Department of Economics and Finance Chair Mathematics 2

Cryptocurrencies: return's analysis using Kalman filter

SUPERVISOR

Professor Marco Papi

CANDIDATE

Annagiulia Di Pasquale

ID number 198021

ACADEMIC YEAR 2017/2018

Table of contents

List of figures

1 Introduction

Cryptocurrencies have become an increasing trend during the last years. Many economists, banks and other financial institutions have started studying their performances but, most importantly, they have started studying which are the variables that might determine their price.
This thesis aims at studying, from a mathematical-statistical point of view, which are the factors that influence the virtual currencies' value, after having given a general overview about cryptocurrencies.
This paper is divided according to the following structure: Chapter one will provide an introduction from an historical point of view and it will analyse the main advantages of virtual currencies with respect to gold. Chapter two will instead give an overview over the Blockchain, that is the technology underlying cryptocurrencies whereas chapter three will discuss about the mathematical-statistical approach, the *Kalman* filter, that will be used in this dissertation so to understand whether in a system where disturbances occur, it is still possible to make a good prediction of the variable outcome after the error perturbed the system. Chapter four will provide the application of the aforementioned method to a data set composed by past prices. In order to do this, I will use a programming software called MatLab that will enable me to design a model and a consequent plot to finally analyse the outcome from a graphical point of view. The *Kalman* filter will be applied to a vector composed by Bitcoin's returns. The data set for past prices I will be using is not provided with Bitcoins' returns so, I will consequently compute the return for every period as the logarithm of the ratio of the $n+1^{th}$ price and the n^{th} price. The *Kalman filter* will be used at the end in order to analyse whether the incidence of external random rumours on Bitcoins' prices has a significant impact. Finally, Chapter five will be the appendix, so it will be entirely devoted to the codes I have used on MatLab.

1.1 From Fiat to Crypto currency

I am going to analyse the historical development of currencies, going from gold to virtual currencies (VCs), in order to asses why people would prefer them with respect to gold coins. The first assumption I need to start with is that many currency users prefer safe and anonymous transactions and all users prefer transactions that take place within a stable, safe and easy to use system. At first glance, VCs may appear distant from fiat coins often used as a standard to be compared to. VCs, in fact, have no physical dimension, no intrinsic value, and their value is not assessed by a governmental authority.

Since 700 BCE, gold coins have been used as a store of value, unit of account, and medium of exchange. You might be wondering why gold among so many elements that we might find in nature. Because gold, as a currency, has many satisfying properties: it has a market value and, most importantly, intrinsic value. Unlike shells that were used as a trading currency in West Africa; unlike beavers' skin that was used as a trading currency in the Hudson's' bay territory; unlike salt used as a trading currency in Europe, China and Ethiopia since the Roman empire (this is where the word *salary* comes from); unlike tobacco leaves and cocoa beans used as a trading currency in Central America during the XIV and XVII century respectively; unlike tea used as a trading currency in Mongolia and Siberia until the XX century (unbelievable!) and unlike calcareous stones used as a trading currency in Yap island in Micronesia, gold is *indestructible*: in fact, the supply of gold in the world has been plentiful enough to guarantee its use as a currency, but not so plentiful as to exhaust its value. If it were to choose another metal, such as platinum or aluminium, they are either too rare or too abundant to be used as currency in fact, in the case of platinum, the intrinsic value would be so high due to its scarcity while, in the case of aluminium, the intrinsic value would be so low due to its profusion.

Gold is also easily divisible, hence easily measurable. In fact, gold and silver coins' value is mainly given by their weight and their pureness even though they are issued by a government. Given this, a central authority is not necessary to establish the value of commodity money[1]. Beside this, another peculiarity of commodity money is its high anonymity: there is no register that keeps records or tracks or monitors transactions made between users. Commodity money's value has suffered of various fluctuations that were beyond the control of any monetary authority although most of commodity-based currencies have preserved stable values over the years. Why did these fluctuations occur? Because the value of a currency in general is determined by the interaction of supply and demand for that particular currency.

For instance, when silver deposits were discovered in South America around 1870, the increase in supply caused the value of silver with respect to the value of gold to fall by one half.

In addition to their value variability, commodity money is difficult to use for large scale, international or distant payments: their use can be reduced just to small and local transactions even because it is not appropriate to be carried around.

[1] Commodity money is composed of actual units of a particular freely-obtainable, non-monopolised commodity (or of warehouse certificates for actually existing units of the commodity) which happens to have been chosen for the familiar purposes of money, but the supply of which is governed – like that of any other commodity – by scarcity and cost of production (Keynes 1930, p. 7)

This is why, most countries decided to shift from commodity-based to paper (fiat) currencies: these currencies are declared to be legal tender by a central authority, have no face value and they can only be converted into a commodity only if the issuing authority decides so. Because of this, currencies' value depends upon users' trust in the central authority in maintaining the currency's value.
The main advantages belonging to fiat currencies over commodity money are:

- Weight: they are lighter, easier to use and they are a useful tool in the hands of governments for the realization of monetary and fiscal policy;
- Anonymity: fiat currencies can provide more anonymous transactions.

However, fiat money is not perfect, it has disadvantages as well. For instance, so to maintain its value and its stability, it is extremely dependent on its central authority. It can experience huge fluctuations, due to governments' macroeconomic policies, even becoming worthless (hyperinflationary episodes).

Nowadays, thanks to financial innovations, it is possible to conduct economic transactions that go far beyond the limits established by physical currency, just think about modern cheques. An ancestor of this instrument that appeared around the XIII century is the bill of exchange: it appeared to simplify trade and to avoid carrying large amounts of gold from country to country. According to the country in which they were issued, they were denominated in their country's currency (like a proper cheque). At that time, they were a proper innovation that allowed users to use traditional currency more efficiently.
More recent technological innovations have allowed users to shift from paper-based exchange systems, such as checks, to electronic systems, like swiping debit cards through a point-of-sale card reader, to using near-field communication (NFC) technology to enable radio communication through mobile-computing platforms (such as via applications on smartphones). As with bills of exchange, they are a real evolution for the whole economic system as they authorise clients to use traditional currency in a functional and practical way. However, unlike VCs, they do not constitute a new type of currency.

1.2 The evolution of Cryptocurrencies

VCs have become even more used in recent years. So far, no government has implemented a VC as its legal tender, even though it might symbolise value for that particular community that uses it as a mean of exchange. VCs have been adopted by many platforms especially online gaming communities and loyalty programs, like airline frequent-flier programs.

As every existing currency, VCs need to have the following characteristics in order to be considered a proper currency. The three features I am talking about are: store of value, unit of account and medium of exchange. They do possess all of them within their community of interest. Unlike physical currency, where people within their community of interest belong to the same country or union of countries (like USA or Europe), so to the same geographical area, the VCs' community of interest does not need to occupy a single geographical unit.

Some of the most recent VCs, such as Bitcoin, differ from the earlier versions of VCs as they are created to function as currency in the real economy and they can be exchanged for fiat currency.

Going back to the comparison with gold coins, Bitcoin has two common points with gold coins:

- Limited supply of currency available in the economy;
- Bitcoin's exchange rate can be volatile.

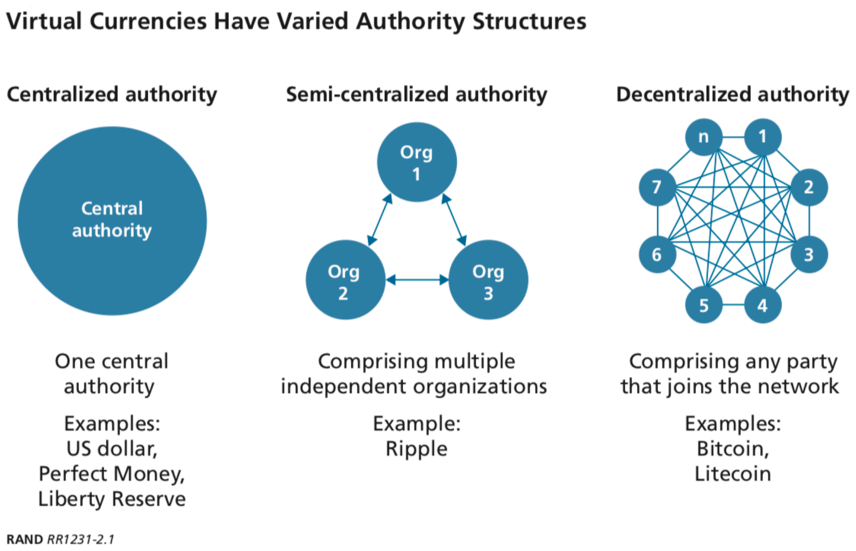

Figure 1. The different types of currencies and their decentralisation

Differently from gold, Bitcoin is easily measurable and divisible, easily transportable and does not need any kind of authorization to transit through international borders as currency, which may facilitate its use and, most importantly, reduce transaction costs. Finally, Bitcoin does not depend on a central authority to maintain its value.

The most important distinction between Bitcoin and previous VCs is that, while VCs do not need a central authority, Bitcoin's main peculiarity is its complete *decentralization*: many recently introduced VCs have followed Bitcoin exactly in this path. Current VCs are structured in a way that they range from having a complete centralization to a complete decentralization (see Figure 1).

After having analysed the monetary perspective of VCs, we will now determine the evolution of the VCs from a technological point of view.

2 The Blockchain

Before introducing the concept of Blockchain, we should make reference to few topics that normally do not have many points in common: first of all, the concept of trust and community, then cryptography, transparency, sharing and "competition" in the achievement of an objective. What is really important is the immutability over time of data and information and decentralization. All these concepts give rise to a complex and powerful innovation that is democratic and potentially supportive too.

According to some, the Blockchain technology is the new Internet generation: more precisely, it represents a sort of Internet of Transactions and for those that basically go beyond the concept of transaction, the Blockchain technology represents the future. Others believe that this technology is the virtual representation of trust and this is why someone believes that the Blockchain may become, in some sense, political, in that it may guarantee a new form of democracy that is truly decentralized and that safeguards the possibility of verifying and checking. Most importantly, it guarantees the creation of immutable archives that are totally transparent and for this reason corruption free.

However, the Blockchain technology *should not* be mistaken with the Bitcoin concept: they are interrelated in that the Blockchain is necessary for the Bitcoin to be exchanged but this is just one of the thousands use of the Blockchain. While the Bitcoin is a type of cryptocurrency, the Blockchain is the technology underlying Bitcoin and it is a platform for the management of transactions and exchanges of data also among sectors that are distant from the finance and payment sector. "Blockchain technology is challenging the status quo in a radical way: by using maths and cryptography, Blockchain provides an open and decentralised database of every transaction involving value as money, goods, property, work or even votes creating a record whose authenticity can be checked by the entire community […] so that *third party trust organization may no longer be necessary*." (Video "What is Blockchain?")

Moreover, Blockchain is so well encrypted that someone believes that in ten years' time it will be used to collect taxes so that people will exactly know for which purpose their taxes have been used. Because of this reason, due to its decentralization and encryption, Blockchain is *the* technology that allows the exchange of information and data on the Internet not only for what it concerns the payment system but also the exchange of information related to contracts, especially to Smart Contracts.

2.1 What Blockchain is and how it works

The Blockchain is a communication protocol that is based on the idea of a distributive database (a database where data are not saved on a single computer but they are saved on many computers connected among them, called *nodes*).

The Blockchain is a series of blocks that archive a set of validated and correlated transactions by a Timestamp. Every block is characterized by a *hash*, an algorithmic unconvertible function that connects a string of arbitrary length to a string of predetermined length. In this way, the block has been identified in an unambiguous way and that allows the connection with the previous block.

However, what are the components that create the Blockchain? First of all, we have *nodes* that are the participants of the Blockchain and they are physically constituted by those servers belonging to every single participant; in the second place we have *transactions* made up by data that represent the object of the physical exchange that have to be verified, approved and later archived; then we have *blocks* represented by the grouping of a set of transactions that have to be verified, approved hence archived by the participants of the Blockchain; furthermore, one more component that needs to be mentioned is the *ledger* that is the public register where all the realized transactions are "noted" in the most transparent, ordered, sequential and immutable way. The *ledger* is composed by the set of blocks that are connected among them through cryptography and *hash*. To conclude, we have the aforementioned *hash* that consists in a nonconvertible operation that allows to relate either a textual or numerical string of random length to a unique string of predetermined length. Thanks to this process, the *hash* allows to identify uniquely and safely every single block. Since a *hash* "transforms" the textual or numerical string into another, there should be no reference to the previous string that generated it.

Hence, every block contains different transactions and every block has its own *hash* located in the *header*. The *hash* registers all information related to the block and it is the *hash* with all information on the previous block that allows to create a chain and to connect blocks among them.

Every transaction, instead, contains information relative to the public address of the recipient, the transaction's features and the cryptographic signature that guarantees the authenticity of the transaction itself. Blockchain has to be seen as a public register that is sharable among all available clients or, in jargon, *nodes*.

At this point, a question would naturally pop up in your minds: how can I join the Blockchain? Is there any form I need to fill in in order to enter this transaction mechanism? Well, Blockchain has been organized in order to automatically refresh once a new client enters the system. Every realized transaction has to be automatically confirmed by every single node through cryptography softwares that are used to sign transactions in order to guarantee their digital identity.

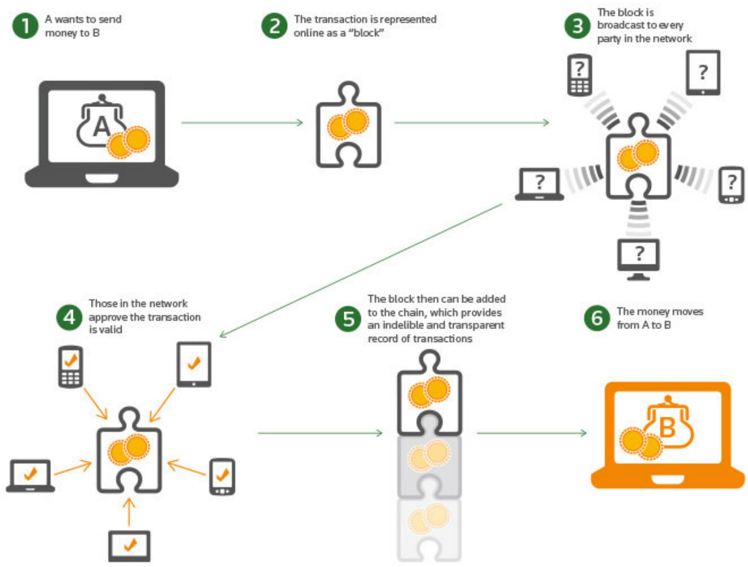

Figure 2. The payment system using the Blockchain technique

2.2 The Blockchain technique applied to Fintech

Before starting talking about Blockchain applied to Fintech, we should give an intuition of what Fintech is. Fintech had a great boost during the global crisis that dates back to 2008, when many people understood how slow the classical banking system was and how fast the Fintech sector is instead. It literally means “Techno-Finance” and it is concerned with the digitalization of the banking and financial system that uses technology in order to make the system more efficient. The Fintech includes many services, i.e.: crowdfunding, peer-to-peer lending, asset management, payment system management, credit-scoring, data collection, exchanges, digital currencies or Cryptocurrencies such as Bitcoins.

This is where I wanted to get: Fintech applied to Cryptocurrencies, hence to Blockchain. Nowadays, ten Central Banks, as Singapore Central Bank, are working on projects related to national Cryptocurrencies; seven Central Banks have started projects on Distributed Ledger systems for interbank transactions while nine institutes have commissioned studies in order to better understand the topic. Among them, we can find some Italian banks such as *Intesa Sanpaolo*, *Unicredit* and *Banca Mediolanum* that are part of a consortium that is willing to develop Distributed Ledger systems to settle interbank relations.

Allianz, the German insurance company, has been the first one to propose a service using the Blockchain technology. The project includes cash payments, real time access to information related to the transaction and an easy to use interface. So far, results demonstrate that Blockchain technology can improve the efficiency of insurance transactions at the international level. The role of the Blockchain, in this context, is to automatically connect all involved parties in the insurance program. As we have already seen, Blockchain is like a financial book shared among a network of participants that is able to record transactions and data. Updates or modifications are shared in real time among all users. In this way, it is easy to obtain a faster, more transparent, safer and more efficient method to provide information, to elaborate and to register commercial transactions among all parties. In the special case of Allianz, the insurance Blockchain prototype speeds and facilitates regular transactions and transfers of money between insurers and clients. Moreover, the entire process is transparent and it can be monitored in real time.

This is just one of the thousands examples I could quote: this is because between 2012 and 2015 (so, just in *three years' time*), the amount of investments in the Fintech sector has increased from two million dollars to one billion. Today, even though most of the applications of the Blockchain are related to payments, this technology can be found in other financial sectors such as trading and capital markets with the Nasdaq being the prevailing entity that monitors this sector. What is really appreciated about Blockchain is that, since everything is transparent, digitalized, encrypted and open, there is no possibility to evade taxes in those countries in which this technology is applied.

To conclude, according to Santander bank, in the next years, Blockchain will not only be applied to the Fintech sector but there could be at least nineteen sectors of the economy that will employ these models to take advantage of the digital revolution. Among these nineteen sectors we find insurance companies, digital payments, agri-food industry, manufacturing (Industry 4.0), IoT (the Blockchain finds a wide application in the Internet of Things thanks to its facility to exchange data as it could be used to facilitate the communication among connected IoT objects beside making the exchange faster and safer), health care, public administration and finally retail so to make payments faster and cheaper.

2.3 The Blockchain technique applied to Cryptocurrencies

So far we have discussed the Blockchain technology that is used in many sectors especially the Fintech. Most of the times the term Blockchain is misused: it is employed when we talk about Cryptocurrencies but it is *not* a synonym of Cryptocurrencies. As aforementioned, Blockchain is the technology underlying bitcoins but this is just one of the uses that Blockchain has. Recently, The

Economist has defined it to be the "trust machine" due to its high degree of safety, its decentralization, its transparency and its precision.

At the beginning, in 2008, Satoshi Nakamoto, pseudonym of a guy that first introduced the Bitcoin, realized a P2P protocol that in the years has been used especially by hackers, activists and in the best case scenario by speculators. However, it has been so disruptive that Bitcoin has now convinced the most conservative analysts too.

The bitcoin digital currency adopts the peer-to-peer technology which designates a model of architectural logic where the network's nodes can achieve both the functions of client and server with respect to the other terminal hosts. Due to its decentralization, it does not need neither authorities nor central institutions: Bitcoins are issued on the network and the management of transactions is also governed by the network itself. It is basically a public operation and anybody who wants to participate in the project can just adhere and take part in it. How is this possible? How can I just decide to participate and as soon as I choose I am part of this project? The idea behind what Satoshi Nakamoto created is the open source software where the development, the management and the update are all public and shared among users. According to Satoshi Nakamoto and his White Paper "the Bitcoin network neither belongs to nor it is controlled by anybody, in other words it belongs and it is controlled by whoever wants to take part in the project." According to me, it is something more than just a mere project: it is something revolutionary that neither Satoshi Nakamoto realized what he did when he first invented it. From a technical point of view, Bitcoin is an online communication protocol that enables users to employ virtual currencies in their daily activities including electronic transactions. Since it was first discovered in 2008/2009, Bitcoin has been used for almost 305 million transactions (source: blockchaininfo.info last estimation: 16/03/2018). These transactions however are not recorded on

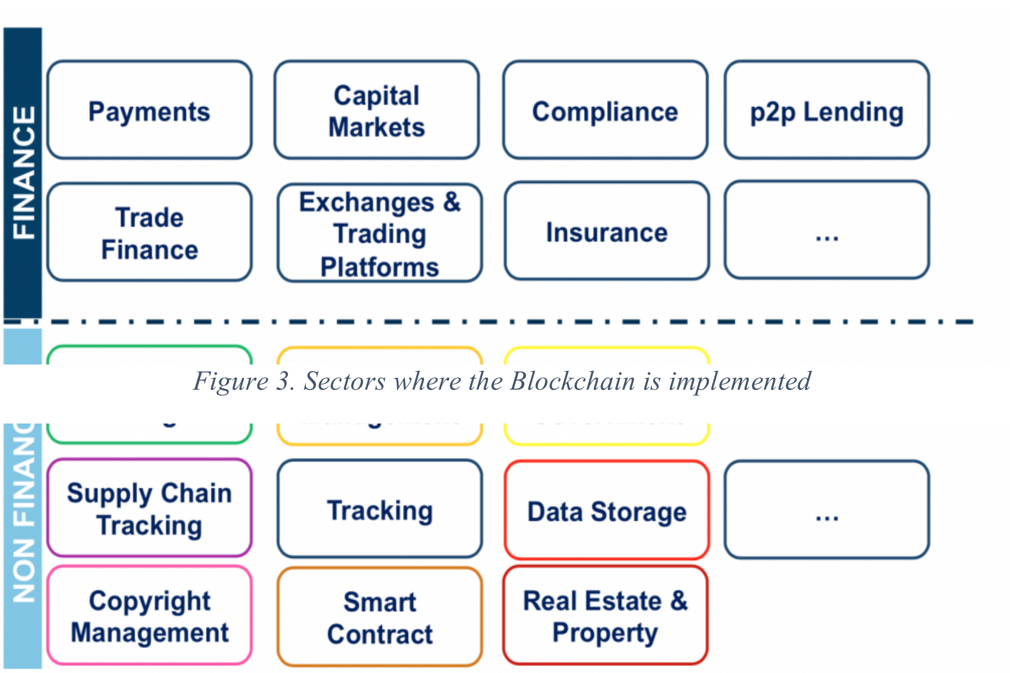

Figure 3. Sectors where the Blockchain is implemented

individual servers but they are rather recorded on a transaction log, where the Bitcoin is built, which is distributed over a network of involved computers. Bitcoin is built in such a way that it rewards users for their honest behaviour and most importantly it avoids that power will be concentrated in the hands of just one single entity. Because of this characteristic, Bitcoins have a con at the same time as, while traditional currencies have a Central Bank regulating their issuance, Bitcoins have not. So this may be an advantage and a disadvantage at the same time because issuing currency and verifying transactions becomes more difficult than in traditional cases.

Transactions made with Bitcoins are not reversible and another feature of this virtual currency is that it basically has a fixed supply, so when the amount of Bitcoins will be entirely distributed, there will be no more newly issued Bitcoins available. It is still true that those existing Bitcoins could be traded but, for those goods that have a fixed supply, they are deflationary constructed meaning that when the amount of Bitcoins will entirely be allocated, nobody will be willing to hold them as they are basically worth nothing.

Why have people been so concerned about them in the past few years? Because, due to its construction, it may disrupt the existing payment and even monetary system.

Now, we are going to take a look at how Bitcoins work in practice. First of all, since Bitcoins should be used as a medium of exchange, we first have to understand what a transaction is: a transaction is composed by two people exchanging a good or service and, the one that is *receiving* the service pays the one that is *selling*. In this case, the mean of payment is Bitcoin and every transaction has a different encryption code, called Cryptographic Key, so that it is safe from external hackers and users. In order for the technology to operate in a fast and efficient way, the transactions are grouped until a certain number is reached so to form a block. A set of blocks becomes the Blockchain.

Now, let’s see a practical application during the purchase of a house: suppose two individuals, Luke and Lucy, where Luke wants to sell his house to Lucy while, on the other hand, Lucy wants to purchase Luke’s house. Instead of using traditional currency, they want to use cryptocurrency, in our particular case, Bitcoins. In this way a transaction constituted by a set of elements such as the public address of the receiver, information relative to the transaction itself and the Cryptographic Keys is set up. In our particular example, the transaction includes information on the real estate, on the price, on Lucy’s financial liquidity, on Luke’s actual ownership of the property and other types of information that are necessary to carry out the transaction.

Therefore, a new block that contains all data about the transaction between Luke and Lucy is created. As previously stated, the block contains other transactions as well that will be then submitted to the other participants of the Blockchain in order to be verified and later approved. Once the block has been verified, it is added to the rest of the *chain of blocks* (Blockchain) that is contained in the participants' archive and it can be accessed by all of them. Once information is verified, the transaction is validated and carried out. At that point, the transaction is part of a newly created block.

Think about another scenario: suppose we have three individuals, Eric, Francis and Garrett, where Eric transfers 7 Bitcoins to Francis. Even though this transaction is not recorded in any book, it is verifiable through the encryption key assigned to every party involved in the transaction. The following table gives you an intuition of the Bitcoins' flow between the parties:

LEDGER

Name	***Balance***
Eric	7 BTC
Francis	2 BTC
Garrett	1 BTC

Eric transfers 7 BTC to Francis →

LEDGER

Name	***Balance***
Eric	0 BTC
Francis	9 BTC
Garrett	1 BTC

Figure 4. Table representation of a transaction involving Bitcoins

The transaction, as already said, needs to be verified hence approved by the peer-to peer network. As soon as it is validated by the network, the transaction is recorded in the public register, the ledger. In this way, anyone on the peer-to-peer network has been updated.

In the case in which, Francis wants to transfer the same amount of Bitcoins to Garrett, the same procedure applies. When the transaction has to be verified by the network, the order must go through the transaction chain in order to check whether Francis disposes of those 7 Bitcoins to be transferred to Garrett. Once Francis's account has been approved, the new transaction either forms a new block or it is added to an existing one and, these blocks gathered together form the Blockchain (as previously stated). However, how are these blocks added to the network? They are added through the usage of *miners*. Miners perform the function of *mining* so they basically solve a very complex, both in terms of power and processing capacity, mathematical algorithm. In this way the blocks made up of transactions are valid and encrypted.

All blocks are like puzzle pieces, so they match between one another and just one piece is the exact link of the other. This is done in order to avoid unauthorized transactions to be unlawfully added and to create a chain where all pieces match between themselves. However, before new transactions are added to a new block, they are pooled all together until they are verified and confirmed. Since, at this point, several transactions can be added to the Blockchain, this could constitute a problem. In order to avoid this issue, the network only accepts those transactions constituting the longest chain. Those transactions belonging to the shorter chains are sent back to the pool of unconfirmed transactions so that they can be processed once again.

3 Statistical analysis

So far, we have been dealing with the practical aspect of Cryptocurrencies, from its history to its functioning passing through the technology underlying them, known as Blockchain, and analysing the sectors where it is and where it will be employed the most. However, this was just an introductory section in order to let the reader fully understand in which context I am going to operate. The scope of this thesis is to analyse whether there is a model through which I can fairly predict a trend or a pattern in Bitcoins' returns. The technology I am applying is the *Kalman filter* which is mainly used in engineering data prediction models. Here, the filter is constructed in such a way that, in the end, it ends up in the maximum likelihood estimation.
An obvious question might pop up in the reader's mind: why do I use a filter at all? What kind of information am I interested in? The idea behind this method is to deduce important information from a signal, neglecting superfluous knowledge. Moreover, I decided to use Kalman filter estimation as it is one of the few models that takes into account the random nature of measurements. Since this randomness has a statistical nature, I can solve the problem by using *stochastic* methods.

3.1 State space derivation

I start by introducing a *state-space model* necessary to conduct my analysis. A state-space model is built in such a way that contains enough equations in order to firstly estimate the model and, later, control it. It is useful because it allows to transform an abstract analysis into a more comprehensible one.

In the model, I considered two equations: the first one called the *state-vector equation* of the process for which I do not have information and the second one will be the *observation vector*, which is the estimate of *x* at time *k*. In formulae,

$$x_{k+1} = \phi x_k + w_k \tag{3.1}$$

$$z_k = H x_k + v_k \tag{3.2}$$

where, in equation 3.1, x_k is the state vector of the process at time *k* also known as *trend*; ϕ is the $n \times n$ state transition matrix of the process from the state at *k* to the state at *k+1* and it is assumed to be stationary over time; w_k is the *noise* process with known covariance. For what it concerns equation 3.2, z_k is the observed value of *x* at time *k*; *H* is the $m \times n$ connection matrix between the state and the observation vector z_k and it is constant over time; v_k is the measurement error with known covariance and is uncorrelated from the error of the process.

The ϕ matrix is a diagonal matrix that presents an extra 1 on the entry α_{12}: this is because the vector x_k is a bi-dimensional vector where the two entries are respectively $\mu_k$and ν_k. μ_k represents the local level component: you can think of it as if it were the intercept with the only peculiarity of being able to change over time (this is why you add the specification *k*). ν_k represents the angle of the trend line which also varies through time. The equations of μ_{k+1} and ν_{k+1} are represented as follows

$$\mu_{k+1} = \mu_k + \nu_k + \varepsilon_{\mu,k} \tag{3.3}$$

$$\nu_{k+1} = \nu_k + \varepsilon_{\nu,k} \tag{3.4}$$

where the two error components, $\varepsilon_{\mu,k}$ and $\varepsilon_{\nu,k}$, have an approximately normal distribution with mean 0 and variance σ_μ^2 and σ_ν^2 respectively. Notice that, in order to make a better estimation, μ_{k+1} and ν_{k+1} depend upon their past values. If I sum these two vectors together, at the end I obtain a third vector that is x_{k+1} (the initial state vector equation). So, the vector x_{k+1} is a linear combination of the two vectors (3.3) and (3.4). In formula,

$$x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_k \\ \nu_k \end{bmatrix} + w_k \tag{3.5}$$

where w_k will be a vector with entries $\begin{bmatrix}\varepsilon_{\mu,k}\\ \varepsilon_{\nu,k}\end{bmatrix}$, approximately distributed with mean 0 and variance Q. Q is the error variance-covariance matrix where, on its main diagonal I will find the variances and in the other cells I will find the covariances. In numbers,

$$Q = \begin{bmatrix} \sigma_\mu^2 & \sigma_\mu \sigma_\nu \rho_{\mu,\nu} \\ \sigma_\mu \sigma_\nu \rho_{\mu,\nu} & \sigma_\nu^2 \end{bmatrix} \tag{3.6}$$

where $\rho_{\mu,\nu}$ is the correlation between $\varepsilon_{\mu,k}$ and $\varepsilon_{\nu,k}$.

3.2 The Observer Design Problem

Since it is difficult to assess or estimate the internal states of a linear system given that you just have access to the final *outcome*, I define the Observer design problem. This is why people refer to this estimation as if it were a kind of "closed box" as you can just observe the output that comes out of the box but you cannot determine what happens inside of it.

Indeed, I now move to the *observation vector* that is the estimate of x at time k. Before talking about the vector z_k, however, I first need to introduce the vector Y. The vector Y is an observation vector, so it is based on past data about bitcoins' return. For the sake of simplicity and, in order to give the estimation a more real effect, I base the estimate of Y_{k+1} only upon Y_k and Y_{k-1}. In the end I have,

$$Y_{k+1} = a\, Y_k + b\, Y_{k-1} + \mu_k + \varepsilon_{y,k} \tag{3.7}$$

where $Y_{k+1} - a\, Y_k - b\, Y_{k-1} = z_k$. So, eventually, I will have $z_k = \mu_k + \varepsilon_{y,k}$ that is the observation vector. μ_k is just the product between matrix H and vector x_k, where matrix H is a 1x2 matrix with entries [1 0] respectively while, vector x_k is the starting state vector equation I used to explain the previous equation too. The error of equation 3.7, $\varepsilon_{y,k}$, is an approximately normally distributed prediction error term with mean 0 and variance σ_y^2. In equation 3.7 and, in a simplified way in equation 3.2, the covariance of the noise is assumed to be stationary over time, that is our term

$$R = E[\nu_k \nu_k^T] \tag{3.8}$$

Since I have written a lot about the uncertainty of measurements, I should devote some lines to explain which are the sources of noise that cause these measurements to be unreliable. For instance, every sensor has its own limitations related to the physical medium linked to it. Using electrical devices might cause erroneous estimates as using sensor and electrical circuits might add the electrical noise attached to them affecting the size and the quality of measurements. This is why, I should select the estimates correctly and interpret them as part of an overall sequence.
Last issue that I have to consider when making this estimation is that long run predictions cannot be made hence the observation vector is only based on short term evidence. In fact, this might cause predictability in future measurements.

3.3 The Kalman Filter

So far, I have been discussing about the statistical analysis I will conduct but, I have not devoted enough time to explain in detail the filter I will be using throughout the estimation. As aforementioned, I will employ the *Kalman filter* estimation for the analysis of Bitcoins' returns as it is one of the few models that takes into account the randomness of data.
When having a physical system, in order to solve it, it is better to develop a suitable mathematical model in order to adequately represent the physical situation. In order to do this, I have fundamental laws and control theories that might help me in solving mathematical models representing my physical system. However, as we know, fundamental laws and control theories have shortcomings. In particular:

I. no mathematical model adequately represents the actual physical system: in this way it does not take into account all features that characterise the reality. As we all know, mathematical methods only *approximate* real effects and do not *truly represent* them and this leads to uncertainty;
II. some real effects can neither be truly modelled nor controlled because they are disturbed by external sources of error that can neither be predicted nor controlled;
III. sensors, used for the estimation and data, do not provide all the information we want to have resulting in incomplete and imperfect measurements: this is because most devices are not planned to generate such information or because the cost to acquire such information is too high.

At this point, an important question becomes natural: how can you construct a model that takes into account for these noises and uncertainties that are inevitable in our mathematical system?
The *Kalman filter* is one of the most well-known and very often-used tool first introduced in 1960 and named after his discoverer Rudolph Kalman who gave a final solution to the data filtering problem.
Why is it one of the most well-known and very often-used filter? Because, as a tool, it incorporates all necessary data that can be supplied to it. In fact, the Kalman filter handles all available data, even though they are not so precise, as it makes use of knowledge of the system and measurement device dynamics; it is provided with information about the statistical distribution of the system errors and uncertainties and with eventual information about initial conditions.
The filter is conceived to be a *data processing algorithm* as if it were a computer program. As you can see from the figure below, this is how it works: the system is drawn using some control variables and the sensors of other electrical measuring apparatus provide the values for other data. This is intuitive when you have all the variables at disposal. When the variables of interest *cannot be* estimated, the help of a filter is necessary.

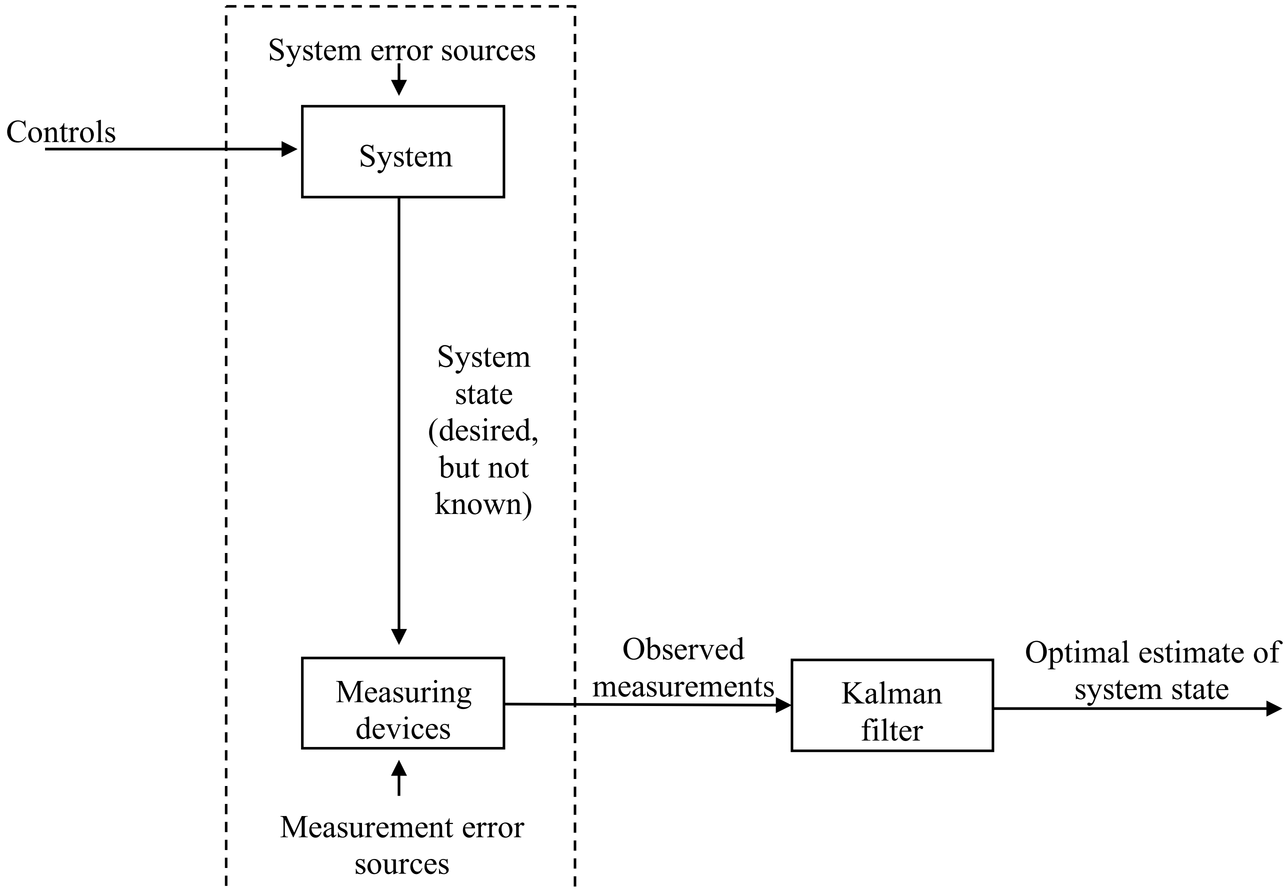

Figure 5. Schema process of the Kalman filter

Furthermore, most of the times, when systems are driven by inputs other than our own control variables, they might be inaccurate and might present noises and errors. This is why I implement the Kalman filter as it combines all available measurements and prior knowledge about the mathematical model so to produce an estimate of the unknown variables by statistically minimizing the error[2].

3.3.1 The Discrete Kalman Filter

After having introduced the state space derivation and how I defined the *state-vector* and the *observation vector equations*, I apply the aforementioned *Kalman filter*.

I define $\hat{x}_{k|k-1} \in \mathfrak{R}^n$ to be my *a priori* estimate at time *k* given measurement of *x* at time *k-1* and $\hat{x}_{k|k} \in \mathfrak{R}^n$ to be my *a posteriori* state estimate at time *k* given measurement x_k. The errors of the *a priori* and of the *a posteriori* estimates will therefore be

$$e_{k|k-1} \equiv x_k - \hat{x}_{k|k-1} \tag{3.9}$$

$$e_{k|k} \equiv x_k - \hat{x}_{k|k} \tag{3.10}$$

The *a priori* and the *a posteriori* estimate error covariances of *x* will hence be

$$\Sigma_{k|k-1} = E[e_{k|k-1}e_{k|k-1}{}^T] \tag{3.11}$$

$$\Sigma_{k|k} = E[e_{k|k}e_{k|k}{}^T] \tag{3.12}$$

Since I have to derive the equations for the final application of the Kalman filter, I need to find an equation that computes an *a posteriori* estimate $\hat{x}_{k|k}$ as a linear combination of the *a priori* estimate $\hat{x}_{k|k-1}$ and of the weighted difference between the actual measurement z_k and a measurement prediction $H\hat{x}_{k|k-1}$. In formulae,

[2] There exists a lot of literature about Kalman filter. I have given a general and introductory idea just to introduce the reader to the method I will adopt from this moment onwards. A much more in depth discussion about the topic has been given by Sorensen in 1970 or by Gelb in 1974 or, to mention someone else, Maybeck in 1979.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\,(z_k - H\hat{x}_{k|k-1})^3 \tag{3.13}$$

The weighted difference between the actual measurement z_k and the measurement prediction $H\hat{x}_{k|k-1}$ is called the *measurement innovation* or, in statistical jargon, *residual*. As in statistics, if this difference is either positive or negative, the prediction differs from the actual measurement so I will have errors in prediction (either an overestimation or underestimation); if this difference is equal to zero, then the two values coincide and I will have no discrepancy between the actual measurement and the prediction (the estimate is equal to the real value).

The $n \times m$ matrix K is set in such a way that minimises the *a posteriori* error covariance depicted in equation 3.12. How do you set K in such a way that minimises the *a posteriori* error covariance?
If I substitute equation 3.2 into 3.13, I obtain the following equation

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\,(Hx_k + \nu_k - H\hat{x}_{k|k-1}) \tag{3.14}$$

Replacing the above formula, equation 3.14, into the equation for the second moment of $\Sigma_{k|k}$, quoted in the footnote of the previous page, it results in

$$\Sigma_{k|k} = E\,[\left((x_k - \hat{x}_{k|k-1})(I - HK_k) - K_k\nu_k\right)\left((x_k - \hat{x}_{k|k-1})(I - HK_k) - K_k\nu_k\right)^T] \tag{3.15}$$

The first term in brackets, on the right side of the equal, is equation 3.9 so the *a priori* estimate of the error at time *k*. Since it is uncorrelated with the measurement noise, I can rewrite equation 3.15 in the following way

$$\begin{aligned}\Sigma_{k|k} = {} & (I - HK_k)\,E\,[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^T](I - HK_k)^T + K_kK_k^TE[\nu_k\nu_k^T] = \\ & = (I - HK_k)\,\Sigma_{k|k-1}(I - HK_k)^T + K_kK_k^TR\end{aligned} \tag{3.16}$$

[3] The explanation to this equation derives from Bayes' formula of conditional probability where the *a priori* estimate $\hat{x}'_k$ is conditioned on all prior measurements z_k. The Kalman filter respects the first two moments of the state distribution $E[x_k] = \hat{x}_{k|k}$ and $E[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^T] = \Sigma_{k|k}$.
The *a posteriori* estimate error covariance depicted in equation 3.12 reflects the variance of the state distribution: if you compute the conditional probability of x_k and z_k, it will result in a normal random variable approximately distributed with mean $\hat{x}_{k|k}$ and variance $\Sigma_{k|k}$.

The very last part of equation 3.16 is the updated version of the error covariance matrix. The matrix, on the main diagonal, contains the mean squared errors:

$$\Sigma_{kk} = \begin{bmatrix} E[e_{k|k-1}e_{k|k-1}^T] & E[e_{k|k}e_{k|k-1}^T] & E[e_{k|k+1}e_{k|k-1}^T] \\ E[e_{k|k-1}e_{k|k}^T] & E[e_{k|k}e_{k|k}^T] & E[e_{k|k+1}e_{k|k}^T] \\ E[e_{k|k-1}e_{k|k+1}^T] & E[e_{k|k}e_{k|k+1}^T] & E[e_{k|k+1}e_{k|k+1}^T] \end{bmatrix} \tag{3.17}$$

The *trace* of a matrix is the sum of the terms that lie on its main diagonal. In this particular case, the trace will be the sum of the mean squared errors. Hence, the mean squared error can be minimised by minimising the trace of $\Sigma_{k|k}$ which will in turn minimise the trace of Σ_{kk}.

In order to minimise the trace of $\Sigma_{k|k}$, I first have to compute the derivative of $\Sigma_{k|k}$ with respect to K_k and set the result to zero so to find its minimum.

After having expanded equation 3.16, I will obtain

$$\begin{aligned} \Sigma_{k|k} &= \Sigma_{k|k-1} - \Sigma_{k|k-1}K_k^T H^T - \Sigma_{k|k-1}K_k H + \Sigma_{k|k-1}K_k K_k^T H H^T + K_k K_k^T R = \\ &= \Sigma_{k|k-1} - \Sigma_{k|k-1}K_k^T H^T - \Sigma_{k|k-1}K_k H + K_k K_k^T (\Sigma_{k|k-1} H H^T + R) \end{aligned} \tag{3.18}$$

The trace of matrix Σ_k, given that *the trace of a matrix is equal to the trace of its transpose*, will therefore be

$$T[\Sigma_k] = T\big[\Sigma_{k|k-1}\big] - 2T\big[\Sigma_{k|k-1}K_k H\big] + T[K_k K_k^T \big(\Sigma_{k|k-1} H H^T + R\big)] \tag{3.19}$$

Differentiating equation 3.19 with respect to K_k will result in

$$\frac{dT[\Sigma_k]}{dK_k} = -2\big(\Sigma_{k|k-1}H\big)^T + 2K_k\big(\Sigma_{k|k-1}HH^T + R\big) \tag{3.20}$$

Setting the above result to zero and rearranging it, it will show the value of K such that minimises the *a posteriori* error covariance

$$K_k = \Sigma_{k|k-1}H^T\big(\Sigma_{k|k-1}HH^T + R\big)^{-1} = \frac{\Sigma_{k|k-1}H^T}{\Sigma_{k|k-1}HH^T + R} \tag{3.21}$$

As the measurement error covariance R goes to 0, the term K_k, since it represents the *Kalman gain* or the heaviness with which the residuals will be weighted, will weight the residual more heavily. That is,

$$\lim_{R_k \to 0} K_k = H^{-1}$$

Another way of thinking of the case when R approaches 0 is that, since R is the variance of equation 3.7, hence of equation 3.2, if R almost equals 0, it means that the actual measurement of z_k is more reliable than the predicted measurement $H\hat{x}_{k|k-1}$.
If I instead reason on the $\Sigma_{k|k-1}$ term, as it approaches 0, the gain K goes to 0 too. In formulae,

$$\lim_{\Sigma_{k|k-1} \to 0} K_k = 0$$

Another way of thinking of the case when $\Sigma_{k|k-1}$ approaches 0 is that, since $\Sigma_{k|k-1}$ is the *a priori* estimate error covariance, if $\Sigma_{k|k-1}$ equal 0, it means that the estimated measurement $H\hat{x}_{k|k-1}$ will be more reliable than the actual measurement z_k.

Going back to equation 3.21, in order to fully understand the filter, I replace it into equation 3.18. It will result in

$$\begin{aligned} \Sigma_{k|k} &= \Sigma_{k|k-1} - \Sigma_{k|k-1} H^T \left(\Sigma_{k|k-1} H H^T + R\right)^{-1} \Sigma_{k|k-1} H = \\ &= \Sigma_{k|k-1} - K_k \Sigma_{k|k-1} H = (I - K_k H)\Sigma_{k|k-1} \end{aligned} \tag{3.22}$$

The above result is essential for the filter implementation: equation 3.22 is the Kalman filter measurement update equation for the error covariance matrix with optimal gain (when K_k has been minimised). Equations 3.13, 3.21 and 3.22 will be necessary for our Kalman filter implementation and, most importantly, necessary to develop an estimate of the variable x_k.
The state projection will be obtained using the equation below

$$\hat{x}_{k+1|k} = \phi \hat{x}_{k|k} \tag{3.23}$$

What about the error term? It is sufficient to find an equation that transfers the error covariance matrix into the following time period, *k+1*. First of all, I construct an equation for the previous error, that is:

$$e_{k+1|k} = x_{k+1} - \hat{x}_{k+1|k} = (\phi x_k + w_k) - \phi \hat{x}_{k|k} = \phi e_{k|k} + w_k \tag{3.24}$$

Expanding equation 3.11 to time *k+1* will give

$$\Sigma_{k+1|k} = E\left[e_{k+1|k} {e_{k+1|k}}^T\right] = E\left[\left(\phi e_{k|k} + w_k\right)\left(\phi e_{k|k} + w_k\right)^T\right] \text{[4]} \tag{3.25}$$

Since $e_{k|k}$ and w_k have a zero cross-correlation as the error w_k accumulates between time *k* and time *k+1* while the error e_k is the error up to time *k*, I have

$$\Sigma_{k+1|k} = E\left[e_{k+1|k} {e_{k+1|k}}^T\right] = E\left[\phi e_{k|k}\left(\phi e_{k|k}\right)^T\right] + E\left[w_k w_k^T\right] = \phi \Sigma_{k|k} \phi^T + Q \tag{3.26}$$

This is the end of the analytical derivation of the Kalman filter. In the next paragraph I will give an intuition of what is the difference between the time update and measurement update equations, as I have mentioned them few times in this chapter and, I will provide two summarizing tables that will report the key equations I have just found. The two tables will be divided according to the specification of time update and measurement update equations.

3.3.2 The Kalman Filter Algorithm

The filter evaluates the process state at a point in time and then uses feedbacks in the form of uncertain measurements. This is why I can classify Kalman filter equations into two groups: *time update* and *measurements update*. The former equations are used to cast the current state forward and the latter equations are used to obtain the *a priori* estimates for what is expected to occur next. The latter are also necessary to obtain the feedbacks: for instance, they are used to asses whether a new measurement should be added to the *a priori* estimate in order to obtain an enhanced *a posteriori* version of the estimate later.

[4] The explanation to the following relationship, $\Sigma_{k+1|k} = E\left[\left(\phi e_{k|k} + w_k\right)\left(\phi e_{k|k} + w_k\right)^T\right]$, can be found in footnote 3 on page 16. In that case, however, it demonstrated the relationship between $\Sigma_k = E\left[\left(x_k - \hat{x}_{k|k}\right)\left(x_k - \hat{x}_{k|}\right)^T\right]$.

The *time update* equation can be considered as a *predictor* equation while the *measurements update* equation is considered as a *corrective* equation.

The table below will differentiate between the *time update* and the *measurements update* equations so that the reader will become more familiar with this classification:

Kalman filter time update equations

$$\hat{x}_{k+1|k} = \phi \hat{x}_{k|k} \quad (3.23)$$

$$\Sigma_{k+1|k} = \phi \Sigma_{k|k} \phi^T + Q \quad (3.26)$$

Notice, as said before, that the time update equations in the above table relate the current state to the forward state: from state *k-1* to state *k*. Matrix ϕ is equal to the one I introduced in equation 3.1 while matrix Q is the error variance-covariance matrix introduced in equation 3.6.

Kalman filter measurements update equations

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \left(z_k - H \hat{x}_{k|k-1} \right) \quad (3.13)$$

$$K_k = \Sigma_{k|k-1} H^T \left(\Sigma_{k|k-1} H H^T + R \right)^{-1} \quad (3.21)$$

$$\Sigma_{k|k} = (I - K_k H) \Sigma_{k|k-1} \quad (3.22)$$

The first thing to do is to compute what I called the gain, K_k. In this way, the process will be estimated so to obtain z_k. Therefore, the *a posteriori* state estimate can be generated by using the measurement for the gain K_k as I did in equation 3.13. Equation 3.22 is just the way through which the *a posteriori* error covariance is estimated.

This process does not end here, it will be iterated and every time previous *a posteriori* estimates will be used to forecast the latest *a priori* estimates. This recursive feature of the Kalman filter is what makes it appealing with respect to other filters: it constantly conditions actual estimates on past data.

Time Update (prediction)

1. Make a projection of the state
$$\hat{x}_{k+1|k} = \phi \hat{x}_{k|k}$$
2. Make a projection of the error covariance
$$\Sigma_{k+1|k} = \phi \Sigma_{k|k} \phi^T + Q$$

Measurement Update (correction)

1. Compute the Kalman gain, K_k
$$K_k = \Sigma_{k|k-1} H^T (H \Sigma_{k|k-1} H^T + R)^{-1}$$
2. Update the estimate with the measurement z_k
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K \left(z_k - H \hat{x}_{k|k-1} \right)$$
3. Update the error covariance
$$\Sigma_{k|k} = (I - K_k H) \Sigma_{k|k-1}$$

The above table depicts all the passages that should be implemented when applying the Kalman filter: it joins the formulae already pointed out in the two above tables and the concept of time update and measurement update equations.

3.3.3 The Maximum Likelihood Estimation

So to conclude my analysis on Bitcoins' returns, I still need to tackle another topic: the maximum likelihood estimate. The maximum likelihood estimate is obtained by finding $\hat{x}$ such that it maximizes the probability of y, meaning that it maximises the probability of having obtained the given sample over the parameter space. Let $\psi \in \mathbb{R}^d$ be the vector of unknown parameters that belongs to the parameter space Ψ. The matrices of the state space model along with all the variances I have used so far depend on ψ.

The likelihood function of the state space model will hence be:

$$l(y, \psi) = p(y_N, y_{N-1}, y_{N-2}, \dots, y_1; \psi) \tag{3.27}$$

The above equation, the likelihood function, depends upon the joint density of the observable data, $y = (y_N, y_{N-1}, y_{N-2}, \dots, y_1)$, and upon the vector made of unknown parameters ψ. It reflects how likely it would have been to have observed the data if ψ were the true values of the parameters.

Applying the definition of conditional probability and using Bayes' theorem, I can write the joint density as a product of conditional densities. In formulae,

$$l(y, \psi) = p(y_N |\, y_{N-1}, y_{N-2}, \dots, y_1; \psi) \cdot \dots \cdot p(y_k |\, y_{k-1}, y_{k-2}, \dots, y_1; \psi) \cdot \dots \cdot p(y_1; \psi) \quad (3.28)$$

The last term of equation 3.28 should be $p(y_1|y_0; \psi)$. In a Markovian system, as this is, future values of y_l ,when $l > k$, are functions of $(y_k, y_{k-1}, y_{k-2}, \dots, y_1)$ basing my estimation on current values of y_k. The above equation can be rewritten so that it depends upon the most recent observations, hence it will be equal to

$$l(y, \psi) = p(y_N |\, y_{N-1}; \psi) \cdot \dots \cdot p(y_k |\, y_{k-1}; \psi) \cdot \dots \cdot p(y_1; \psi) \quad (3.29)$$

But our question is still left unanswered: how do I estimate the parameter vector ψ? It will be evaluated by using the likelihood function expressed in terms of the prediction error, v_k, which is the same as the conditional variance of y_k:

$$Cov\ (v_k) = \ Cov\ (y_k) \quad (3.30)$$

In this way, by expressing the conditional variance of the two variables, I can state the density function. The density function of $p(y_k |\, y_{k-1}; \psi)$ is a Gaussian Normal distribution with conditional mean equal to

$$E[y_k] = \ H_k \hat{x}_{k|k-1} \quad (3.31)$$

Having a conditional variance equal to

$$Cov\ (y_k) = K_{k|k-1} = \Sigma_{k|k-1} H H^T + R \quad (3.32)$$

The last part of the above equation, $\Sigma_{k|k-1} H H^T + R$, is the result in brackets I found in equation 3.21. The density function of a n-dimensional Normal distribution can be written in matrix form as

$$\frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

Where μ is the mean value and Σ is the covariance matrix I found before. In my case $(x-\mu)$ is replaced by $\nu_k = y_k - H_k\hat{x}_{k|k-1}$ and its covariance matrix is $K_{k|k-1}$. Hence, the probability density function can be rewritten as

$$p(y_k|\, y_{k-1};\psi) = \frac{1}{(2\pi)^{\frac{p}{2}}\sqrt{|K_{k|k-1}|}} e^{-\frac{1}{2}\nu_k^T K_{k|k-1}^T \nu_k} \tag{3.33}$$

Taking the logarithm of the above equation (3.33) gives

$$\ln(p(y_k|\, y_{k-1};\psi)) = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln|K_{k|k-1}| - \frac{1}{2}\nu_k^T K_{k|k-1}^{-1}\nu_k \tag{3.34}$$

Which will result in the log-likelihood function

$$L(y,\psi) = -\frac{1}{2}\sum_{k=1}^{N} n\ln(2\pi) + \ln|K_{k|k-1}| + \nu_k^T K_{k|k-1}^{-1}\nu_k \tag{3.35}$$

To estimate the unknown values from the equation 3.35 I use an optimization method aimed at maximising $L(y,\psi)$ with respect to ψ. The optimization will be expressed as follows

$$\hat{\psi}_{ML} = \underset{\psi\in\Psi}{\operatorname{argmax}}\, L(y,\psi) \tag{3.36}$$

This optimization can be either unconstrained if $\psi \in \mathbb{R}^d$ or constrained if the parameter space $\Psi \subset \mathbb{R}^d$.

4 Application

After having analysed how the Bitcoin works and which are the components necessary to its functioning and after having explained the statistical approach I am going to use to analyse Bitcoins' returns, I will apply this latter method to historical data. This chapter will be entirely devoted to explanations, results and comments on the plots I will obtain. So to obtain graphs, I will use a programming software, called MatLab, that will enable me to insert the entire model, apply historical data, and run an estimate to determine whether Bitcoins' returns change can be modelled using a distribution even after disturbances occurred. The model's commands can be found in the appendix. Most of the commands are explained in chapter 3 as they are equal to the ones I used to explain, step by step, the statistical approach I will use. Since the model is applied to historical data, it is better to first provide a graph of the past returns over time and some statistical indicators such as mean, variance, standard deviation, skewness and kurtosis.

The graph below shows the Bitcoins' returns over a time period that goes from the 28^{th} of April 2013 to the 9^{th} of October 2017 (1627 days but 1626 returns as they are computed as the logarithm of the ratio between the $n+1^{th}$ price and the n^{th} price so, the last day's return cannot be computed).

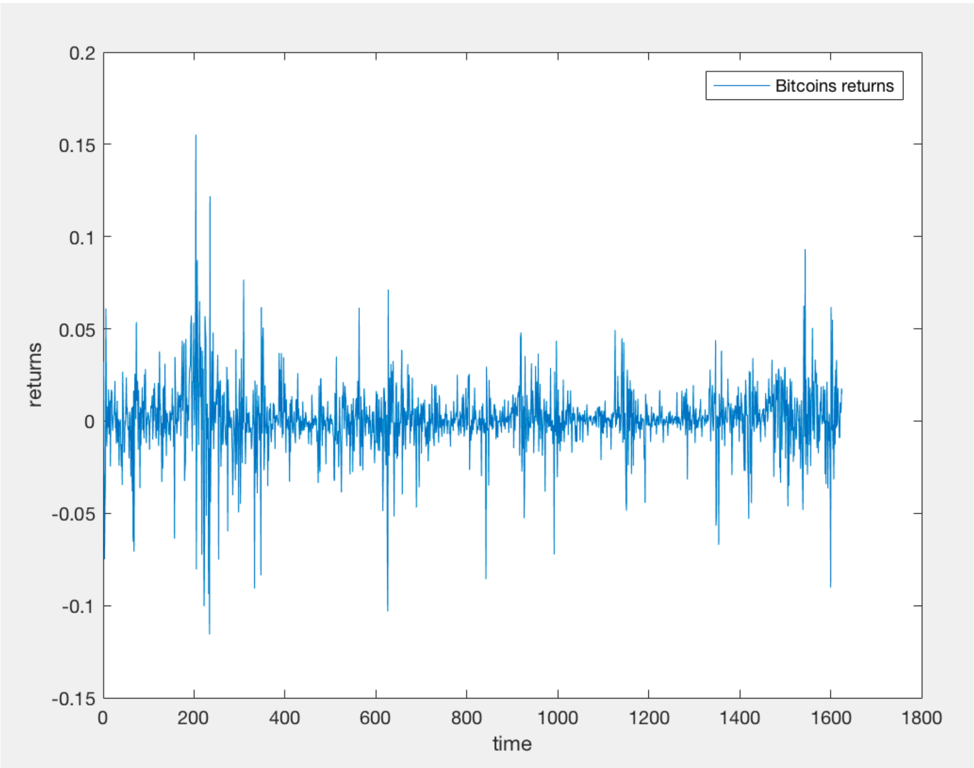

Figure 6. Graph representing Bitcoins' returns

What I can observe from the graph is that Bitcoins' returns are very volatile going from a maximum of 0.1552 to a minimum of -0.1156. Thanks to MatLab I have also been able to compute the mean,

variance, standard deviation, skewness[5] and kurtosis[6] of the historical series. The data set presents a mean and a variance equal to 9.5442e-04 and 3.4723e-04 respectively. Since the variance gives information about the deviation of a variable from its mean value, a variance equal to 3.4723e-04 makes me concluding that every observed return will differ from its mean by 3.4723e-04.
Another important datum is the one about the skewness which, in this case, happens to be -0.2562: so, basically, the data set deviates by -0.2562 from symmetry. In the end, the last indicator I am going to report is the one about the kurtosis. The kurtosis is equal to 12.8204 which explains the amount of variance due to outliers. From a graphical point of view, it means that the tails of the probability density function are quite fat: since the kurtosis is bigger than 0, the distribution is said to be leptokurtic.

Before starting, however, I will make a comparison between two different distributions, the t-Location Scale and the Stable distributions, that are the ones employed the most when data present heavy tails. Moreover, I will also include the Normal distribution so to let the reader clearly understand how the two distributions that better fit the data set work with respect to the Normal one.
The first distribution I am going to deal with is the Normal one, so to give a background. The probability density function (pdf) of the Normal distribution is given by the following formula

$$y = f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \qquad (4.1)$$

The two parameters μ and σ are the mean and the standard deviation. It is a common continuous probability distribution generally used to represent random variables whose distribution is unknown.

The second distribution I will introduce is the t-Location Scale. Along with the Stable distribution, it fits the data set better than the Normal one does. The t-Location is used to model data distributions with fat tails which are present when the data set is prone to outliers. The pdf of the t-Location distribution is given by the formula below

[5] The *skewness* of a distribution provides a mathematical way to describe how much a distribution deviates from symmetry (Introduction to Econometrics, third edition, 2012)
[6] The *kurtosis* of a distribution is a measure of how much mass is in its tails and it is a measure of how much of the variance of a random variable arises from extreme values (Introduction to Econometrics, third edition, 2012)

$$\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sigma\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)}\left[\frac{\nu+\left(\frac{x-\mu}{\sigma}\right)^2}{\nu}\right]^{-\left(\frac{\nu+1}{2}\right)} \tag{4.2}$$

The parameters in this equation are respectively μ that represents the mean, while the variance is represented by the values σ and ν[7]. μ represents the location parameter and it can assume any value between $-\infty$ and $+\infty$; σ is the scale parameter and it only assumes values grater than 0; and finally, ν is the shape parameter that can only be greater than 0 and it affects the general shape of the distribution. However, this distribution approaches the normal one as one of the parameters, ν, that is the one that tells information about the shape, goes to infinity.

The last distribution is the Stable distribution. As the t-Location Scale, it is a class of probability distribution used to model fat tails (kurtosis) and skewness. This specific type of distribution does not provide any pdf. However, it is represented by a characteristic function given by the following equation,

$$E\left(e^{itX}\right)=\begin{cases} exp\left(-\gamma^{\alpha}|t|^{\alpha}\left[1+i\beta sign(t)tan\frac{\pi\alpha}{2}\left((\gamma|t|)^{1-\alpha}-1\right)\right]+i\delta_0 t\right) & for\ \alpha\neq 1 \\ exp\left(-\gamma|t|\left[1+i\beta sign(t)\frac{2}{\pi}\ln(\gamma|t|)\right]+i\delta_0 t\right) & for\ \alpha=1 \end{cases} \tag{4.3}$$

The parameters are respectively α that is a number greater than 0 and *at most* equal to 2 and it is denoted as the first shape parameter; β is the second shape parameter and takes on values *at least* equal to -1 and *at most* equal to +1; γ is the scale parameter and can only be positive so, it assumes values from 0 to $+\infty$ and, finally, δ is the location parameter that can take on any value.
The first shape parameter, α, describes the tails of the pdf while the β, the second shape parameter, illustrates the skewness of the pdf. If $\beta=0$, the distribution is symmetric; if $\beta>0$, the distribution is right-skewed while, if $\beta<0$, the distribution will be left-skewed. The scale parameter, γ, generally stretches or shrinks the distribution and the larger it is, the more the distribution is spread out. The location parameter, δ, has a shifting graph function.
These distributions are depicted in the graph below. The plot makes a comparison between the three distributions using the vector Y_btc that is the vector containing all the returns on Bitcoin. I have

[7] Note that ν in this case it is not the error term of the observation vector but just a parameter of the probability density function of the t-Location Scale.

computed the returns by taking the logarithm of the ratio between the $n + 1^{th}$ Bitcoin s' price and the n^{th} Bitcoin's price measured over a time period. As you can see from the plot, the Normal distribution (in red) does not fit the data set: generally speaking, the Normal distribution is not a good approximation of a financial data set. The Bitcoins' case is even more extreme as Bitcoins are very volatile due to their construction: their supply is fixed by their developers. This peculiarity of the Bitcoin makes its value unpredictable. However, I put it just to have a general idea of how peaked the data set is and how it differs from the Normal approximation. Moreover, the t-Location and the Stable distributions show how the centre of mass is concentrated around zero while, the Normal one shows a wider distribution (with a larger variance). The other two distributions, the t-Location and the Stable, in blue and brown respectively, as depicted from the graph, better approximate the data set composed by the rates of return of the Bitcoin over the sample period.

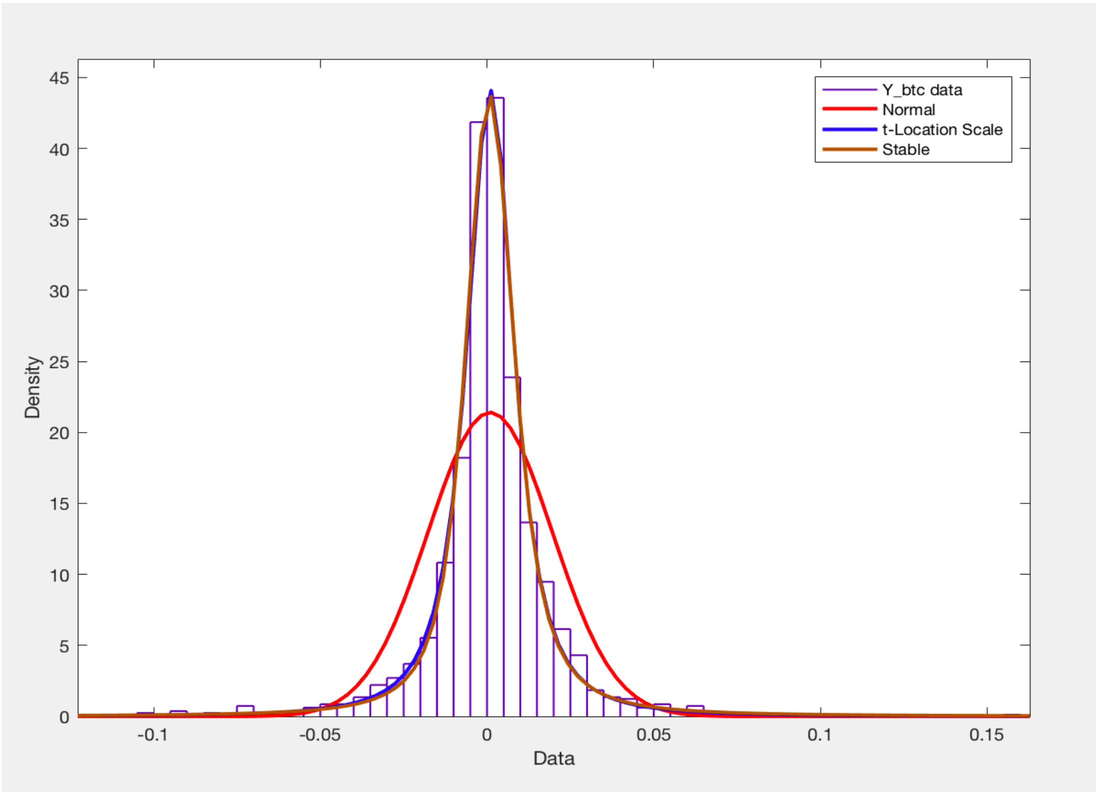

Figure 7. Plot representing the Probability density functions of Bitcoins' returns according to either a Normal, a t-Location or a Stable distribution

Thanks to a MatLab tool, called `dfittool`, apart from drawing the distributions graphs, I have been able to compute the corresponding means, variances that are the estimates of μ and σ with their corresponding standard errors. All the results are reported in the table below.

	Normal	t-Location Scale	Stable
Mean (μ)	0.000954421	0.00104507	0.00208361
Variance (σ^2)	0.00034723	Infinite	Nan
Parameter Estimate μ	0.000954421	0.00104507	
Parameter Estimate σ	0.0186341	0.00788913	
Parameter Estimate ν		1.75467	
Parameter Estimate α			1.26921
Parameter Estimate β			0.0740078
Parameter Estimate γ			0.00674468
Parameter Estimate δ			0.000974437
Standard error of μ	0.000462255	0.000252933	
Standard error of σ	0.000327015	0.000320008	
Standard error of ν		0.114617	
Standard error of α			0.0346448
Standard error of β			0.0564947
Standard error of γ			0.000205114
Standard error of δ			0.000276765

As it can be seen from the plot and, more evidently, from the table containing the most significant data about the different distributions above, the Stable distribution is the one that provides me with much information about the data set. In fact, the first shape parameter α, is approximately equal to

1.27 which describes how fat the tails of the distribution are. Since α can at most be equal to 2, I see that the Stable distribution depicts a discrete fatness of the tails. Other pieces of information can be collected by looking at the second shape parameter, β: since it is approximately 0.07, which is a value quite close to 0, I conclude the distribution is approximately symmetric.
Moreover, if I look at the other two parameters, γ and δ, I can conclude that the scale parameter, γ, is very small which indicates that data will not be spread out but rather gathered around a mean value. For what it concerns the location parameter, δ, since it is very close to 0, I end up saying that the plot will not be shifted with respect to the mean value. In the end, I can assess that the Stable distribution is a good fitter of the Bitcoins' returns data set.

As I said before, the t-Location Scale distribution is used when the data set is prone to outliers, so when it has fat tails. As already depicted by the Stable distribution, this data set is characterized by fat tails. So, the t-Location Scale can be a good approximation of the data set too. The only information I am provided with when I use the following distribution is the information about the shape of the graph, that is given by the parameter ν. In this case ν is roughly equal to 1.755. ν can assume any value greater than 0 up to infinity. As it approaches infinity, the shape of the graph resembles the Normal distribution. Compared to infinity, the value of ν I have in this particular case is really small. Hence, I can conclude that the shape of the graph, as I see, is not approximately bell-shaped but rather peaked.

After having analysed the Probability density functions of Bitcoins' returns according to the three different distributions, I am going to analyse the Cumulative density functions and the Probability functions. As before, I will use the MatLab tool, `dfittool`, that will provide me with two different plots according to the two functions I want to graph. I will examine the Cumulative density function (cdf) first and the Probability plot later. As for the Probability density graph, the red line represents the Normal distribution, the blue line the t-Location distribution and the brown line the Stable distribution.

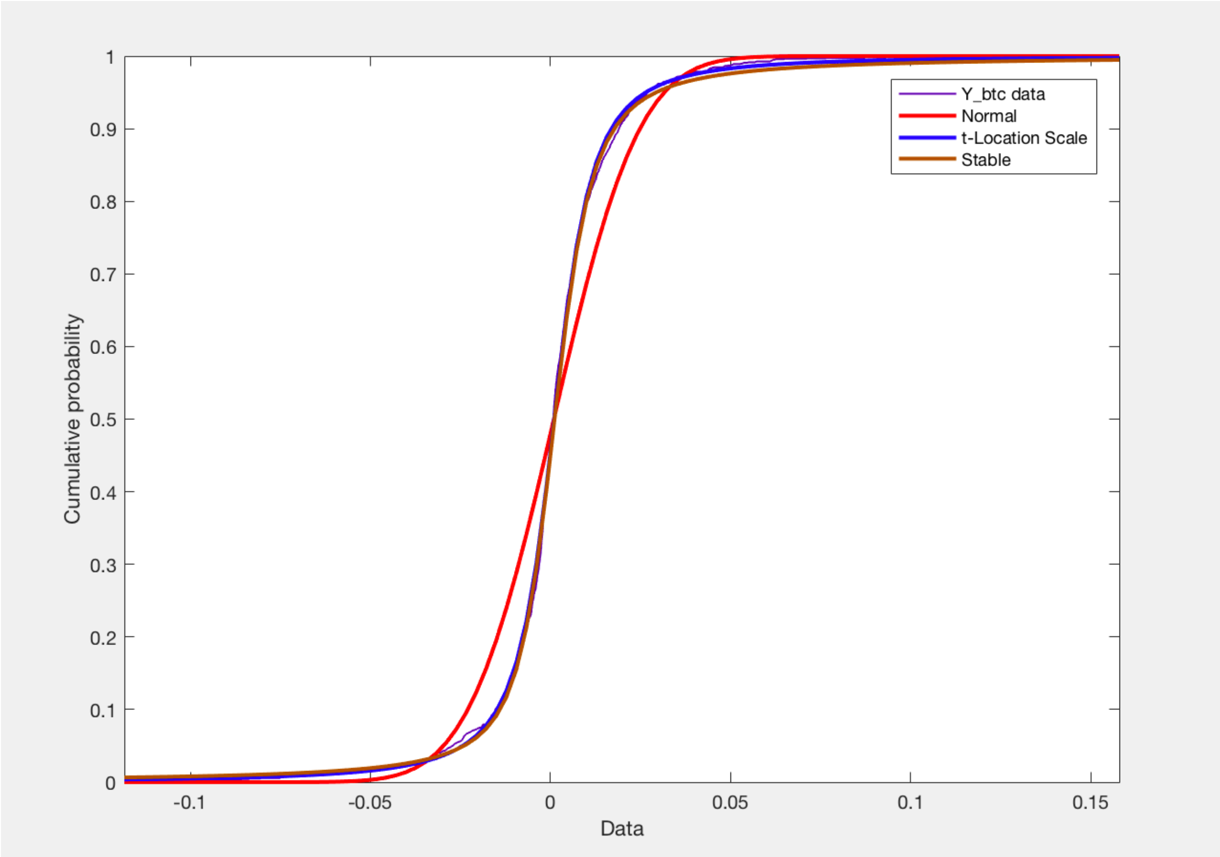

Figure 8. Plot representing the Cumulative density functions of Bitcoins' returns according to either a Normal, a t-Location or a Stable distribution

The Cumulative distribution function of a random variable x is defined to be the probability that X will take a value smaller or at least equal to x. As you see from the chart above, both the t-Location Scale and the Stable distributions almost fit the data set while the Normal distribution is less precise.

For the Normal distribution, the Cumulative distribution function becomes more stretched as σ^2 increases, while it shifts either to the left or to the right according to the sign of μ. Since this data set is more prone to outliers, as I said in the previous paragraph too, the Normal distribution will not be a good approximation of this data set: in fact, the Normal approximation creates a sort of gap between the actual data set graphed by a thin violet line and the red line.

A different observation can be made for both the t-Location and the Stable distributions that almost perfectly graph the distribution of the data set: for the t-Location Scale, the higher the ν the more stretched the Cumulative density function will be and the more it will resemble the Normal distribution. With a parameter ν almost equal to 2, the plot will not be as stretched as when I have a ν equal to infinity but, it will be stretched enough to represent the data set.

Same result can be drawn for the Stable distribution: differently from the t-Location, the Stable distribution has two shape parameters α, the stability parameter, and β, the skewness parameter, that can take values in between 0 and 2 and -1 and +1 respectively. With α being equal to almost 1.27 and β being equal to 0.07 more or less, the shape of the Cumulative function using the Stable distribution will be sufficiently stretched so to fit the data considering outliers too, hence taking into account the asymmetry caused by some random returns.

These results are even more evident if I analyse the Probability plot below:

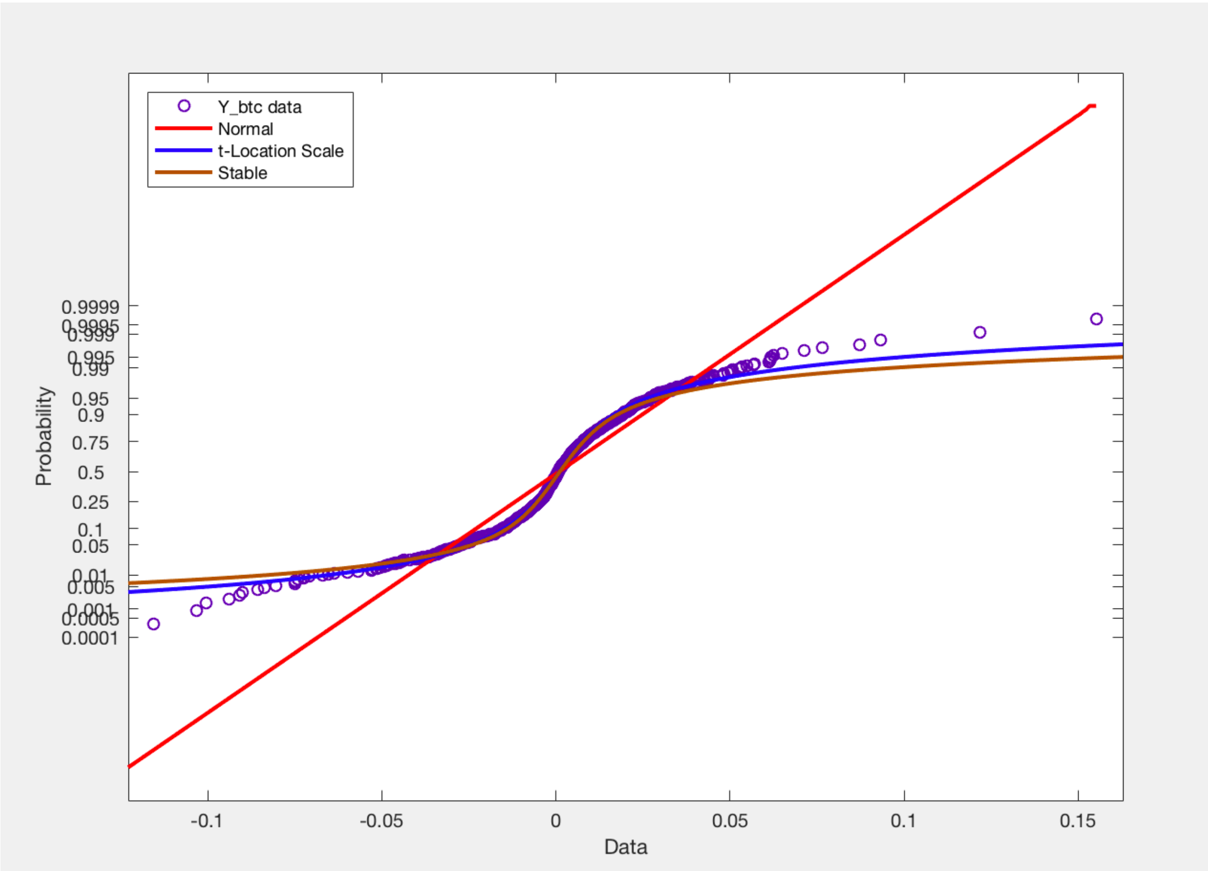

Figure 9. Plot representing the Probability functions of Bitcoins' returns according to either a Normal, a t-Location or a Stable distribution

The Probability plot is generally used to compare two data sets. As usual, the red line represents the Normal, the blue line represents the t-Location Scale and the brown line represents the Stable distribution. What differs in this graph is the line representing Bitcoins' returns: in fact, they are represented by circles rather than a continuous thin line. The ones that do not lie on the red line represent departures from normality. However, as you see, they are not really well-fitted neither by the t-Location nor by the Stable distributions, especially at the extremes. This is because the Bitcoin, as I wrote before, is very volatile, prone to outliers and with fat tails. This plot is the evidence that even if there are some distributions that *mostly* fit the data set, there will always be outliers.

Now, after having analysed the different types of distributions with their respective densities, cumulative and probability functions, I will apply the *Kalman filter* on MatLab so to compute the value of the maximum likelihood function L introduced in equation 3.27. After having computed the value L I will have to maximise it so to observe whether that is a global (absolute) or local maximum. In order to carry out the aforementioned operations I will use two different scripts on MatLab each for every different function, one to compute $-L$ and the other to maximise it. The *Kalman filter* is already contained in the first script. In the Appendix, you will also find the codes so to generate a random trajectory that in this case is not necessary as I already have it: it is the series of past returns that I obtained by observing the Bitcoins' prices over a time period and then I computed the return for every period by applying the logarithm to the ratio of the $n+1^{th}$ price and the n^{th} price. In the Appendix you will also find the *Kalman filter* as if it were a separate script, however, in this case I assumed it to be contained in the *"compute_minusL"* script. The function so to compute $-L$ is given by the following relation:

```
[L,Err,mu,v] = calcolo_menoL(parametri,x0,Y_btc')
```
(4.4)

Where `L` is the value of the maximum likelihood function; `Err` is the estimated error term; `mu` is the first component of the *state-vector equation*, x, and `v` is the estimated error. The function to the right of the equal requires the computation of minus L given the `parametri` that are `a,b,R,Q`, `x0` and `Y_btc'` where `a` and `b` are the two constants introduced in equation 3.7, `R` and `Q` are the two variance-covariance matrices of the error terms w and v of equations 3.1 and 3.2. The two matrices have been introduced in equations 3.6 and 3.8. As previously stated, matrix Q will be a diagonal matrix having variances on its main diagonal while matrix R will be a 1x1 matrix. `x0` will instead be the initial observed values of variable `x`: it was introduced in the first script so to start generating the trajectory. `Y_btc'` is the transposed vector of Bitcoins' returns.
After having found `L`, I will apply another command necessary to maximise the value of the maximum-likelihood function, that is

```
[a,b,R,var_mu,var_v,Err,mu,v,maxL,BTC_stimata]=
            =massimizzoL(x0,Y_btc')
```
(4.5)

The new output variables are `var_mu` and `var_v` that are the variances of μ and ν and they are respectively the terms that lie on the main diagonal of Q. `maxL` is the variable representing the maximised value of `L`, that will be obtained by running the above command, while `BTC_stimata` is `Y_btc`.

By posing all parameters equal to 1, `x0` being equal to a 1×2 matrix [0.1, 0.2] and `Y_btc'` being equal to the Bitcoins' returns vector, `L` will be equal to `1.3996e+03`, `Err`, `mu` and `v` will be equal to three vectors made of as many error terms as the number of returns contained in `Y_btc` (1625) while `v_k`, that is the prediction error, is equal to `-0.0059`. The command in order to compute `v_k` will be

```
v_k= Y_btc(:,t+2)-a*Y_btc(:,t+1)-b*Y_btc(:,t)-(H*x_k_given_k)
```
(4.6)

Equation 4.5 is the estimated error that is equal to the difference between the actual Bitcoins' returns observed on previous periods and the estimation already mentioned in Chapter 3 as the difference between z_k and $H\hat{x}_{k|k-1}$ in equation 3.13.

So to maximise `L` now, I will have to use the command in equation 4.5 that, given the parameters being all equal to 1 and given `x0` being equal to a 1×2 matrix [0.1, 0.2] and `Y_btc'` being equal to the Bitcoins' returns vector, `maxL` will be equal to `7.7033e+03`. This is the maximised value of L over a given interval whose extremes have been established in the script denominated *"maximiseL"*.

In this estimation, the lower bound is given by `lb=[a-20 b-20 0.00001 0.00001 0.000001]` and the upper bound is instead given by `ub=[a+20 b+20 1 1 2]`. According to the following estimation, the graph of the estimated error, `Err`, of `mu` and of `v` is given by the plot below:

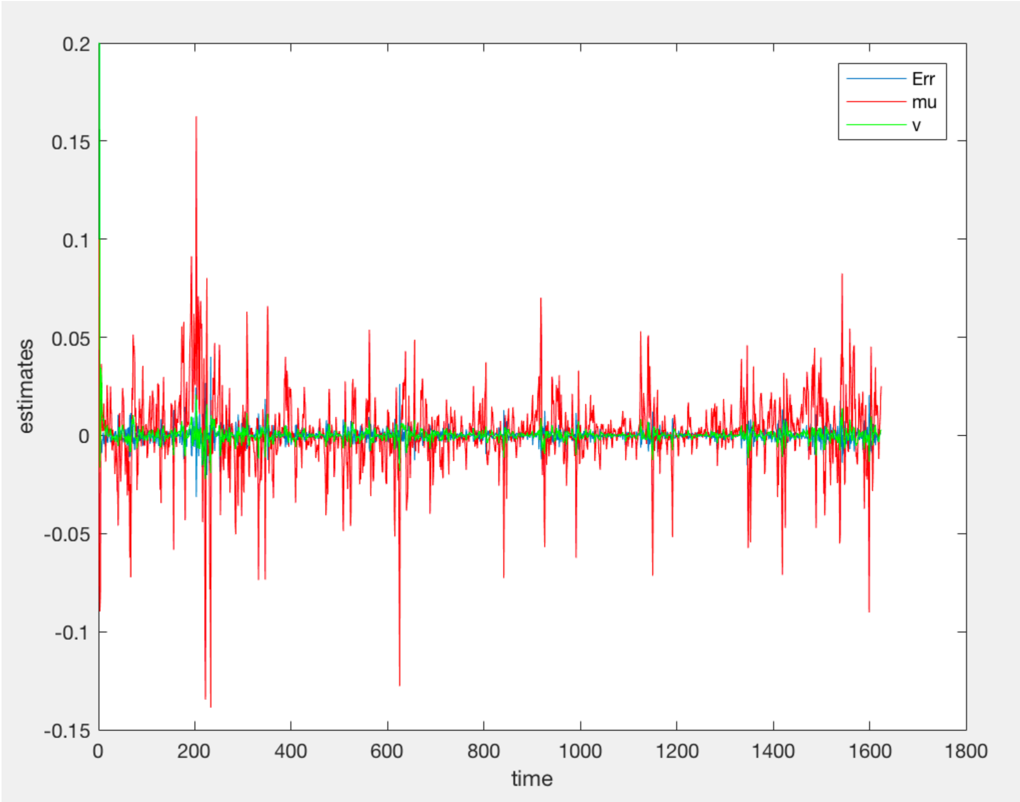

Figure 10. Graph representing the estimated error, the first component of vector x and the measurement Gaussian error when all parameters are equal to 1

I can however carry out the same analysis by changing the numbers assigned to the parameters in equation 4.4.

Suppose I now change the values of the parameters and of `x0`: I set `a` being equal to `0.01`, `b` being equal to `0`, `R` being equal to `0.0846`, `var_mu` being equal to `0.9649` and `var_v` being equal to `0.4325` and `x0` will have to be a row vector whose entries will be randomly established by the command

`x0(i)=normrnd(0, i)` (4.7)

where `i` takes the value of either 1 or 2 depending on the entry of the row vector. With this new values, `L` will be equal to `612.9585`, `Err`, `mu` and `v` will be equal to three vectors made of as many error terms as the number of returns contained in `Y_btc` (1625) and `v_k` is equal to `-0.0059` as before. The value that maximises `L` happens to be the same as before `7.7033e+03` as I am still analysing the same interval. What if I change the interval too? I set the lower bound so that it is equal to `lb=[a-50 b-50 0.00001 0.00001 0.000001]` and the upper bound being equal to `ub=[a+50 b+50 1 1 2]`. `maxL` will now be equal to `7.6805e+03` so it will be lower than before hence I can carry on the analysis by changing the interval upon which I am making the observations so to see whether the maximisation of `L` happens to be absolute or just local.

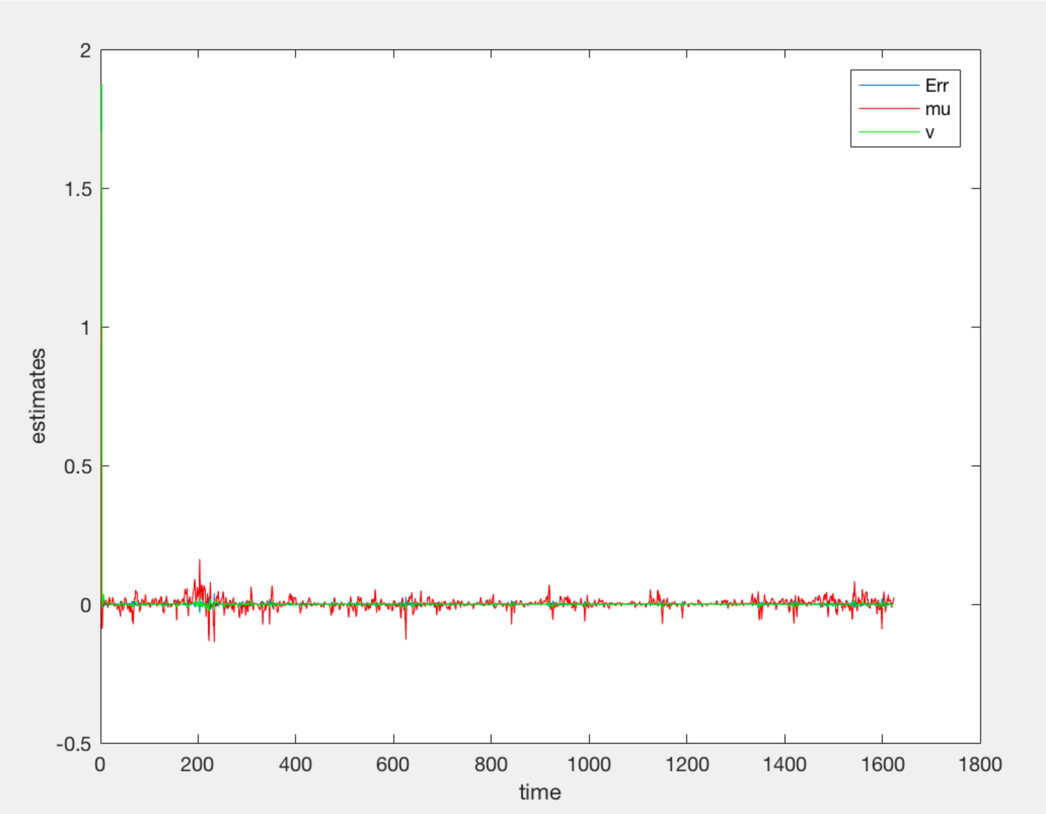

Figure 11. Graph representing the estimated error, the first component of vector x and the measurement Gaussian error when parameters are all different and the interval widens to ± 50

Suppose I change the parameters setting them equal to `[a b R var_mu var_v]=[0.00351,0.034,0.059,0.83,0.7091]`, the value of `x0` to be set randomly by the software (equation 4.7) and the interval, shrinking it to ± 5. The graph now becomes very similar to the one I obtained when the interval was ± 50 with a `maxL` still equal to `7.7033e+03`. Hence, what I can conclude is that the value of `maxL` equal to `7.7033e+03` results in being the global maximum: this will maximise the probability of having obtained the given sample.

4.1 Error Metrics

This one to last paragraph will be devoted to error metrics. Error metrics are used to measure accuracy when we deal with continuous variables. The three metrics I will briefly discuss are: the Average Prediction Error (***APE***), the Average Relative Prediction Error (***ARPE***) and the Root Mean Square

Error (***RMSE***). They all measure how much the observed values diverge from the estimated ones. The first one I am going to analyse is the APE whose formula is given by

$$\boldsymbol{APE} = \sum_{j=1}^{N} \frac{|y_j - \hat{y}_j|}{y_j} \tag{4.8}$$

Where the y_j is the observed value while $\hat{y}_j$ is the estimated one. The numerator is the result of the difference between the actual observed value and the estimated one, so it is the error caused by the estimation while the denominator is given by the sum of all observed values. As the name suggests, it is an average of the predicted error caused by the estimation.

The other error metric I will talk about is the ARPE which is explained by the following equation

$$\boldsymbol{ARPE} = \frac{1}{N} \sum_{j=1}^{N} \frac{|y_j - \hat{y}_j|}{y_j} \tag{4.9}$$

This formula is similar to the one I used for the ***APE*** but for the fact that, at the denominator, I am no longer summing all the observed values so it will be equal to just one observed value for the period I am looking at multiplied by the number of values observed.

The last metric I am dealing with is the ***RMSE*** which is equal to

$$\boldsymbol{RMSE} = \sqrt{\sum_{j=1}^{N} \frac{|y_j - \hat{y}_j|^2}{N}} \tag{4.10}$$

This equation is the square root of the average of squared differences between prediction and actual observation. Differently from other equations, since the errors are squared before they averaged, the RMSE gives a higher weight to large errors than small ones. In fact, the RMSE should be employed when large errors are not expected to occur.

Now, I apply the following equations to my data set composed by Bitcoins' returns so to compute these error metrics. The results are reported in the table below

Error Metrics	**Measurements**
Average Prediction Error (APE)	0.141905

Average Relative Prediction Error (ARPE)	0.21734
Root Mean Square Error (RMSE)	0.05182

As these numbers show, since they are all very small, the Kalman filter is a good model to predict Bitcoins' returns as it is a good fit leading to small and inconsistent errors.

4.2 Conclusion

The goal of this thesis was to prove that even if there might be some perturbations that I cannot observe in Bitcoin's returns, it is still possible to compute an estimate of it by observing the state before and after the disturbance occurred. The filter enables me to observe the perturbed system after the disturbance occurred but it does not enable me to observe *exactly when* the perturbation occurred. The situation can be portrayed as such: suppose I am in a condition in which the system can be observed and some measurements can be taken. As some inaccuracies might arise, that cannot be observed (as if the whole system were in a box from which I can only observe the outcome), the filter still enables me to make a good prediction of those unknown variables that result from the estimation. It makes use of the joint probability distribution for every time period. As shown with the filter, since it takes into account the randomness of errors, it has been possible to make an estimate of Bitcoins' returns by designing a proper observation vector which has been used along with the measurement prediction to compute the estimated error. At the end the error term has been minimized in order to have a prediction that is very close to the actual measurement.

5 Appendix

In this section I will report all the commands implemented on MatLab.

To generate the trajectory:

```
function [X_tot,Y_tot] = genero_traiettoria(a,b,R,Q) %where both a and b are two parameters while R and Q are defined as diag(rand(1,1))*0.001 and diag(rand(2,1))*0.001 respectively


T=50 %time
dt=1 %sample time
aa=ones(2,1) %implementation of matrix PHI
PHI=diag(aa)
```

```
PHI(1,2)=1

X=0 %to insert the solution vector and the output variable x_{k+1}

X_tot=[] %to collect them all
Y_tot=[]

H=[1 0] %to implement matrix H

Q=diag(rand(2,1))*1 %to determine the error variance-covariance matrix Q
R=diag(rand(1,1))*1 %to determine the variance of the error term v_k

x0=[0.1; 0.2] %initial data to generate the trajectory
y0=[0.1, 0.3] %in order to compute y_2

X=[x0]
Y=[y0]

times=[ 0 : dt : T] %time interval that goes from 0 to T with a pace equal
to dt

%application of the for-loop so to estimate the errors of the state-space
and of the observation equation, of the state-space and of the observation
vectors too. While X_tot contains the solution for every single time period,
Y_tot contains the ending value for every time period
for t =times(2):max(size(times))
w=(mvnrnd(zeros(2,1),Q,1))'
v=(mvnrnd(zeros(1,1),R,1))'
Y_tot = [Y_tot, Y]
Y=a*Y_tot(end)+b*Y_tot(end-1)+H*X+v
X=PHI*X+w
X_tot=[X_tot,X]
end
```

To implement the filter:

```
function[MEAN_PREDICTION,ERROR_PREDICTION]=filtro_new_3(a,b,R,var_mu,var_v,
Y_btc)

Sigma0=eye(2)*100  %a priori estimate of the covariance of x – in chapter 3,
I called it Σ_{k|k-1}
```

```
X0=ones(2,1) %a priori estimate of the mean of x

x_k_given_k_minus_one=X0 %initial value of x_{k|k-1}

Sigma_k_given_k_minus_one=Sigma0 %initial value of Σ_{k|k-1}

MEAN_FILTERING=[] %to store all values of x_{k|k}

COV_FILTERING=[] %to store the norm of Σ_{k|k}

MEAN_PREDICTION=[X0] %to store the values of x_{k+1|k}

COV_PREDICTION=[norm(Sigma_k_given_k_minus_one)] %to store the norm of Σ_{k+1|k}

ERROR_PREDICTION=[] %to collect all estimates of the prediction error

COV_ERROR_PREDICTION=[] %to collect all estimates of the covariance of the prediction error

L=0

%application of the for-loop so to estimate the measurement update and the time update equations. The measurement update equation will have to be compared to the initial X0, the a priori estimate of the mean of x while the time update equations will constitute the prediction
for t=1:max(size(times)-2) %to scan all time periods
x_k_given_k=X0+Sigma0*H'*inv(H*Sigma0*H'+R)*((Y_tot(:,t+2)-a*Y_tot(:,t+1)-b*Y_tot(:,t))-H*X0)
Sigma_k_given_k=Sigma0-Sigma0*H'*inv(H*Sigma0*H'+R)*H*Sigma0
MEAN_FILTERING=[MEAN_FILTERING,x_k_given_k] %to store all values of x_{k|k}
COV_FILTERING=[COV_FILTERING,norm(Sigma_k_given_k)] %to store the norm of Σ_{k|k}
X0=PHI*x_k_given_k
Sigma_k_given_k_minus_one=PHI*Sigma_k_given_k*PHI'+Q
MEAN_PREDICTION=[MEAN_PREDICTION,X0] %to store x_{k+1|k}
COV_PREDICTION=[COV_PREDICTION,norm(Sigma0)] %to store the norm of Σ_{k+1|k}
end

L=-0.5*L %correction of the log-likelihood function. In chapter 3 I wrote it as a unique function. On MatLab it estimates the log-likelihood function first and it corrects it for -0.5 after
```

```
%all the commands below will be necessary to draw the plot
figure()
subplot(2,2,1)
plot(times,Y_tot(1,:),'b-')
hold on
plot(times(1:length(MEAN_PREDICTION(1,:))),MEAN_PREDICTION(1,:),'k-')
xlabel('time')
ylabel('z')
legend(' \z storica',' \z  filtro')

subplot(2,2,2)
plot(times(1:length(MEAN_PREDICTION(1,:))),abs(ETH(1,3:end)'-
MEAN_PREDICTION(1,:)),'m-')
xlabel('time')
ylabel('|z storica - z filtro|')
```

To compute minus L:

```
[L,Err,mu,v] = calcolo_menoL(parametri,x0,Y_btc') %parameters are
[a,b,R,Q,x0]

v_k= Y_tot(:,t+2)-a*Y_tot(:,t+1)-b*Y_tot(:,t)-(H* x_k_given_k)

K_k_given_k_minus_one=H*Sigma0*H'+R

ERROR_PREDICTION=[ERROR_PREDICTION,v_k] %to compute the error prediction
COV_ERROR_PREDICTION=[COV_ERROR_PREDICTION,K_k_given_k_minus_one]       %to
compute the covariance matrix of the error prediction

L=L+log(det(K_k_given_k_minus_one))+v_k'*(K_k_given_k_minus_one)^(-1)*v_k
%log-likelihood function

Err(t)=v_k
mu(t+1)=x_k_given_k(1)
v(t+1)=x_k_given_k(2)
```

To maximise L:

```
[a,b,R,var_mu,var_v,Err,mu,v,maxL]=massimizzoL(x0,Y_btc)


a11=0
a12=0
```

```
a22=0
a21=0
b11=0
b21=0

c=length(BTC')

for k= 2:c-1
a11=a11+(BTC(k)'.^2)
a12=a12+(BTC(k)'*BTC(k-1)')
a22=a22+(BTC(k-1)'.^2)
b11=b11+(BTC(k+1)'*BTC(k)')
b21=b21+(BTC(k+1)'*BTC(k-1)')
end

a21=a12
A=[a11,a12;a21,a22]
B=[b11;b21]
D=inv(A)*B
a1 = D(1)
b1 = D(2)

a=a1
b=b1

%the code below has been implemented in order to find the upper boundary of
the interval, the optimum of every parameter. The last three terms
represent the variances that can be changed with the only constraint of
being positive
lb=[a-20 b-20 0.00001 0.00001 0.000001]

%so to find the upper boundary of the interval, the optimum of every
parameter. Same thing holds for the last three terms
ub=[a+20 b+20 1 1 2]

%initial data where to start from in order to find the optimum of every
parameter. Always choose a value in between the upper and lower boundary of
every parameter
c0=[a b 0.5 0.5 0.5]

[parametri_opt,FVAL]=fmincon(f,c0,[],[],[],[],lb,ub)

maxL=-FVAL
a=parametri_opt(1)
b=parametri_opt(2)
```

```
R=parametri_opt(3)
var_mu=parametri_opt(4)
var_v=parametri_opt(5)

[L,Err,mu,v] = calcolo_menoL(parametri_opt,x0,BTC)
```

To compute the bitcoin's return:

```
for i=2 : length(F)
BTC(i-1)=log(F(i)/F(i-1))
end
```

6 Bibliography

Joshua Baron, Angela O'Mahony, David Manheim and Cynthia Dion Schwarz, *National Security Implications of Virtual Currency. Examining the Potential for Non-state Actor Deployment*, Chapter Title: *The Current State of Virtual Currencies*, Published by: *RAND Corporation (2015)*

Paolo Paesani, *Types of Money, Money and Banking Lecture 3 (2017)*

Enrico Marro, Article *The five weirdest currencies of the history, Il sole 24 ore – finanza e mercati (12/01/2016)*

Mauro Bellini, Article *Blockchain: what it is, how it works and where it is applied in Italy, Blockchain 4 Innovation (14/03/2017)*

Video *What is Blockchain?, World Economic Forum (21/01/2016)*

Carlo Moras, Article *Five Banking trends: among AI, Blockchain and FinTech, TSW (27/01/2017)*

Alessandra Caparello, Article *FinTech: what it is and its definition, WSI – Wall Street Italia (20/04/2017)*

Mauro Bellini, Article *Big Data, Blockchain, AI and FinTech change the world of banks and finance, BigData 4 Innovation (03/01/2018)*

Alessio Sarnelli, Article *FinTech and Blockchain: all secrets of 4.0 Economy, Copernico - where things happen (04/07/2017)*

Article *FinTech, Allianz releases a Blockchain prototype for the firm insurance market, EconomyUp (14/11/2017)*

Rainer Böhme, Nicolas Christin, Benjamin Edelman, and Tyler Moore, *Bitcoin: Economics, Technology and Governance, The Journal of Economic Perspectives,* Vol. *29,* No. *2,* pp. *213-238,* Published by*: American Economic Association (2015)*

Charles-Antoine Flament, *Blockchain technology: A general purpose technology for the decentralization of governance?, Solvay Brussels School Economics & Management (2015/2016)*

Tony Lacey, *Tutorial: The Kalman Filter, Chapter 11, Massachusetts Institute of Technology (MIT)*

Greg Welch, Gary Bishop, *An Introduction to the Kalman Filter, University of North Carolina at Chapel Hill, Department of Computer Science (2001)*

Florian Herzog, *Kalman Filter and Parameter Identification, ETHzürich (2013)*

Obryan Poyser, *Exploring the determinants of Bitcoin's price: an application of Bayesian Structural Time Series, Cornell University (2017)*

Peter S. Maybeck, *Stochastic models, estimation and control, Volume 1, Department of Electrical Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio (1979)*

MathWorks Documentation

James H. Stock, Mark W. Watson, *Introduction to Econometrics, Third Edition,* Chapter 2: *Expected Values, Mean and Variance,* pp. *65-67,* Published by: *Pearson (2012)*

Figure 1 taken from Joshua Baron, Angela O'Mahony, David Manheim and Cynthia Dion Schwarz, *National Security Implications of Virtual Currency. Examining the Potential for Non-state Actor Deployment*, Chapter Title: *The Current State of Virtual Currencies*, Published by: *RAND Corporation (2015)*

Figure 2 and figure 3 taken from Mauro Bellini, Article *Blockchain: what it is, how it works and where it is applied in Italy, Blockchain 4 Innovation (14/03/2017)*