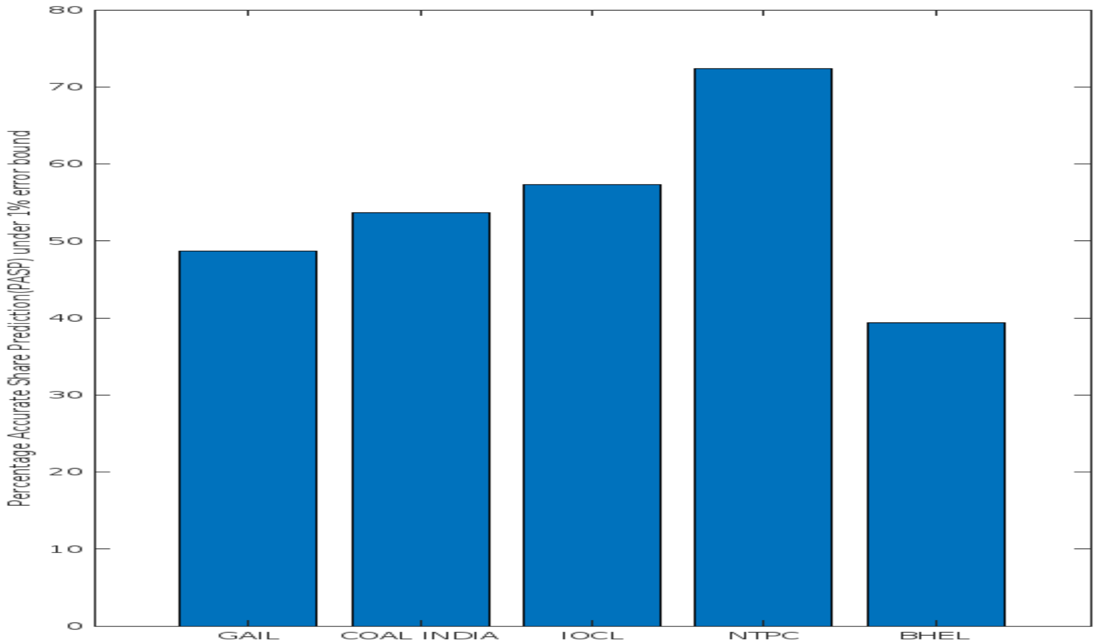

Figure 4.7: Percentage Accurate Share Prediction under 1 % error bound for different companies

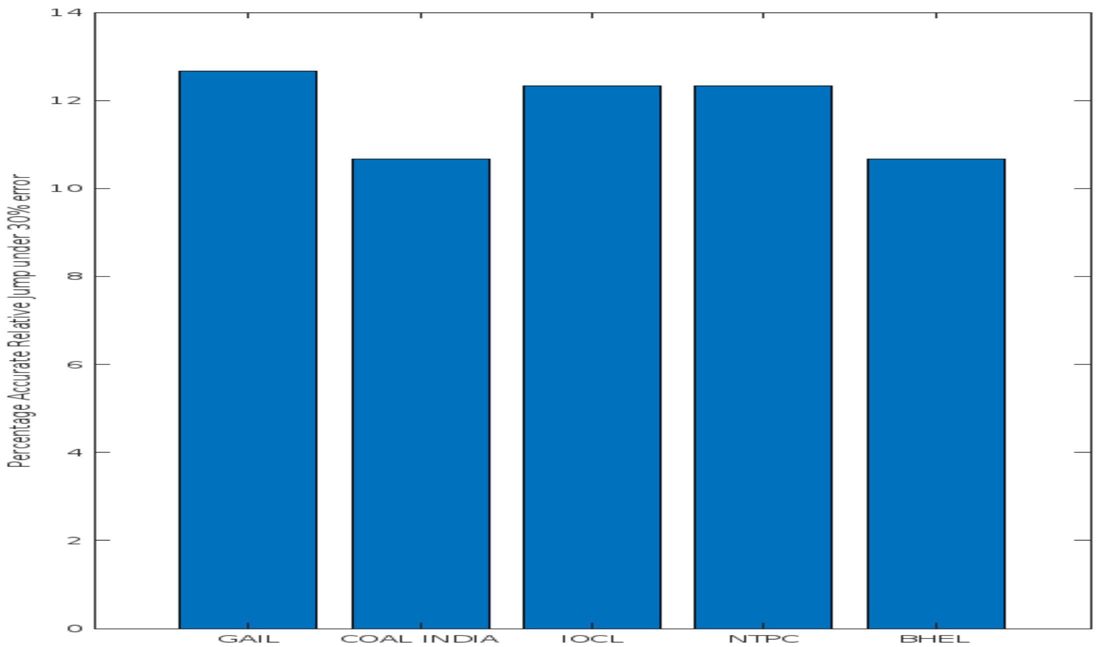

Figure 4.8: Percentage Accurate Relative Jump under 30 % for different companies

4.5.1 Comparison plots on the basis of different types of errors

Comparison analysis in the stock price prediction is done for a few companies on the basis of different types of errors which are listed below :

1. **RMSE**: It stands for root mean squared error which provides a measure of how much the predicted values deviate from the actual values on an average.

   The formula for RMSE is as follows:

   $$\mathrm{RMSE} = \sqrt{\frac{1}{n} * \sum_{i=1}^{n} ((\mathrm{predicted_i} - \mathrm{actual_i})^2)} \tag{4.10}$$

   Where:

   n is the number of data points or samples, $\mathrm{predicted_i}$ is the predicted value for the $\mathrm{i^{th}}$ sample, $\mathrm{actual_i}$ is the actual or true value for the $\mathrm{i^{th}}$ sample.

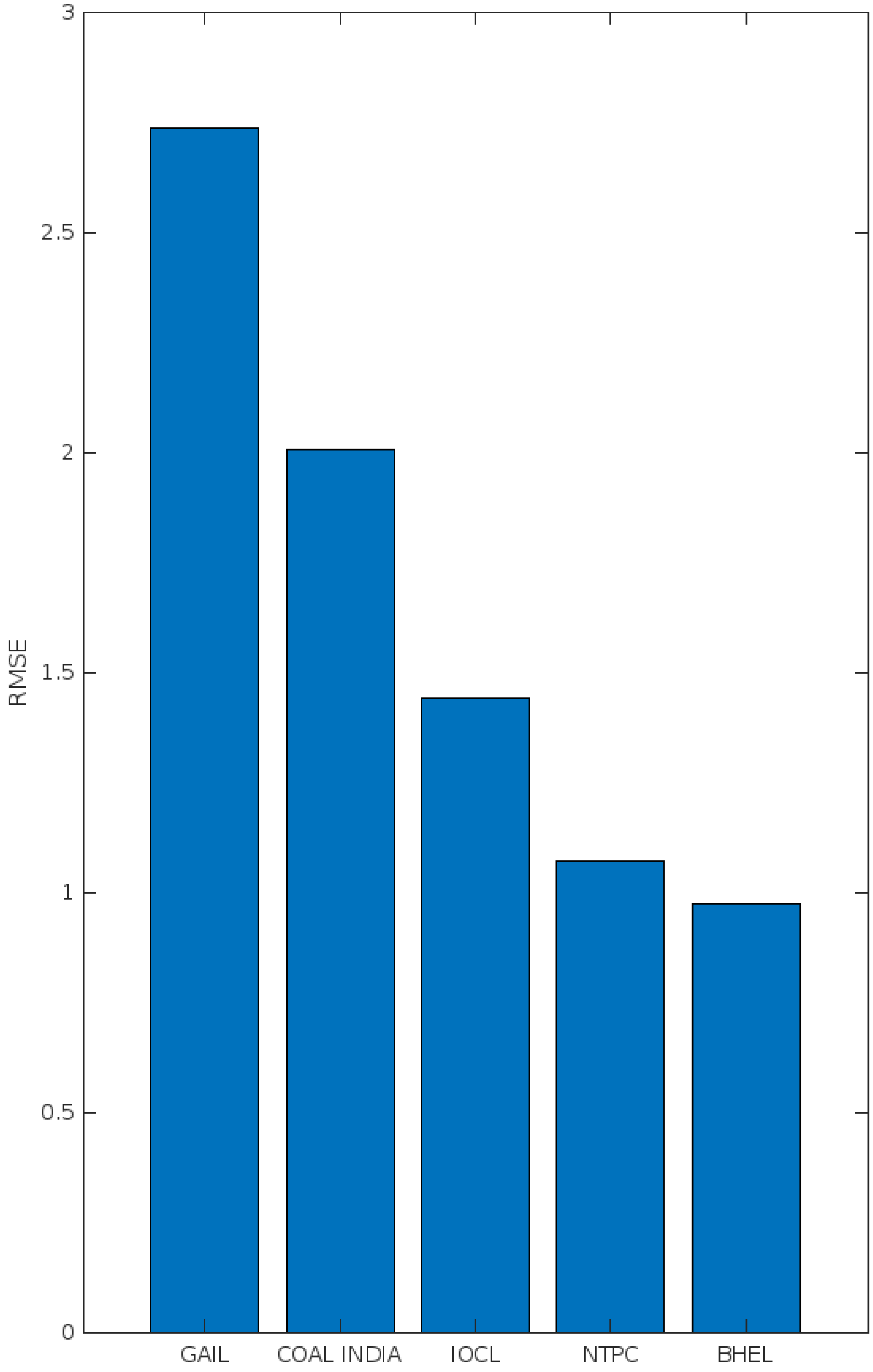

Figure 4.9: RMSE plots for different companies

2. **MSE**: It stands for mean squared error which provides a measure of the average squared deviation between the predicted values and the actual values.

   The formula for MSE is as follows:

$$\mathrm{MSE} = \frac{1}{n} * \sum_{i=1}^{n} ((\mathrm{predicted_i} - \mathrm{actual_i})^2) \tag{4.11}$$

   Where:

   n is the number of data points or samples, $\mathrm{predicted_i}$ is the predicted value for the $\mathrm{i^{th}}$ sample, $\mathrm{actual_i}$ is the actual or true value for the $\mathrm{i^{th}}$ sample.

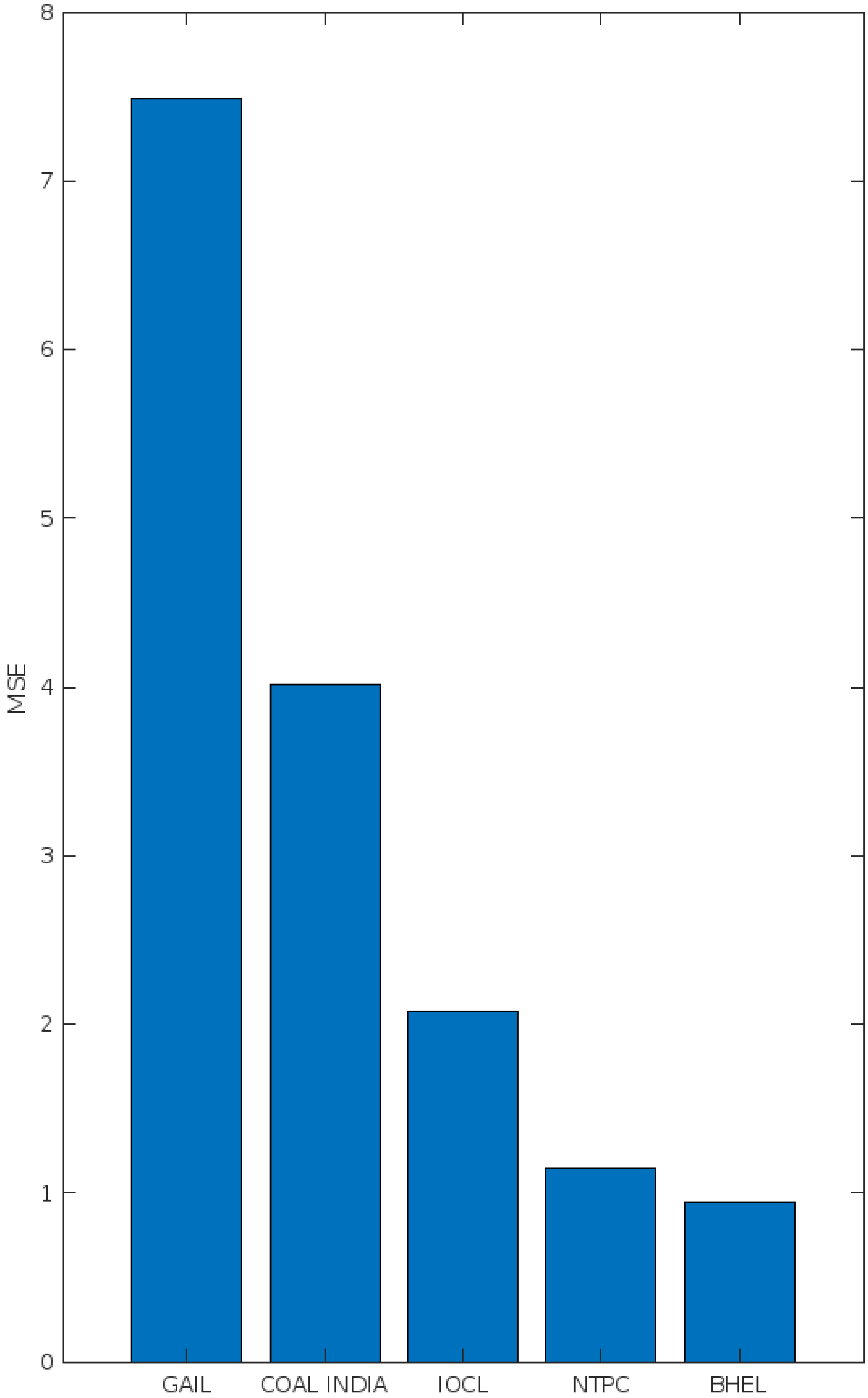

Figure 4.10: MSE plots for different companies

3. **MAE**: It stands for mean absolute error which provides a measure of the average absolute deviation between the predicted values and the actual values.

   The formula for MAE is as follows:

$$\text{MAE} = \frac{1}{n} * \sum_{i=1}^{n} (\text{predicted}_\text{i} - \text{actual}_\text{i}) \tag{4.12}$$

   Where:

   n is the number of data points or samples, $\text{predicted}_\text{i}$ is the predicted value for the i^th sample, actual_i is the actual or true value for the i^th sample.

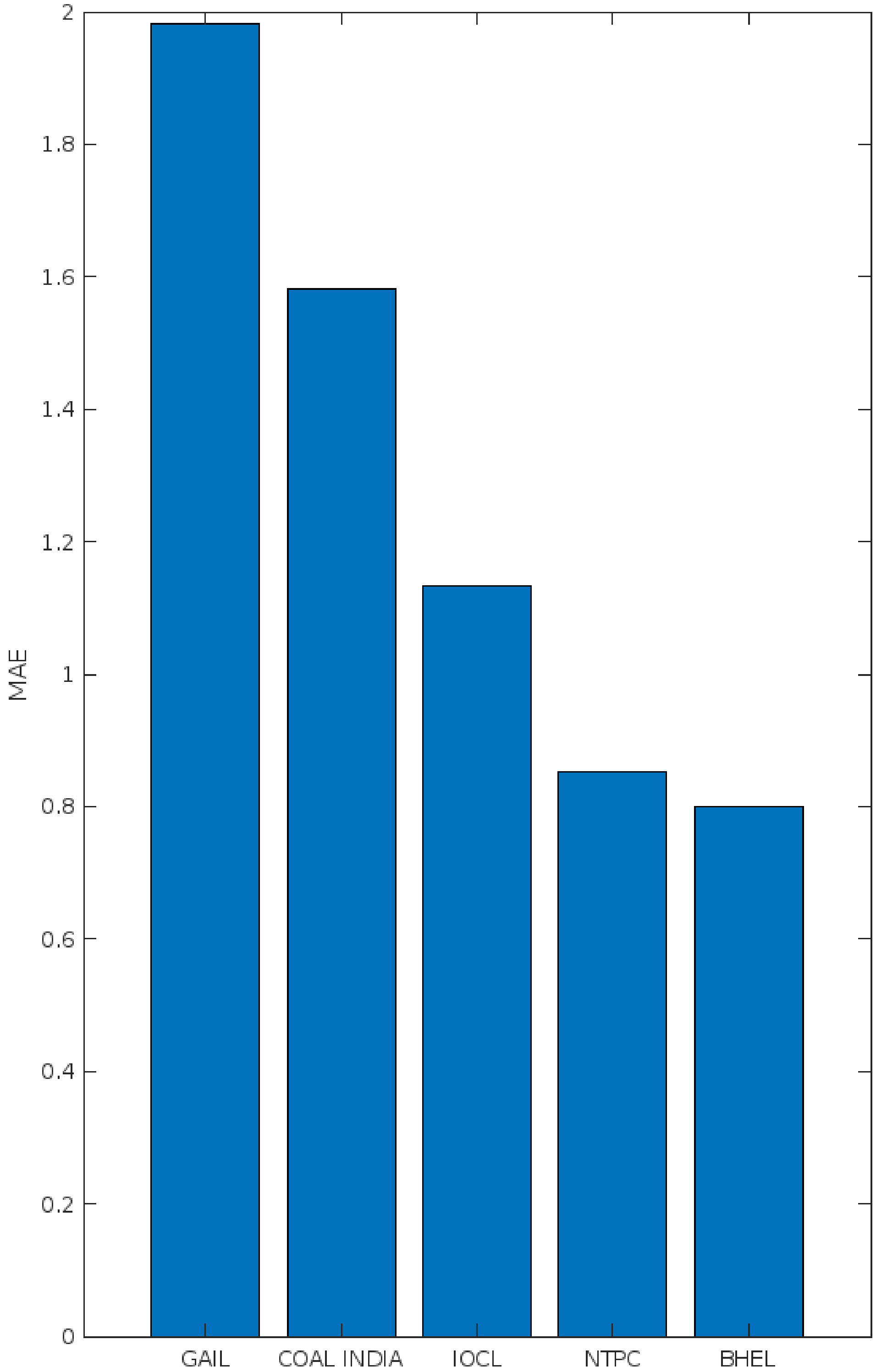

Figure 4.11: MAE plots for different companies

4. **MARD**: It stands for mean absolute relative difference which provides a measure of the average relative deviation between the predicted values and the actual values.

   The formula for MARD is as follows:

$$\text{MARD} = \frac{1}{n} * \sum_{i=1}^{n} \left(\left| \frac{(\text{predicted}_\text{i} - \text{actual}_\text{i})}{\text{actual}_\text{i}} \right| \right) \tag{4.13}$$

   Where:

   n is the number of data points or samples, $\text{predicted}_\text{i}$ is the predicted value for the i^th sample, actual_i is the actual or true value for the i^th sample.

   $|x|$ represents the absolute value of x.

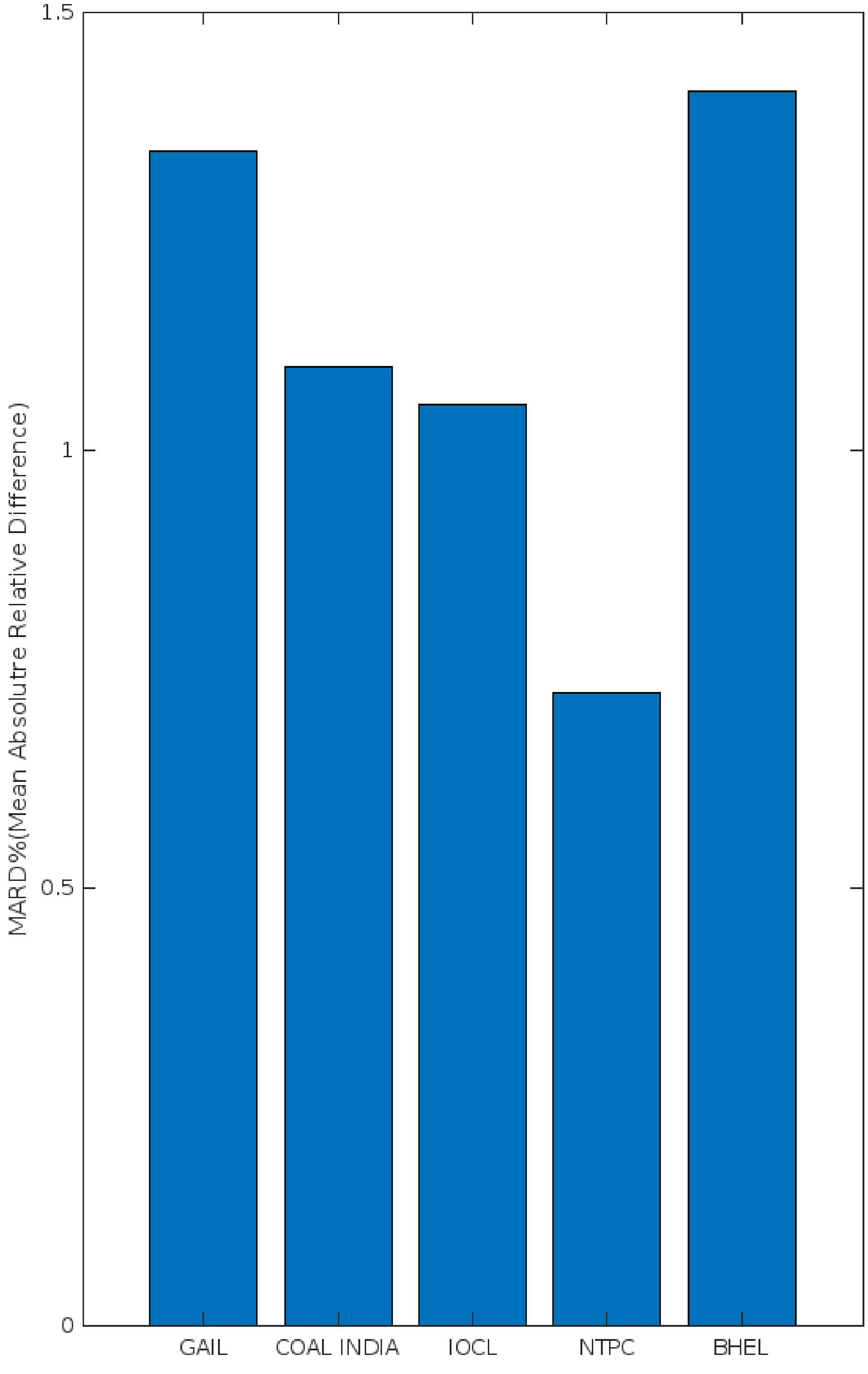

Figure 4.12: MARD plots for different companies

Chapter 5

Conclusions and Future Works

The thesis examines the performance of the following four system models: LSTM, Linear Regression, DMD, and the KF. The derived closed-form expressions for performance metrics are validated by Monte-Carlo-based simulations.

5.1 Conclusions

Through this Project, one can observe a few Machine Learning Techniques, The KF, and DMD are used to predict and compare the prices of the stock market. with reasonable accuracy the result shows how historical data are used to predict stock movement based on the *RMSE* obtained using different algorithms. So, on a concluding note, the following points can be said:

- Kalman Filter can be a powerful tool for financial modeling when used properly. Since the *RMSE* obtained is the lowest among all other algorithms used.
- The Kalman filter is computationally efficient, making it a suitable tool for real-time applications. Because the iterative process of measurement and estimation continues until we find the optimal state.
- LSTM is a good choice when the requirement is better accuracy and low variance but it is comparatively slower.
- Collecting and aligning the assumptions that are to be used in the financial market is also not easy, as it requires a lot of time and energy.
- It has been observed that the DMD algorithm can capture the original trend only when the exogenous variables such as government policies are not considered.

- Linear regression works on the principle that the variables assumed are linearly related to the stock market variables which sometimes leads to failure in the financial market because of its dynamic nature.

Below is the comparison table for stock price prediction based on *RMSE* for NTPC. we can see that the error obtained in the case of the Kalman filter is minimal, which proves to be an efficient method in finance, especially in stock market price prediction.

Method	RMSE(NTPC)
Kalman Filter	1.01315
LSTM	3.6251
Linear Regression	5.0291
DMD	17.3652

Figure 5.1: Comparison table of stock price prediction for NTPC based on *RMSE*

5.2 Future Works

The implemented method is based on the KF to predict the stock market price. And the simulation result has shown that the Kalman filter is best among the other three methods used for prediction.

- As of now, we have incorporated the predictive analysis assuming the coefficients to be linearly varying. So, the future scope might be improved by extending the process equation of the parameters to non-linear modeling.
- Future Work might be done by considering other external factors like government policies, political decisions, etc which are some important factors to affect the market data.
- We will be incorporating and evaluating the impact of technical indices such as the Popularity Index (AR), Willingness Index (BR), Sentiment Analysis [20], etc.
- One can improve the prediction by Identifying a further suitable process equation of the parameters for increased accuracy.

Bibliography

[1] A.Alberto A.Agudelo, N.D. D.Méndez, and R.A. R.Medina. “Artificial intelligence applied to investment in variable income through the MACD (moving average convergence/divergence) indicator”. In: *Journal of Economics, Finance and Administrative Science* 26.52 (2021), pp. 268–281.

[2] D.J. Abadi. “Data management in the cloud: Limitations and opportunities.” In: *IEEE Data Eng. Bull.* 32.1 (2009), pp. 3–12.

[3] A.N. Akansu and M.U. Torun. *A primer for financial engineering: financial signal processing and electronic trading*. Academic Press, 2015.

[4] I. Arasaratnam and S. Haykin. “Cubature Kalman Filters”. In: *IEEE Transactions on Automatic Control* 54.6 (2009), pp. 1254–1269.

[5] D. Arndt and W. Gersten. “Data management in analytical customer relationship management”. In: *Proceedings of the Workshop „Data Mining for Marketing Applications “at the 12th European Conference on Machine Learning and the 5th European Conference on Principles and Practices of Knowledge Discovery in Database, Freiburg*. Citeseer. 2001, pp. 25–38.

[6] T. Aven. “Risk assessment and risk management: Review of recent advances on their foundation”. In: *European Journal of Operational Research* 253.1 (2016), pp. 1–13.

[7] J.S. Bay. *Fundamentals of linear state space systems*. WCB/McGraw-Hill New York, 1999.

[8] N. Bergman. “Recursive bayesian estimation”. In: *Department of Electrical Engineering, Linköping University, Linköping Studies in Science and Technology. Doctoral dissertation* 579.11 (1999).

[9] S. Bisgaard and M. Kulahci. *Time series analysis and forecasting by example*. J Wiley & Sons, 2011.

[10] G. Bishop, G. Welch, et al. “An introduction to the kalman filter”. In: *Proc of SIGGRAPH, Course* 8.27599-23175 (2001), p. 41.

[11] P.P. Boyle. “Options: A monte carlo approach”. In: *Journal of financial economics* 4.3 (1977), pp. 323–338.

[12] D. Broby. “The use of predictive analytics in finance”. In: *The Journal of Finance and Data Science* (2022).

[13] R.L. Buchanan and R.C. Whiting. “Risk assessment and predictive microbiology”. In: *Journal of food Protection* 59.13 (1996), pp. 31–36.

[14] J.V. Candy. *Bayesian signal processing: classical, modern, and particle filtering methods*. John Wiley & Sons, 2016.

[15] R. Chen, Y. Feng, and D. Palomar. “Forecasting intraday trading volume: a kalman filter approach”. In: *Available at SSRN 3101695* (2016).

[16] J.R. Coakley and C.E. Brown. “Artificial neural networks in accounting and finance: Modeling issues”. In: *Intelligent Systems in Accounting, Finance & Management* 9.2 (2000), pp. 119–144.

[17] A. Cowles. “Stock market forecasting”. In: *Econometrica, Journal of the Econometric Society* (1944), pp. 206–214.

[18] C.Eduardo D.Moura, A. Pizzinga, and J. Zubelli. “A pairs trading strategy based on linear state space models and the Kalman filter”. In: *Quantitative Finance* 16.10 (2016), pp. 1559–1573.

[19] S.M. Daniali et al. “Predicting volatility index according to technical index and economic indicators on the basis of deep learning algorithm”. In: *Sustainability* 13.24 (2021), p. 14011.

[20] M.D. Devika, Cª. Sunitha, and A. Ganesh. “Sentiment analysis: a comparative study on different approaches”. In: *Procedia Computer Science* 87 (2016), pp. 44–49.

[21] A. Doucet, A.M. Johansen, et al. “A tutorial on particle filtering and smoothing: Fifteen years later”. In: *Handbook of nonlinear filtering* 12.656-704 (2009), p. 3.

[22] H. Drygas. *The coordinate-free approach to Gauss-Markov estimation*. Vol. 40. Springer Science & Business Media, 2012.

[23] D.J. Earl and M.W. Deem. “Monte carlo simulations”. In: *Molecular modeling of proteins* (2008), pp. 25–36.

[24] R.J. Elliott, W.C. Hunter, and B.M. Jamieson. “Financial signal processing: a self calibrating model”. In: *International Journal of Theoretical and Applied Finance* 4.04 (2001), pp. 567–584.

[25] F.W. Fairman. *Linear control theory: the state space approach*. John W & Sons, 1998.

[26] J.P. Fouque, A. Papanicolaou, and R. Sircar. "Filtering and portfolio optimization with stochastic unobserved drift in asset returns". In: *Communications in Mathematical Sciences* 13.4 (2015), pp. 935–953.

[27] M. Gastaldi and A. Nardecchia. "The Kalman filter approach for time-varying ß estimation". In: *Systems Analysis Modelling Simulation* 43.8 (2003), pp. 1033–1042.

[28] F.A. Gers, J. Schmidhuber, and F. Cummins. "Learning to forget: Continual prediction with LSTM". In: *Neural computation* 12.10 (2000), pp. 2451–2471.

[29] F. Gorunescu. *Data Mining: Concepts, models and techniques*. Springer Science & Business Media, 2011.

[30] H.M.Shirazi, A.M.Shirazi, and M.A.Dastgheib. *A Step by Step Mathematical Derivation and Tutorial on Kalman Filters*. 2019.

[31] J.D. Hamilton. "State-space models". In: *Handbook of econometrics* 4 (1994), pp. 3039–3080.

[32] P.J. Hargrave. "A tutorial introduction to Kalman filtering". In: *IEE colloquium on Kalman filters: introduction, applications and future developments*. IET. 1989, pp. 1–1.

[33] Y. Hari and L.P. Dewi. "Forecasting system approach for stock trading with relative strength index and moving average indicator". In: *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 10.2-3 (2018), pp. 25–29.

[34] M.R. Hassan and B. Nath. "Stock market forecasting using hidden Markov model: a new approach". In: *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*. IEEE. 2005, pp. 192–196.

[35] E. J.Skinner et al. "A Novel Algorithmic Trading Strategy using Hidden Markov Model for Kalman Filtering Innovations". In: *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE. 2021, pp. 1766–1771.

[36] A. Javaheri, D. Lautier, and A. Galli. "Filtering in finance". In: *Wilmott* 3 (2003), pp. 67–83.

[37] M. Jiang, J. Liu, and L. Zhang. "An extended regularized Kalman filter based on Genetic Algorithm: Application to dynamic asset pricing models". In: *The Quarterly Review of Economics and Finance* 79 (2021), pp. 28–44.

[38] O.N. Joseph, O. Albert, and J. Byaruhanga. "Effect of internal control on fraud detection and prevention in district treasuries of Kakamega County". In: *International Journal of Business and management invention* 4.1 (2015), pp. 47–57.

[39] R. Kandepu, B. Foss, and L. Imsland. “Applying the unscented Kalman filter for nonlinear state estimation”. In: *Journal of process control* 18.7-8 (2008), pp. 753–768.

[40] K. Keasey and R. Watson. “Financial distress prediction models: a review of their usefulness 1”. In: *British journal of Management* 2.2 (1991), pp. 89–102.

[41] J. Kienitz and D. Wetterau. *Financial modeling: Theory, implementation, and practice with MATLAB source*. John Wiley & Sons, 2013.

[42] H. Kobayashi, B.L. Mark, and W. Turin. *Probability, random processes, and statistical analysis: applications to communications, signal processing, queueing theory and mathematical finance*. Cambridge University Press, 2011.

[43] B. Kumar, S.B. Karanki, and M. Manikandan. “Dynamic Mode Decomposition Based Approach for Accurate Measurement of Voltage Flicker Components”. In: *IEEE Transactions on Instrumentation and Measurement* PP (Jan. 2023), pp. 1–1.

[44] J.A. Lee, M. Verleysen, et al. *Nonlinear dimensionality reduction*. Vol. 1. Springer, 2007.

[45] M. Levin and G.M. Krause. “Incident detection: A Bayesian approach”. In: *Transportation Research Record* 682 (1978), pp. 52–58.

[46] J. Lewellen. “Momentum and autocorrelation in stock returns”. In: *The Review of Financial Studies* 15.2 (2002), pp. 533–564.

[47] B. Li. “State estimation with partially observed inputs: A unified Kalman filtering approach”. In: *Automatica* 49.3 (2013), pp. 816–820.

[48] Y. Liang et al. “Dynamic data science applications in optimal profit algorithmic trading”. In: *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE. 2020, pp. 1314–1319.

[49] B. Lim and S. Zohren. “Time-series forecasting with deep learning: a survey”. In: *Philosophical Transactions of the Royal Society A* 379.2194 (2021), p. 20200209.

[50] M.Kaufinann. “Data mining: Concepts and techniques”. In: *Mining, What Is Data* 10 (2006), pp. 559–569.

[51] D. MacKenzie. *An engine, not a camera: How financial models shape markets*. Mit Press, 2008.

[52] J. Mann and J.N. Kutz. “Dynamic mode decomposition for financial trading strategies”. In: *Quantitative Finance* 16.11 (2016), pp. 1643–1655.

[53] P.S. Maybeck. "The Kalman filter: An introduction to concepts". In: *Autonomous robot vehicles* (1990), pp. 194–204.

[54] R.J. Meinhold and N.D. Singpurwalla. "Understanding the Kalman filter". In: *The American Statistician* 37.2 (1983), pp. 123–127.

[55] D.C. Montgomery, C.L. Jennings, and M. Kulahci. *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.

[56] K.M. Nagpal and P.P. Khargonekar. "Filtering and smoothing in an H/sup infinity/setting". In: *IEEE Transactions on Automatic Control* 36.2 (1991), pp. 152–166.

[57] E.I. Okoye and D.O. Gbegi. "Forensic accounting: A tool for fraud detection and prevention in the public sector.(A Study of Selected Ministries in Kogi State)". In: *Okoye, EI & Gbegi, DO (2013). Forensic Accounting: A Tool for Fraud Detection and Prevention in the Public Sector.(A Study of Selected Ministries in Kogi State). International Journal of Academic Research in Business and Social Sciences* 3.3 (2013).

[58] R.B. Pachori. *Time-Frequency Analysis Techniques and their Applications (1st ed.)* CRC Press, 2023.

[59] E. Panopoulou. "Predictive financial models of the euro area: a new evaluation test". In: *International Journal of Forecasting* 23.4 (2007), pp. 695–705.

[60] E. Parzen. *Stochastic processes*. SIAM, 1999.

[61] G.K. Pasricha. "Kalman filter and its economic applications". In: (2006).

[62] M. Pelikan, D.E. Goldberg, E. Cantú-Paz, et al. "BOA: The Bayesian optimization algorithm". In: *Proceedings of the genetic and evolutionary computation conference GECCO-99*. Citeseer. 1999.

[63] "Predicting stock market index using LSTM". In: *Machine Learning with Applications* 9 (2022), p. 100320. ISSN: 2666-8270.

[64] G. Priya and S. Saradha. "Fraud detection and prevention using machine learning algorithms: a review". In: *2021 7th International Conference on Electrical Energy Systems (ICEES)*. IEEE. 2021, pp. 564–568.

[65] S. Qian and D. Chen. "Joint time-frequency analysis". In: *IEEE signal processing magazine* 16.2 (1999), pp. 52–67.

[66] N.L. Ricker. "Model predictive control with state estimation". In: *Industrial & engineering chemistry research* 29.3 (1990), pp. 374–382.

[67] Z.A. Sadik, P.M. Date, and G. Mitra. "Forecasting crude oil futures prices using global macroeconomic news sentiment". In: *IMA Journal of Management Mathematics* 31.2 (July 2019), pp. 191–215. ISSN: 1471-678X.

[68] N.I. Sapankevych and R. Sankar. "Time series prediction using support vector machines: a survey". In: *IEEE computational intelligence magazine* 4.2 (2009), pp. 24–38.

[69] S. Shamim et al. "Big data management capabilities in the hospitality sector: Service innovation and customer generated online quality ratings". In: *Computers in Human Behavior* 121 (2021), p. 106777.

[70] L.J. Shaw and J. Narula. "Risk assessment and predictive value of coronary artery disease testing". In: *Journal of Nuclear Medicine* 50.8 (2009), pp. 1296–1306.

[71] S. Shen, H. Jiang, and T. Zhang. "Stock market forecasting using machine learning algorithms". In: *Department of Electrical Engineering, Stanford University, Stanford, CA* (2012), pp. 1–5.

[72] A.K. Singh. "Major development under Gaussian filtering since unscented Kalman filter". In: *IEEE/CAA Journal of Automatica Sinica* 7.5 (2020), pp. 1308–1325.

[73] G.A Terejanu. *Extended kalman filter tutorial. Online*. 2008.

[74] T.P. Trappenberg. *Fundamentals of Machine Learning*. Oxford University Press, Nov. 2019. ISBN: 9780198828044.

[75] G. Welch, G. Bishop, et al. "An introduction to the Kalman filter". In: (1995).

[76] G.F. Welch. "Kalman filter". In: *Computer Vision: A Reference Guide* (2020), pp. 1–3.

[77] C. Wells. *The Kalman filter in finance*. Vol. 32. Springer Science & Business Media, 1995.

[78] W. Whitt. "Stochastic-process limits: an introduction to stochastic-process limits and their application to queues". In: *Space* 500 (2002), pp. 391–426.

[79] H.P. Williams. *Model building in mathematical programming*. John Wiley & Sons, 2013.

[80] W. Wu, S. Nagarajan, and Z. Chen. "Bayesian Machine Learning: EEG\/MEG signal processing measurements". In: *IEEE Signal Processing Magazine* 33.1 (2015), pp. 14–36.

[81] Y.Chen and Y.Hao. "A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction". In: *Expert Systems with Applications* 80 (2017), pp. 340–355.

[82] Y. Yuanxi. "Robust bayesian estimation". In: *Bulletin géodésique* 65 (1991), pp. 145–150.

[83] M.E. Zmijewski. "Methodological issues related to the estimation of financial distress prediction models". In: *Journal of Accounting research* (1984), pp. 59–82.

[84] M.J. Zyphur and F.L. Oswald. “Bayesian estimation and inference: A user’s guide”. In: *Journal of Management* 41.2 (2015), pp. 390–420.