ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Víctor Revuelta

Design and implementation of a software system for the composition of a database and automated trading system on different cryptocurrency trading markets

Master's Thesis

Chair of Entrepreneurial Risks
Department of Management, Technology and Economics
Swiss Federal Institute of Technology (ETH) Zurich

Examiner:
Prof. Dr. Didier Sornette

Supervisor:
Dr. Dorsa Sanadgol

Zürich, March 18, 2018

Abstract

Cryptocurrency is a means of digital exchange that, by its design and operation, complies with the functions of traditional money allowing the exchange of goods and services by using cryptography to verify and store transactions in a public ledger.

The purpose of this project is to design and implement a cryptocurrency trading platform able to connect to multiple exchange APIs in order to gather market data information as well as place orders automatically on different market pairs. The platform is designed to allow the integration of new cryptocurrency exchange APIs. Furthermore, the software also allows the configuration of different trading strategies based on real time market data information. The present project will constitute the basis of future trading research analysis on the cryptocurrency market.

In addition, an analysis, description and comparison of the most important cryptocurrencies and exchanges is presented. More specifically, the working principles and the key technological differences between most relevant cryptocurrenies based on market capitalization are discussed. A general overview of characteristics, digital coins offered, current performance and fee analysis of different cryptocurrency exchanges is conducted.

Finally, an evaluation of trading strategies based on real data gathered by the designed platform is performed.

Contents

Introduction

1.1 Motivation

Over the last years, cryptocurrencies such as Bitcoin, have become more and more popular due to their disruptive innovation and, most importantly, because of their meteoric rising value. Nowadays, the exchange of virtual currencies is present in a very active way and it is one of the most revolutionary transaction systems of all times. Unlike traditional currencies, such as conventional fiat money, cryptocurrencies are electronic assets that are independent of banks and states.

The number of cryptocurrencies and exchanges that offer them has been growing enormously in the last years. As of march 2018, there are more than 4500 cryptocurrencies available over the internet and more than 120 active exchanges[1].

Because the cryptocurrency market is very new, many investment opportunities have emerged by means of applying different trading strategies. Although the market is growing very fast, the options to trade automatically are limited. High-speed algorithmic trading in cryptocurrency markets can only be performed via an exchange Application Programming Interface. For this reason, good understanding in programming is a critical requirement. Nowadays, very few open source projects are up-to-date and fully developed.

The main focus of this project is to create a software system that can send automated orders to an exchange, based on a defined strategy using the historic data of one or more cryptocurrencies.

---

[1] Data from Cryptocoincharts. https://cryptocoincharts.info/markets/info.

1.2 Research objectives

The purpose of this thesis is to design and develop a program able to connect to one or more exchanges accessing their API in order to allow an easy placement of orders. In addition, the program should be able to collect and save the most relevant market data in a database. This database should be designed to handle the large size of data that will accumulate.

Other specific objectives of this project are:

- Real-time data acquisition.
- Parallel connectivity with two or more exchanges to allow interexchange strategies.
- Real-time trade execution.

Among that, an analysis and comparison between cryptocurrencies and exchanges will be conducted.

1.3 Thesis outline

The project software result is divided in two different parts:

- Design and implementation of a system able to collect on real time the most important financial data from different cryptocurrency exchanges in order to locally store the information. This will be implemented through a Time Series Database specifically designed to handle large size of data. This data will be used for research, fundamental analysis and strategy backtesting.
- Design and implementation of an automated transaction system able to send orders to a cryptocurrency exchange based on a trading strategy using the above mentioned database.

Both systems will be connected to a cryptocurrency exchange API (Application Programming Interface) in order to achieve a high level of automation, allowing the user to interact minimally with the platform.

The software system will be implemented incrementally. Therefore, the preliminary system should collect data from one exchange and send orders to the same exchange.
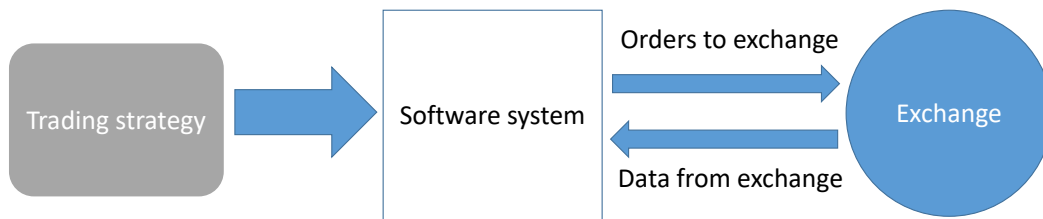
Figure 1.1: Design of the software system

Once the system is correctly designed with a robust performance, it will be implemented with an additional cryptocurrency exchange API both for data gathering and for order sending. This implies more development time and code complexity but will mitigate the risk by not relying on a single exchange for trades and will give more data and the possibility to backtest interexchange strategies.

The program will be designed to enable new exchange APIs implementations. An analysis between exchange characteristics will be made in order to decide the most suitable exchange for every case. Some of the specifications that are going to be taken into consideration are: liquidity, transaction costs, reliability, connectivity and number of digital coins offered.

The platform can be implemented with different degree of automation. It could either be an assisted trading system, which means that the trader manually approves the order before its send to the exchange, or it could be fully automated, which means that the system will send unsupervised orders to the exchange.

Finally, to complete the system, it will be implemented with the possibility to trade with different trading strategies or using different indicators. In the last part of this report, an analysis of possible trading strategies using real time data from 2 different exchanges will be made.

1.4 State of the art

A brief description of the current up-to-date cryptocurrency trading bots projects are as follows.

Haasbot

Haasbot[2] is an advanced crypto trading bot designed primarily for crypto trading professionals, but it is also suitable for beginning traders and hobbyist.

---

[2]Haasbot Bitcoin Bot. https://www.haasonline.com/.

3

Haasbot offers three types of standard bots: Trade, Arbitrage and Maximum Order. Trade bots are supported by a range of technical analysis indicators, safeties and insurances. The 2.0 update also brings customizable Script bots. Depending on the license, users can set up several interconnected bots capable of performing a range of different functions. The new version of Haasbot currently supports Bitfinex, Bitstamp, Binance, Bittrex, Kraken, Poloniex, OkCoin and a few smaller exchanges. The connectivity between Haasbot and the exchanges is done via Representational State Transfer (REST) API. Figure 1.2 illustrates the graphical user interface of Haasbot.

The software is available in three different pricing tiers. The full license is priced at 0,44 BTC per year, around 4.400€ at actual BTC price.

Figure 1.2: Haasbot Interface

Gunbot

GunBot[3] is an automated cryptocurrency trading platform that operates on the following exchanges: Poloniex, Bittrex, Binance, Bittfinex and Kraken. This automatic crypto trading bot includes some strategies, such as: Bollinger Bands, Gain, Step Gain, PingPong, 1000Trades and Supergun. The platform also has the option of mixing these strategies simultaneously in different exchanges. It can run on Linux, Windows or MacOS. The connectivity between Gunbot and the exchanges is done via REST API. Figure 1.3 illustrates the graphical user interface of Gunbot.

The software has three different pricing plans depending on the possibility to run the program in more than one exchange. Prices range from 0,1 to 0,8 BTC, between 1.000€ and 8.000€ at actual BTC price, depending on the license.

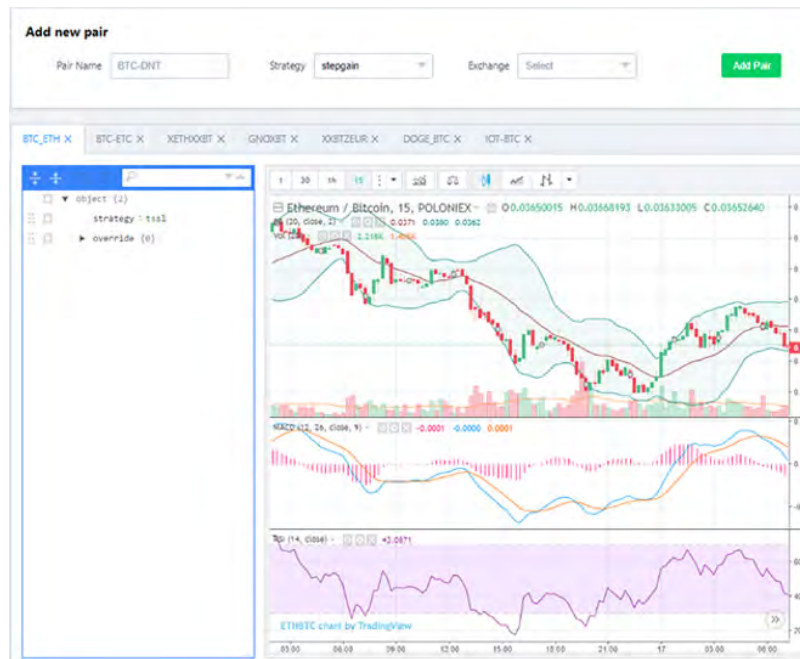[3]Gunbot. Automated Bitcoin and Crypto-Coins Trading Bot. https://gunbot.trading/.

Figure 1.3: Gunbot Interface

Gekko

Gekko[4] is a Bitcoin Technical Analysis trading and backtesting platform that connects to popular Bitcoin exchanges. The project is free and fully open source and it comes with some pre-installed strategies such as: DEMA, MACD, PPO, RSI, StochRSI and CCI. Although it comes with this strategies, the author recommends programming own trading strategies and because of that, good programming skills are required. There is also a Technical analysis library installed, that allows to calculate and use indicators in strategies.

Gekko supports 23 different exchanges (including Bitfinex, Bitstamp and Poloniex). The software offers the possibility to run against the live market (using either a paper trader or real trader) or to backtest a strategy over historical market data. Gekko implements a webinterface that shows the local market data stored. In addition, the program can run in backtesting mode and is able to create automatic graphs in order to visualize the results. The software connection with the exchanges is done via REST API. Figure 1.4 illustrates the graphical user interface of Gekko.

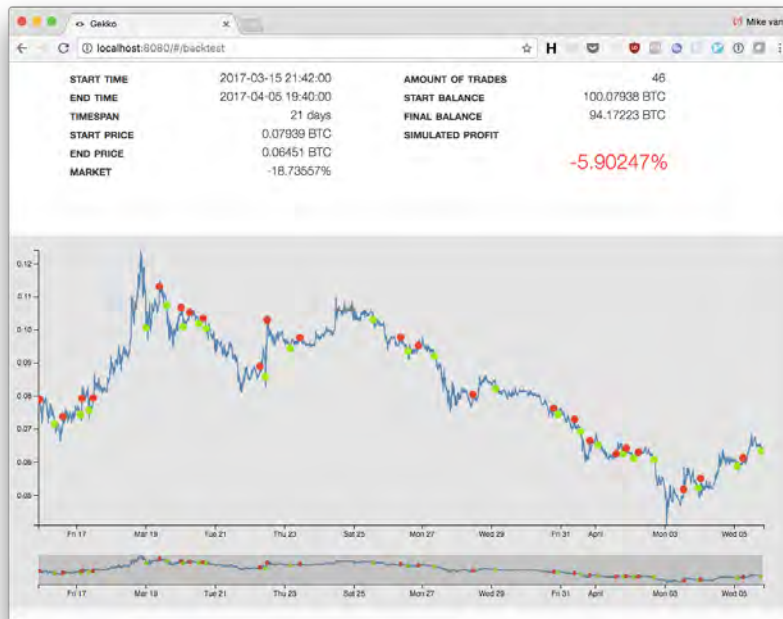[4]Gekko. Open source bitcoin trading bot platform. https://gekko.wizb.it/.

Figure 1.4: Gekko Interface

Tribeca

Tribeca[5] is an automated cryptocurrency market making trading bot. Tribeca supports 4 different exchanges: Coinbase, HitBTC, OKCoin and Bitfinex. The program is featured with a web client user interface and it has been designed to work as a high frequency trading bot. Tribeca is mostly written in Typescript and HTML for the user interface.

The software has different market making trading modes implemented that calculate the best position to place bid and ask orders depending on the order book. The exchange connectivity is mostly done through WebSocket API, but REST API is also used to persist market data information through a MongoDB database.

Although the software is very complete, there is a big lack of documentation and design information. In addition, the project has not been properly updated in the last year while cryptocurrency exchanges have upgraded their APIs causing important divergences between APIs and Tribeca connectivity.

---

[5]Tribeca. A high frequency, market making cryptocurrency trading platform in node.js. https://github.com/michaelgrosner/tribeca.

Cryptocurrencies and Exchanges

2.1 Introduction to cryptocurrencies

Cryptocurrency is a digital or virtual asset used as a way of exchange that implements cryptography to secure transactions and control the production of additional units. By using cryptography, cryptocurrency transactions are verified and stored in a public ledger known as the blockchain[1].

Cryptocurrency is a means of digital exchange that, by its design and operation, complies with the functions of traditional money allowing the exchange of goods and services. It is stored in electronic wallets, which allows carrying out the buying and selling operations needed by the user.

A cryptocurrency differs distinctly from other currencies in that it is has no central regulating body i.e. the European Central Bank for the Euro. The exchange rate of a cryptocurrency is given by the perception of value that the users have of the cryptocurrency, since there is no entity that determine its price. As such its value is much less stable than a standard currency and fluctuates in a manner more similar to other commodities such as precious metals or oil[2].

The first cryptocurrency that started operating was bitcoin in 2009 and since then, many others have appeared, with different characteristics and protocols such as Ether, Litecoin, Ripple or Dogecoin.

These currencies don't have an intrinsic value and they are not backed by an amount of something or by some service, so they are similar to the current fiduciary money. Its value depends on the law of supply and demand, which gives the advantage of not being affected by monetary policies of central banks. Like a foreign currency, bitcoins can be bought or sold in exchange houses or companies that accept it as a form of payment.

2.2 Basic Concepts

2.2.1 Blockchain

A blockchain, also known as distributed ledger, is a distributed database that registers blocks of information and interlaces them to facilitate the recovery of information and the verification that it has not been changed. The blocks of information are linked by means of hash pointers that connect the current block with the previous block and so on until the block genesis is reached [3].

The blockchain is stored by all those nodes in the network that stay in synchrony with it. The generation of new blocks is called mining, as an analogy with gold mining.
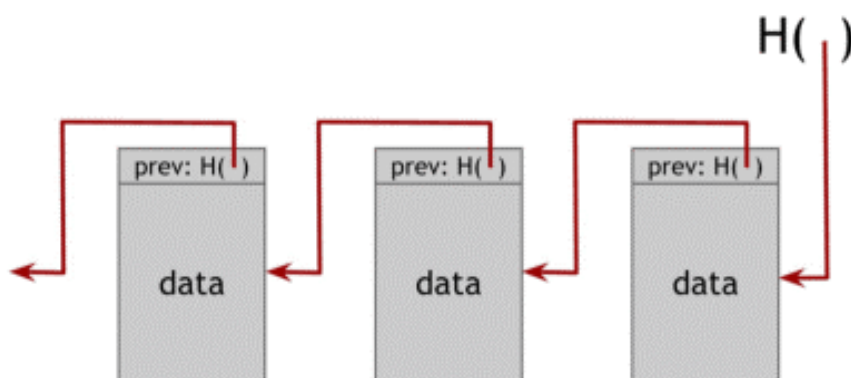

Figure 2.1: Structure of a blockchain. Reprinted from Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, by Narayanan, A. (2016). Princeton.

Each block belonging to the blockchain contains information regarding the transactions related to a period of time, the cryptographic address (hash pointer) of the previous block and a unique arbitrary number (nonce).

In the case of Bitcoin, a new block appears, on average, every 10 minutes[1], and includes information of new transactions that are stored chronologically in the blockchain. One transaction is normally considered valid when it gets up to 6 confirmations (1 hour), although this time is variable and may depend on the seller and the monetary amount[4].

The Bitcoin network is programmed to create six blocks per hour[1]. Each 2016 blocks (around 14 days), all Bitcoin customers compare the real number created with this objective and modify the target by the percentage that has varied. This increases (or decreases) the difficulty of generating blocks.

---

[1]Bitcoin average adding rate. https://en.bitcoin.it/wiki/Block.

2.2.2 Cryptographic hash function and mining difficulty

Cryptographic hash functions are hash functions that are used in many cryptographic algorithms and protocols. It is a mathematical algorithm that maps data of an arbitrary size to a bit string of a fixed size (a hash) and is designed to be a one-way function, that is, a function that is infeasible to invert. There are many applications in the area of information security, some of the most common algorithms in this category include algorithms such as the SHA-256, a successor of SHA-1. There are also other algorithms such as RIPEMD, BLAKE or Skein[5].

Historically, the applications of this type of hash functions were in the context of digital signatures, which are used today in many different applications as a fundamental pillar of many e-commerce protocols. The cryptographic hash functions are also used to generate protocol authentication messages with the generation of random numbers and security passwords. They are characterized by reducing the original message to a sequence of bits that identifies it and is called the "Fingerprint" of the message.

In the case of Bitcoin, the miners in the network compete to be the first to find the solution (the hash) to the cryptographic problem of their current candidate block through a system of Proof of Work.

A Proof of Work is a piece of data which is difficult (costly, time-consuming) to produce but easy for others to verify and fulfills certain requirements. Proof of Work is a random process with low probability which needs a lot of trial and error on average before a valid Proof of Work is generated. Bitcoin uses the Hashcash Proof of Work system[6].

Miners have to solve a problem, the hash, that requires several repetitive attempts, by brute force, that is non-deterministic, preventing miners with high level of processing from leaving out the smallest. The frequency of localization of each block follows a Poisson distribution[6] and the probability that a miner finds it depends on his computational power in relation to the computational power of all the combined nodes. Validation of the solution provided by the miner is trivial and takes place immediately.

The hash rate is the speed at which a computer is completing a hash operation. It measures the number of times a hash function can be computed per second. A miner's expected profit is directly proportional to the hash rate. A higher hash rate increases the opportunity of finding the next block and receiving the reward, in other words, new bitcoins distributed by the network.
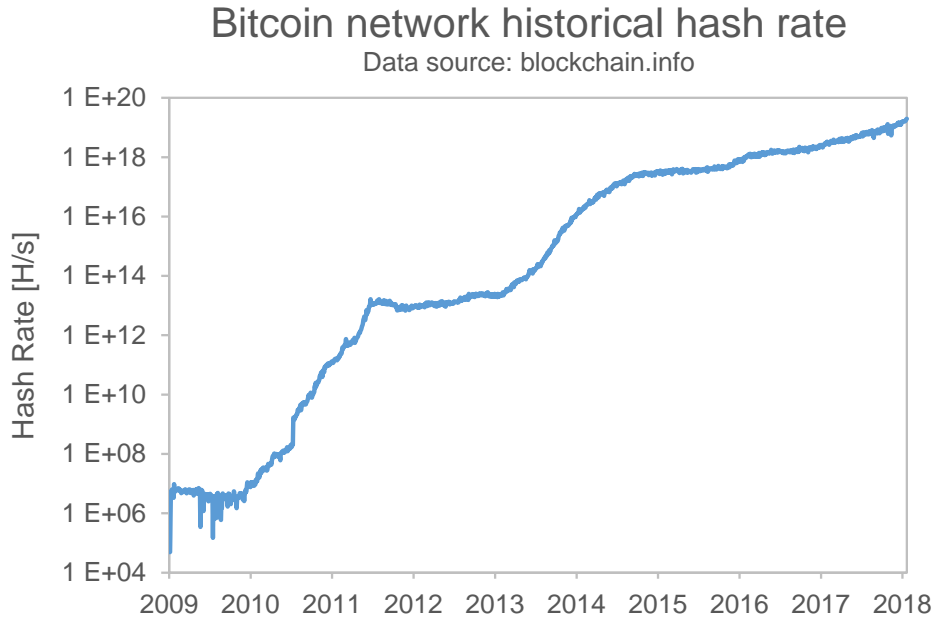
Figure 2.2: Bitcoin network hash rate. Represents the evolution of Bitcoin network hash rate over the last years in a semi-logarithmic scale.

The mining difficulty is a measure of how difficult it is to find a hash below a given target. Bitcoin difficulty started at 1[7]. As mentioned before, the Bitcoin community defined that the creation of new blocks should have a rate of 6 blocks per hour. At this rate, the time needed to create 2016 blocks should be 14 days. We can calculate how much should the difficulty be increased or decreased as follows:

$$D_{n+1} = D_n \frac{14 \text{ days}}{T} \tag{2.1}$$

Where:

D_{n+1}: the network difficulty of the hash after the difficulty update

D_n: the network difficulty of the hash before the difficulty update

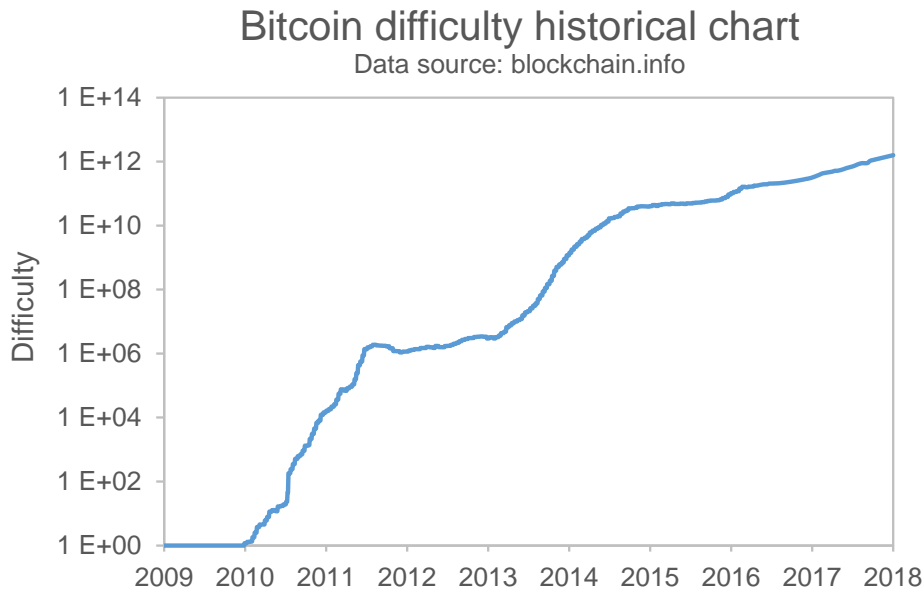T: is the real time needed to create the 2016 blocks

Figure 2.3: Mining difficulty. Represents the evolution of Bitcoin mining difficulty over the last years in a semi-logarithmic scale.

2.2.3 Proof of Work vs Proof of Stake

As mentioned before, Proof of Work (PoW) is a piece of data which is difficult (costly, time-consuming) to produce but easy for others to verify and fulfills certain requirements. A Proof of Work is a random process with low probability so that a lot of trial and error is required on average before a correct Proof of Work is generated[8].

Mining serves to verify the legitimacy of a transaction, or avoiding the so-called double-spending[6]. It also serves to create new digital currencies by rewarding miners for performing the previous task.

All the network miners compete to be the first to find a solution for the mathematical problem that concerns the candidate block, a problem that cannot be solved in other ways than through brute force so that essentially requires a huge number of attempts.

When a miner finally finds the right solution, he/she announces it to the whole network at the same time, receiving a cryptocurrency prize (the reward) provided by the protocol.

Proof of Stake (PoS) is an alternate way of verifying and validating the transaction or block. PoS is a distributed consensus protocol for distributed networks that secures a cryptocurrency network by requesting evidence of possession of said currencies[8]. With PoS the probability of finding a block of transactions, and receiving the corresponding prize, is directly proportional to the amount of coins that one has accumulated (thus avoiding the confidence given by the amount of work invested).

This system is based on the assumption that those who own more units of a PoS-based currency are especially interested in the survival and proper functioning of the network that gives value to these currencies and therefore they are the best suited to have the responsibility to protect the system from possible attacks. That is why the protocol rewards them with a lower difficulty in finding blocks (it is inversely proportional to the number of coins they demonstrate to own). In PoS, a validator (equivalent of "miner" in PoW) is picked based on the amount of stake (coins) and the respective age of the stake. In other words, a validator that holds a significant amount of stake with good aging will get a higher chance to validate a block[8].

With PoS the probability of finding a block of transactions is directly proportional to the number of cryptocurrencies accumulated, which implies that the wallet is connected to the network. This way, the wallet is exposed to possible security problems. Trying to avoid this problem, a variant of the protocol called Delegated Proof of Stake (DPoS) has been developed. In DPoS, the cyrptocurrency owner nodes allow to delegate their privileges to build new blocks in a new type of nodes called witnesses[9]. An example of platform using this algorithm is EOS.

PoS offers the following advantages compared to PoW:

- In order to attack the system a lot of money is required. Although someone manages to collect the sizeable amount of money, he will suffer from the attack since the stability of the cryptocurrency will be disturbed.

- No need of expensive and specific hardware.

- It is possible to achieve a high level of security similar to PoW but with a lower energetic cost.

- Faster validations.

PoS has also some disadvantages compared to PoW:

- Since the wallet is online connected to proof the stake, it exposes to security issues.

- It is more difficult to maintain anonymity since the funds in stake are protected with an IP address.

- Too much power for the biggest coin owners.

Figure 2.5: Proof of Stake representation. Reprinted from https://www.eventbrite.com

2.2.4 Address, public and private digital keys

Digital keys give access to the cryptocurrencies stored in a digital wallet. In the case of Bitcoin, digital keys come in pairs: private and public key. We can think of the public key as if it wpuld be the number of a bank account and the private key as if it would be the secret PIN.

A private key and a public key are mathematically related, and in fact, the public key is derived from the private key. While it is possible to create the public key from a private key, it is practically impossible to create the private key from a public key[10].

The possession of both keys automatically determines the control over the bitcoins that are stored in a Bitcoin wallet and therefore it is important that the private key always remains under the control of the owner of the account. In addition, it is important to note that digital keys in Bitcoin are created and stored by users (or can be generated

and managed by the user's wallet software) and, therefore, are completely independent of the Bitcoin protocol[11].

An example of a private key of Bitcoin could be:

```
18E14A7B6A307F426A94F8114701E7C8E774E7F9A47E2C2035DB29A206321725
```

An example of a public key of Bitcoin could be:

```
0450863AD64A87AE8A2FE83C1AF1A8403CB53F53E486D8511DAD8A04887E5B2352
2CD470243453A299FA9E77237716103ABC11A1DF38855ED6F2EE187E9C582BA6
```

The most common way to send bitcoins is to an address, which is a hash (mathematical process) of a Bitcoin public key. A Bitcoin address is a string of numbers and letters that are normally produced from public and private keys and that we could define as the fingerprint of those keys.

The reason of working with Bitcoin address (hash of Bitcoin key) and not public keys is because it gives more security to the user, since the public key will only be used when the money is transferred. These public and private key versions are called public and private addresses. The private address is the one that gives access to the bitcoins and therefore must be kept secret. The public address is used to share with other Bitcoin users to indicate the transaction recipient[10].

An example of a public address of Bitcoin could be:

```
16UwLL9Risc3QfPqBUvKofHmBQ7wMtjvM
```

An example of a private address of Bitcoin could be:

```
1CC3X2gu58d6wXUWMffpuzN9JAfTUWu4Kj
```

2.3 Cryptocurrency analysis and comparison

In this section a short description and analysis of the most relevant cryptocurencies will be done. Furthermore, a comparison between these currencies will be made.

2.3.1 Bitcoin

Bitcoin is an electronic payment system that came into existence in 2009 with the release of the first open source bitcoin client and the issuance of the first bitcoins. It is characterized by being built based on a cryptographic protocol. For this reason, it offers a high level of security since it has been observed resistant to fraud, falsification and other attacks. It offers anonymity to its users because transactions are not associated with personal data[12].

The system is distributed and decentralized (it does not depend on a central authority, such as governments or banks) so it allows transactions to be carried out without intermediaries, reducing the operation cost compared to other systems such as Paypal. This decentralization also means that the Bitcoin network is controlled and owned by all of its users and, as all users must adhere to the same set of rules, there is a great incentive to maintain the decentralized nature of the network[13].

Bitcoin uses blockchain technology, which keeps a record of every single transaction, and the transaction and authentication process is carried out by the network of users. Although the decentralized nature offers many advantages, critics often argue that apart from its users, there is nobody overlooking the whole system and that the value of Bitcoin is unfounded.

In return for contributing their computing power to the network to carry out some of the tasks mentioned above, also known as mining, users are rewarded with Bitcoins. The processing power of the miners is used to validate transaction blocks and adding them to the blockchain. As mentioned before, Bitcoins are mined using a proof-of-work function by individual miners and verified by the decentralized nodes in the peer to peer Bitcoin network[4].
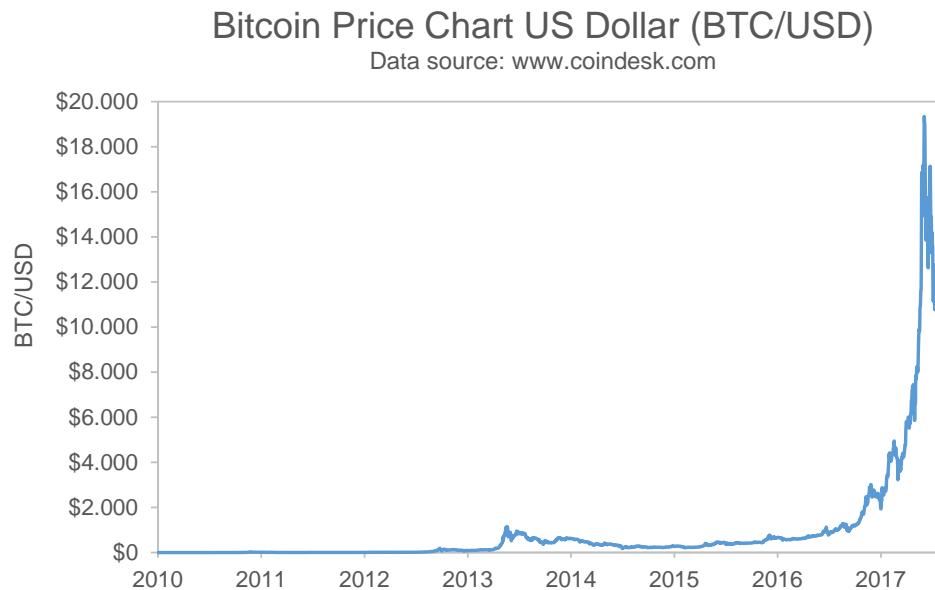

Figure 2.6: Bitcoin to US dollar evolution rate

2.3.2 Ether

Ethereum is an open source decentralized platform that allows the creation of intelligent contract agreements between peers based on the blockchain model without any possibility of downtime, censorship, fraud or third-party interference[14]. Any developer can create and publish distributed applications that make intelligent contracts. Ethereum also provides a cryptocurrency token called 'ether'[2].

The Ethereum network enables developers to create markets, store registries of debts or promises and move funds in accordance with instructions given long in the past (like a will or a futures contract) without a middleman or counterparty risk. The Ethereum protocol is built to allow flexibility to users and increase the functionality of the system as it progresses and provides the ability to program many different types of smart contracts within the Ethereum system[15].

Ethereum offers the possibility to create decentralized applications[2]. Unlike other centralized applications, Ethereum decentralized applications do not need servers or other centralized entities. The application lives in the blockchain, as well as all its content. Typical applications such as Facebook are centralized, that is, we deposit our trust (data, photos, videos and content in general) in a main entity or server.

In the same way as Bitcoin, Ether tokens are put into circulation through a mining process. Although Ether tokens are also mined using a proof-of-work function, it is planned to make a leap towards a proof-of-stake function or PoS. In PoS, the reward system changes drastically so that the electricity cost is not so bulky[15].

In a Proof of Stake model there will no longer be miners, but validators. Instead of difficult mathematical functions that the miners must solve, validators will be required to have Ether and to be able to validate a block. Validators take turns proposing and voting on the next block, and the weight of each validator's vote depends on the size of its deposit (i.e. stake). Significant advantages of PoS include security, reduced risk of centralization and energy efficiency.

In chain-based Proof of Stake, the algorithm pseudo-randomly selects a validator during each time slot (eg. every period of 10 seconds might be a time slot), and assigns that validator the right to create a single block, and this block must point to some previous block (normally the block at the end of the previously longest chain), and so over time most blocks converge into a single constantly growing chain[15].
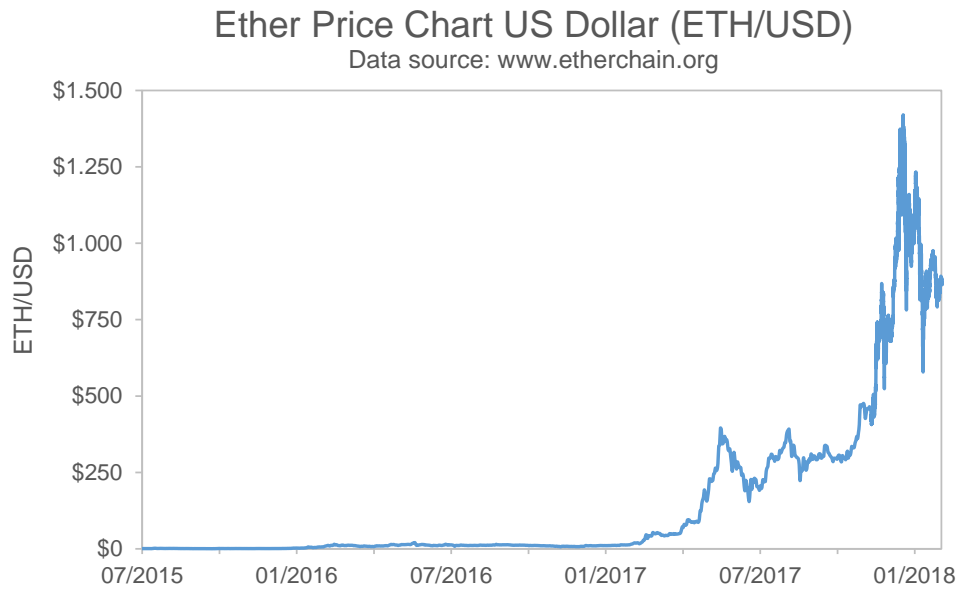
Ether Price Chart US Dollar (ETH/USD)

Data source: www.etherchain.org

Figure 2.7: Ether to US dollar evolution rate

2.3.3 Litecoin

Litecoin is a decentralized P2P cryptocurrency created in October 2011. It was created as an open source project inspired by Bitcoin, so it is virtually identical in most of its technical aspects. However, there are at least three fundamental differences between Litecoin and Bitcoin: its speed, its number of coins and changes in the algorithm[16].

First of all, one of the biggest differences is the speed at which transactions are made. Litecoin network performs the processing of a block every 2.5 minutes instead of every 10 minutes[16], which means that it generates the blocks of its chain four times faster than Bitcoin. Because of this, transactions are carried out at a higher speed.

Another difference is the amount of coins planned to generate. One of the things that gives more value to Bitcoin than conventional currencies is that the network generates a fixed amount of coins every year, and the total number of Bitcoins is finite. Bitcoin will stop creating new units when it reaches 21 million coins, while Litecoin will stop at 84 million[16].

Finally, Litecoin is the first coin that changed the hashing algorithm and used scrypt[16]. This allows to use any PC to dedicate itself to mining, helping non-professional miners to participate in the network.
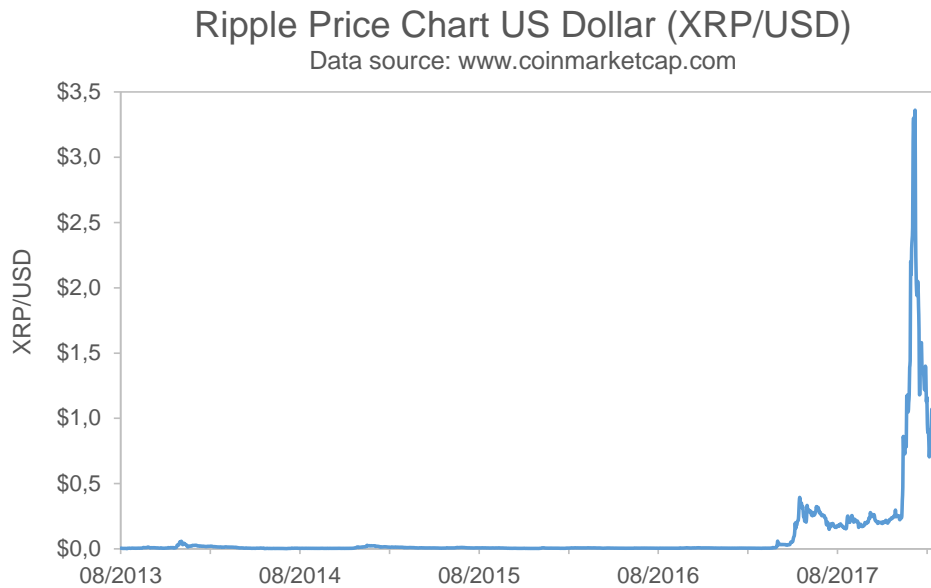
Figure 2.8: Litecoin to US dollar evolution rate

2.3.4 Ripple

Originally released in 2012 as a subsequent iteration of Ripplepay, Ripple is a real-time gross settlement system (RTGS), currency exchange and remittance network. Using a common ledger that is managed by a network of independently validating servers that constantly compare transaction records, Ripple does not rely on the energy and computing intensive proof-of-work used by Bitcoin[17]. Ripple is based on a common public database that uses a consensus process between validating servers to ensure integrity. Validating servers can be owned by anyone, from individuals to banks.

The Ripple protocol (token represented as XRP) is meant to enable the near instant and direct transfer of money between two parties. Any type of currency can be exchanged, from fiat currency to gold. Ripple technology claim to avoid the fees and wait times of traditional banking and even cryptocurrency transactions through exchanges[18].

Ripple transactions rely on a consensus protocol in order to validate account balances and transactions on the system. The consensus works to improve the integrity of the system by preventing double spending. Individual distributed nodes decide by consensus which transaction was made first by taking a poll to determine the majority vote. The confirmations are instant and take roughly 5 seconds[17]. Since there is no central authority that decides who can set up a node and confirm transactions, Ripple platform is described as decentralized.

Thus far, Ripple has been stable since its release with over 35 million transactions processed without issue. It is able to handle 1,500 transactions per second (tps)[18] and has been updated to be able to scale to Visa levels of 50,000 transactions per second. By comparison, Ethereum can handle 15 tps and Bitcoin 3 to 6 tps.

Ripple's token, XRP, is not mined like Bitcoin, Ethereum, Litecoin and many other cryptocurrencies. Instead, Ripple issued a fixed number of tokens at its inception, 100 billion of XRP[18]. 61 billion of them are nowadays under control of the Ripple company and they are planned to be released at a rate of 1 billion XRP a month for at least four and a half years.

There are three components forming Ripple: Ripple Labs, the parent company, based in San Francisco, which has raised nearly $ 100 million in funds; RippleNet, the payment network, now used by important partners such as American Express; and XRP, the settlement token of the Ripple network.
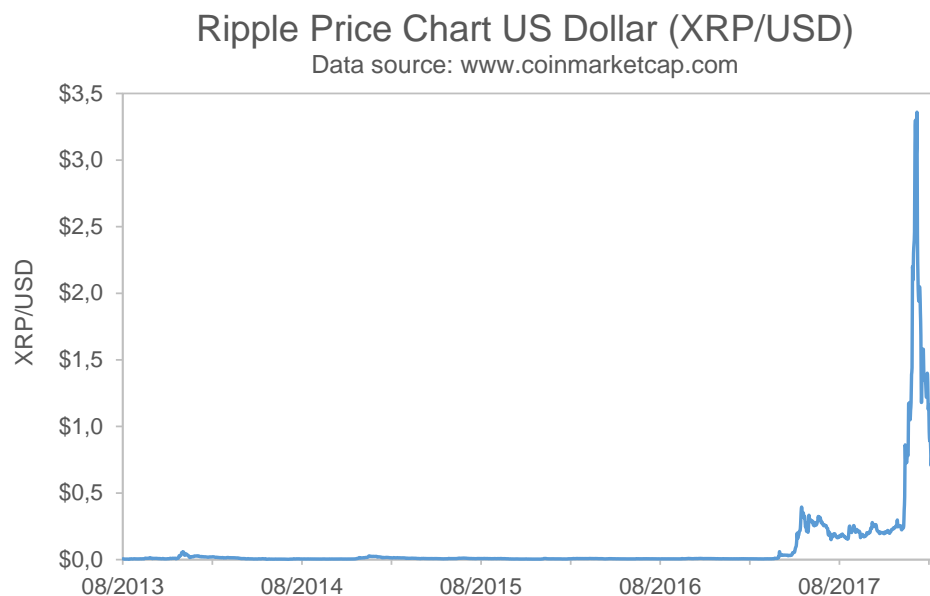
Figure 2.9: XRP to US dollar evolution rate

2.3.5 Cryptocurrency comparison

In this section, a more detailed comparison will be made between top 10 cryptocurrencies, based on market capitalization. Most suitable characteristics and properties have been chosen in order to analyze and compare the main differences between cryptocurrencies.

Table 2.1 shows main properties from top 10 cryptocurrencies based on market capitalization as of 27th February 2018. Figure 2.10 represents the market capitalization of

each coin in million US dollars and as a proportion of the total market capitalization including alternative cryptocurrencies.

Coin	Symbol	Market capitalization [1]	Market Cap(%) [1]	Price USD [1]	Price BTC [1]	Circulating supply [1]	Max supply	Transaction Time [1]	Mining	Mining difficulty [1]	Proof type	Hash algorithm
Bitcoin	BTC	$180.660.002.170	39,49%	$10.697	1 BTC	16.889.650	21.000.000	10 minutes	Processor intensive	3.007.383.866.429	PoW	SHA-256
Ethereum	ETH	$86.238.962.573	18,85%	$ 881	0,083 BTC	97.877.341	Inflationary	12-14 seconds	Memory intensive	3.103.058.930.915.360	PoW	KECCAK-256
Ripple	XRP	$37.255.743.602	8,14%	$ 1,0	0,089 mBTC	38.291.387.790	100.000.000.000	4 seconds	Not mineable	Not mineable	RPCA[2]	SHA-512
Bitcoin Cash	BCH	$21.204.312.207	4,63%	$ 1.248	0,12 BTC	16.482.113	21.000.000	10 minutes	Processor intensive	352.589.267.721	PoW	SHA-256
Litecoin	LTC	$12.001.318.815	2,62%	$ 216,6	0,02 BTC	52.051.682	84.000.000	2,5 minutes	Memory intensive	5.034.453	PoW	Scrypt
Neo	NEO	$9.045.725.000	1,98%	$ 139,2	0,013 BTC	50.000.000	100.000.000	15-20 seconds	Not mineable	Not mineable	PoS	-
Cardano	ADA	$8.656.815.509	1,89%	$ 0,3	0,031 mBTC	25.927.070.538	45.000.000.000	20 seconds	Not mineable	Not mineable	PoS	-
Stellar	XLM	$6.656.263.930	1,45%	$ 0,4	0,033 mBTC	18.468.076.589	100.000.000.000	20 seconds	Not mineable	Not mineable	PoS	-
EOS	EOS	$6.002.921.137	1,31%	$ 8,6	0,81 mBTC	695.886.137	900.000.000	30 seconds	Not mineable	Not mineable	DPoS[3]	-
IOTA	IOT	$5.253.590.188	1,15%	$ 1,9	0,17 mBTC	2.779.530.283	2.779.530.283	1-3 minutes	Not mineable	Not mineable	None	-

[1] As of 27th February 2018. Source: https://coinmarketcap.com

[2] Ripple Protocol Consensus Algorithm

[3] Delegated Proof of Stake

Table 2.1: Top 10 cryptocurrencies comparison. Elaborated by the author.
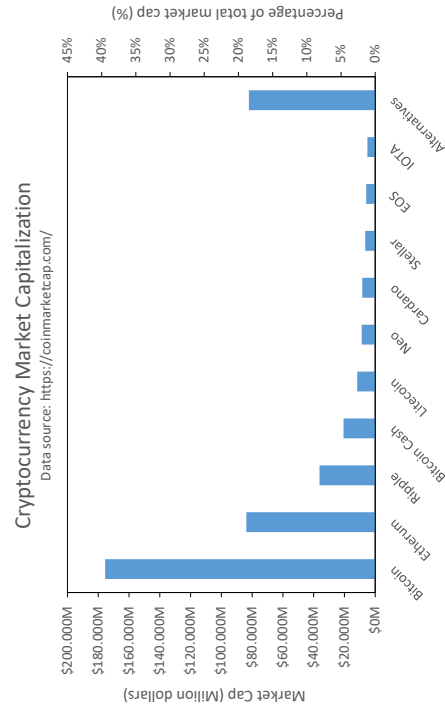

Figure 2.10: Top 10 cryptocurrency market capitalization. Elaborated by the author.

2.4 Cryptocurrency exchanges analysis and comparison

Cryptocurrency exchanges are businesses, normally in form of websites or platforms that allow customers to trade digital currencies for other assets, such as conventional fiat money (USD, EUR, etc.) or different digital currencies. The profit of these exchanges is made in the transactions. They can take the spread of bid/ask as a transaction commission for their services or charge fees as a matching platform. Cryptocurrency exchanges accept many payment methods, such as bank transfer, postal money orders, credit card payment or cryptocurrencies.

The first Bitcoin currency exchange called BitcoinMarket appeared in February 2010 and it was created by a user from the Bitcointalk forum[2]. In May 2010, the first financial transaction using Bitcoins took place. A different user from Bitcointalk forum offered to pay 10,000 Bitcoins for a couple of pizzas[3] and made the deal.

Nowadays there are more than 120 active cryptocurrency exchanges. They differ in many areas such as reputation, fees, payment methods, verification requirements, coins and base currency offered and the exchange rate.

2.4.1 Exchange connectivity

Cryptocurrency exchanges provide APIs (Application Programming Interface) to allow users to connect their own software directly to the exchange server. They offer three types of connection between client and server:

- **REST API**. REST, or REpresentational State Transfer, is an architectural style for designing network applications[19]. REST APIs are interfaces between systems that use HTTP to obtain data or indicate the execution of operations of the data, in any format (XML, JSON, etc). The vast majority of web applications are architected using HTTP through a REST API[19]. REST APIs are normally programmed in Ruby, Java, Go or NodeJS and they are fundamentally similar in that they receive Request of information and then make a Response to them. Because REST APIs are based on HTTP it has some limitations, in particular the way that connections are handled. Every time a request is made (e.g. BTC actual price), a port or socket is opened and data is transferred. Once the data transfer is completed, the port is closed. This means that every time that the user request some information to the server a gate is opened and closed, which creates overhead. Obviously, this is not suitable for applications that need real time interactions or display big streams of data. Another limitation of REST APIs is related to the HTTP "pull" standard. The client has to request or pull information from the server, because the server can't push the data to the client when is needed. This

---

[2] Global online Bitcoin forum. https://bitcointalk.org/.
[3] Pizza for Bitcoins. https://bitcointalk.org/index.php?topic=137.0.

means that the client needs to send requests to the server repeatedly to check if there is new information available (e.g. new price level in the order book).

- **WebSocket API**. Unlike REST APIs, WebSocket allow bi-directional connection in which the server and client can continuously send messages back and forth[20]. This means that the connection between server and client remains open until the client decides to close it. Because of that, WebSocket APIs are more suitable for real time applications because the server can send the information directly to the client the very moment it changes on the server[20]. In the case of using a REST call, the client needs to request the information every some fixed time interval and will only get new data at the point of their polling interval. Moreover, in order to avoid server saturation, exchanges limit the number of requests per second per client to the server. Usually, the request limit fixed by the cryptocurrency exchanges using REST APIs is around 1 request per second. If the call rate exceeds the exchange limit, the exchange bans the IP of the client and the connection is lost for a certain time. In the case of WebSocket, there is no request limit since the client does not need to send request calls to the server every time data has changed. In WebSocket, the server sends new data to the client automatically every time it is updated, following a "push" pattern. WebSocket technology offers a faster and easier connectivity on a networking infrastructure, because less operations are done to send a packet over an existing WebSocket connection[20].

- **FIX API**. FIX (Financial Information Exchange) protocol facilitates the transfer of electronic information between a trader and liquidity provider, allowing a quick and accurate execution[21]. Nowadays, FIX protocol is the technology used by the global financial markets and it is extensively used by trading platforms, buy and sell-side firms and even regulators to communicate trade information[21]. The protocol does not have a proprietary; it is free and open. Its standards are constantly being developed to support evolving business and regulatory needs, and is used by thousands of firms every day to complete millions of transactions. Although FIX protocol would be suitable for real time applications, such as a real time trading robot, very few exchanges offer this technology and, within this exchanges, most of them have their FIX API outdated.

Below the network operations done to gather price data from a cryptocurrency exchange using a REST API are summarized:

1. Client defines a polling interval.

2. In the next polling interval trigger, the client creates a new socket connection to the server.

3. The server receives the request to open a new socket with the client.

4. When the handshake is made with the server, the client sends a request for the new pricing information to the server.

5. The server receives the request for new pricing info and replies with new data (if any).

6. The client receives the new pricing information.

7. The client closes the socket.

8. The server receives socket close

On the other hand, the network operations needed to gather new data from an already open Websocket connection are summarized below:

1. The server register a price change and immediately sends a message to every client.

2. The client receives the message about new pricing information.

As may be seen, WebSocket technology is much more efficient and suitable than REST APIs for applications that need real time data or when the client needs to request data to the server continuously, for example, in a real time trading robot.

To understand better the difference between both methodologies, below is represented an analysis using WebSocket API and REST API. Figure 2.11 shows the time needed to process N messages of a constant payload size (1000 bytes) using both technologies.
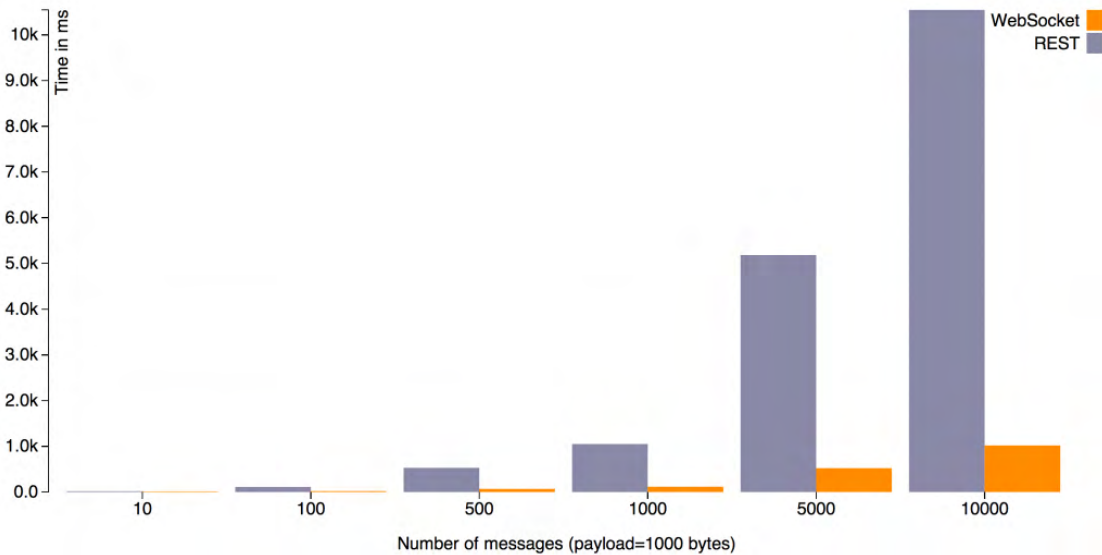

Figure 2.11: Connectivity test between protocols showing the increasing overhead of REST API compared to WebSocket API due to multiple opening and closing connections. Reprinted from: REST vs WebSocket Comparison and Benchmarks, by Gupta, A. (2014). http://blog.arungupta.me/rest-vs-websocket-comparison-benchmarks/

As can be noted, REST API overhead increases when the number of messages increase. This is caused because the more messages need to be sent, the more connections between client and server need to be initiated and terminated. By contrast with WebSocket, the client only needs to do the initial handshake with the server, so the messages can be processed directly.

Because of Websocket superiority in real time systems, this project will try to use this technology to interact with the cryptocurrency exchanges. Although Websocket offers a significant improvement compared to REST, the programming difficulty to connect to a WebSocket API is considerable.

2.4.2 Exchange analysis

In this section a short description and analysis of the most relevant cryptocurrency exchanges will be done. Furthermore, a detailed comparison between this exchanges will be presented.

Coinbase

Coinbase is a cryptocurrency exchange based in the USA founded in 2012. Is one of the most recognized and accredited platforms that supports users from more than 32 countries. In 2014, Coinbase created GDAX, the Global Digital Asset Exchange, developed for professionals with high trading volumes. GDAX functions as a traditional exchange offering a real-time market to negotiate 4 different cryptocurrencies and 16 market pairs. Compared to Coinbase, GDAX is an advanced platform that allows advanced order placement, such as Market or Limit order, Stop Loss and different time in force options: Good Til Cancelled, Immediate or Cancel or Fill or Kill.

Among that, GDAX also offers a better fee plan compared to Coinbase. Trading fees vary depending on whether the order is executed immediately (market order) or if it is placed on the orderbook (limit order). Limit orders are fee free while market orders are charged with a 0.25%[4] fee. In case of big trading volume, market order fees can be reduced up to 0.1%[4].

The firm offers a free mobile wallet, offline storage and insurance protection for the currency stored in their servers. GDAX supports several fiduciary currencies: US dollars, Euros and Pounds Sterling.

Regarding the exchange connectivity, GDAX offers two types of API: REST and WebSocket, both updated. There is also a FIX API used for order management.
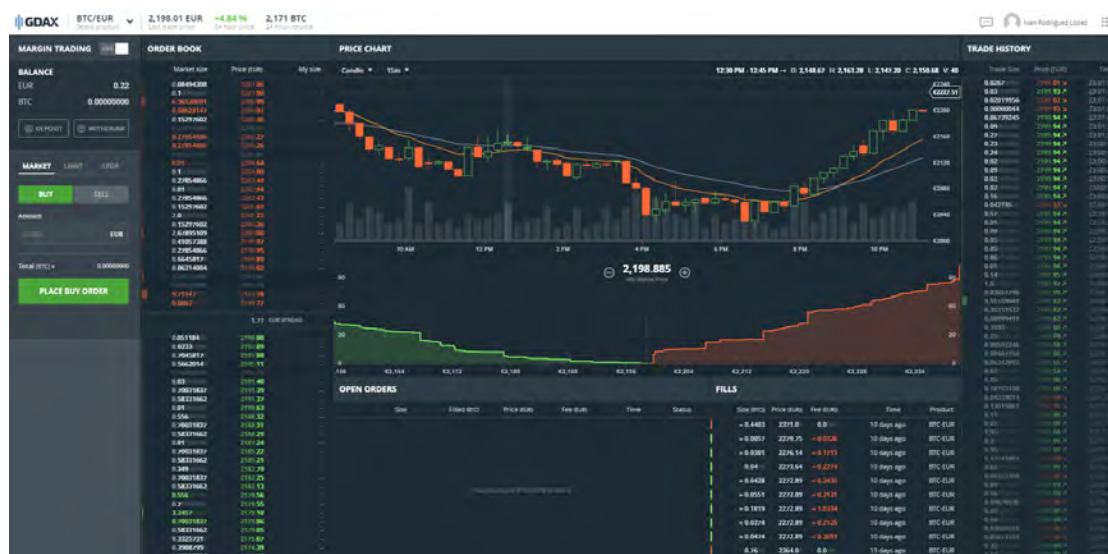
---

[4]GDAX Fee Structure https://www.gdax.com/fee.

Figure 2.12: GDAX User Interface.https://www.gdax.com/

Bitfinex

Bitfinex is a cryptocurrency trading platform based in Hong Kong that became famous for its leverage services of Margin Trading: The exchange supports clients from worldwide countries, except USA. Bitfinex offers the possibility to obtain "loans" when carrying out cryptocurrency purchase transactions, as long as the initial equity of the funds held have to be at least the 30%[5] of the position opened. Bitfinex also allows the possibility to open short positions. A short position is a way of trading where the investor sells borrowed currencies in the market. The idea is that an investor expects that the price will decrease over time and decides to sell the borrowed currencies and purchase them again in the next future, so that he can return them to the entity, in this case the exchange, which he borrowed them from. By doing this, it is possible to have profit even if the price of a cryptocurrency is decreasing.

Bitfinex offers 16 different cryptocurrencies with 38 market pairs and US Dollars as its base currency. Market order fees range from 0,1%[6] to 0,2%[6] depending on the volume traded. Limit order fees range from 0%[6] to 0,1%[6] depending on the volume traded.

Regarding Bitfinex networking connectivity, they offer two kinds of APIs: REST and WebSocket, both updated.

---

[5]Intro to Margin Trading https://support.bitfinex.com/hc/en-us/articles/115004555165-Intro-to-Margin-Trading.
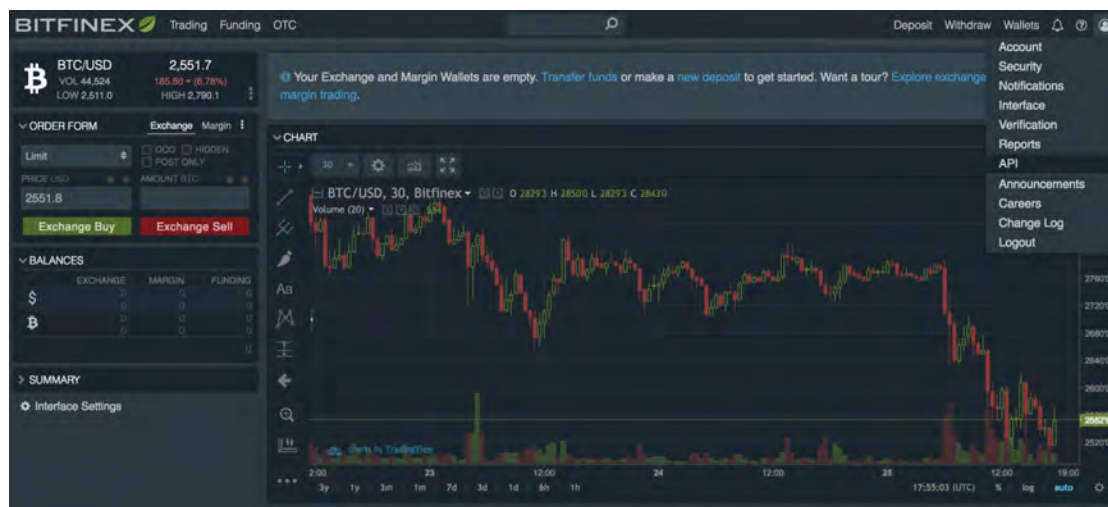
[6]Bitfinex Fees Schedule https://www.bitfinex.com/fees.

Figure 2.13: Bitfinex User Interface.https://www.bitfinex.com/

HitBTC

HitBTC is a cryptocurrency exchanges that offers a large number of cryptocurrencies and rising altcoins. HitBTC also offers lots of market to negotiate with different tokens and ICOs. HitBTC is registered in the UK since 2013. Its volume has been steadily rising which provides a good liquidity with a narrow bid ask spread.

HitBTC offers a total of 149 different digital coins in 160 market pairs and supports two fiduciary currencies: US Dollars and Euros. Trading fees vary depending on whether it is placed a market order or limit order. Market orders are charged with a 0,1%[7] fee from the trade while limit orders are not charged. Moreover, limit orders receive a 0,01%[1] rebate from the trade.

HitBTC offers two types of APIs: REST and Websocket. Recently, a new version of the API has been released, called APIv2.
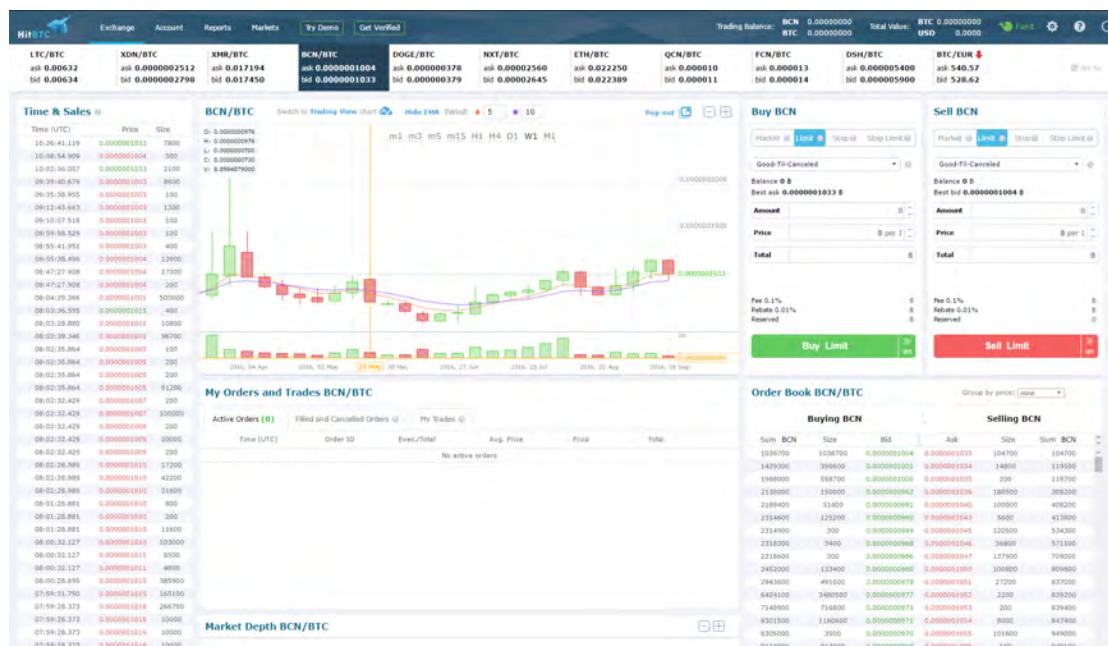
---

[7]HitBTC trading fees https://hitbtc.com/fees-and-limits.

27

Figure 2.14: HitBTC User Inteface.https://hitbtc.com/

OKCoin

OKCoin is one of the largest Bitcoin exchange in the world. The exchange was founded in 2013 and its based in Beijing, China. Currently, the exchange is focused on the Chinese market, providing cryptocurrency trading services to the rest of the world. OKCoin offers 5 cryptocurrencies that can be bought with US Dollars. This exchange offers also the possibility to leverage up to 20 times. Trading fees are fixed at 0,2%[8] whether is a market order or limit order.

OKCoin has a strong institutional support, since it has been receiving funds and investments from important actors, such as VenturesLab, or Chinese funds, Ceyuan and Longling Capital.

Similarly to Coinbase, OKCoin has also a second exchange called OKex whose target are more experienced traders. In OKex, trading fees range between 0,02% and 0,15% for limit orders and between 0,05% and 0,2% for market orders, depending on the volume traded.

In the last years, opening an account with this broker has been more complicated, specially for US citizens. This is linked to the fact that the US Securities and Exchange Commission (SEC), is trying to limit as much as possible that US traders can trade in brokers of foreign countries that are not legally registered in the United States[22].
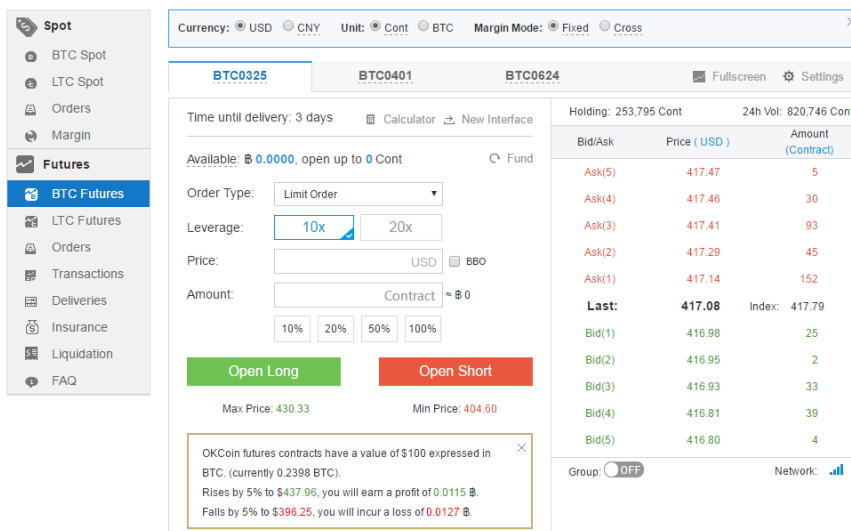
---

[8]OKCoin trading fees https://www.okcoin.com/fees.html.

Figure 2.15: OKCoin User Interface.https://www.okcoin.com/

2.4.3 Exchange comparison

In this section, a more detailed comparison based on traded volume between most important cryptocurrency exchanges will be made. Most suitable characteristics and properties have been chosen in order to analyze and compare the main differences between exchanges.

Table 2.2 and Table 2.3 show main properties from top cryptocurrency exchanges based on trading volume as of 7th March 2018. Trading volumes presented in Table 2.2 are calculated as the average of the monthly volume between January 2018 and February 2018. Table 2.3 shows the base currencies and the most important digital coins offered by the exchanges in their different market pairs.

Digital coins presented in Table 2.3 are: Bitcoin (BTC), Ether (ETH), Bitcoin Cash (BCH), Ripple (XRP), Litecoin (LTC), Digita Cash (DASH), NEM(XEM), IOTA (IOT), Monero (XMR), OmiseGO(OMG), NEO (NEO), Ethereum Classic (ETC), EOS (EOS), Lisk (LSK), ZCash (ZEC) and TenX (PAY). Base currencies presented in Table 2.3 are: US Dollar (USD), Euro (EUR), Yen (JPY), Pound (GBP), Canadian Dollar (CAD), Won (KRW), Yuan Renminbi (CNY), Bitcoin (BTC), Ether (ETH), Bitcoin Cash (BCH), Monero (XMR) and Tether Dollar (USDT).

Table 2.2: Cryptocurrency exchange comparison. Elaborated by the author.

Exchange	Monthly trading volume ($) [1]	Number of digital coins [2]	Number of base currency [2]	Number of market pairs [2]	Most traded pair [2]	Country [2]	Has been hacked?	WS? [4]	REST? [4]	REST call rate (requests/min)
Binance	$2.700.000.000	114	3	262	BTC/USDT	Hong Kong [3]	No	No	Yes	1200
Bithumb	$2.000.000.000	12	1	12	XRP/KRW	South Korea [3]	Yes	No	Yes	120
Okex	$1.900.000.000	90	4	271	BTC/USDT	Hong Kong [3]	Yes	Yes	Yes	600
Bitfinex	$1.800.000.000	38	4	105	BTC/USD	Hong Kong [3]	Yes	Yes	Yes	60
Bittrex	$1.450.000.000	199	3	273	BTC/USDT	US	No	No	Yes	60
GDAX	$860.000.000	4	4	12	ETH/USD	US	No	Yes	Yes	180
Huobi	$850.000.000	97	3	202	BTC/USDT	Singapore [3]	No	Yes	Yes	60
Poloniex	$725.000.000	68	4	98	BTC/USDT	US	Yes	Yes	Yes	360
Kraken	$650.000.000	17	7	57	BTC/EUR	US	No	No	Yes	60
Bitstamp	$580.000.000	5	3	14	BTC/USD	Luxembourg	Yes	Yes	Yes	60
HitBTC	$505.000.000	300	4	562	BCC/BTC	UK	No	Yes	Yes	60
Coinone	$440.000.000	9	1	9	XRP/KRW	South Korea [3]	No	No	Yes	90
Coincheck	$420.000.000	1	1	1	BTC/JPY	Japan [3]	No	No	Yes	60
Bitflyer	$280.000.000	3	2	3	BTC/JPY	Japan [3]	No	No	Yes	200
OKCoin	$110.000.000	3	1	6	BTC/CNY	China [3]	No	Yes	Yes	600

[1] Monthly average volume of January and February 2018. Source: https://bitgup.com/

[2] As of 7th March 2018. Source: https://bitgup.com/

[3] In Asia, order filling delay of 2-3 seconds

[4] Evaluation of the type of API offered by the exchange (REST or WebSocket)

Table 2.3: Digital coins and base currencies offered by exchanges. Elaborated by the author.

Exchange	Base currency [4]												Digital coins offered [4]														
	USD	EUR	JPY	GBP	CAD	KRW	CNY	BTC	ETH	BCH	XMR	USDT	BTC	ETH	BCH	XRP	LTC	DASH	XEM	IOT	XMR	OMG	NEO	ETC	EOS	LSK	ZEC	PAY
Binance								X	X			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
Bithumb						X							X	X	X	X	X	X						X			X	
Okex								X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X
Bitfinex	X	X						X	X	X			X	X	X	X	X	X	X	X		X	X	X	X		X	X
Bittrex	X							X	X			X	X	X	X	X	X	X				X	X	X			X	X
GDAX	X	X						X					X	X	X		X											
Huobi				X				X	X			X	X	X	X	X	X	X	X			X	X	X	X	X	X	X
Poloniex								X	X		X	X	X	X	X	X	X	X	X			X	X	X	X	X	X	
Kraken	X	X		X	X			X	X				X	X	X	X	X	X			X			X	X		X	
Bitstamp	X	X		X	X			X	X	X			X	X	X	X	X	X	X			X	X	X	X		X	X
HitBTC	X							X	X	X			X	X	X	X	X	X		X	X			X	X			
Coinone			X			X							X															
Coincheck			X										X															
Bitflyer			X					X					X				X											
OKCoin	X						X						X	X	X									X				

[4] As of 7th March 2018. Source: https://www.coinhills.com/

Fees

The fee schedule of the most important cryptocurrency exchanges is presented below. Table 2.4 summarizes the most relevant exchange fees: trading, deposit and withdrawal fees.

Table 2.4 also contains information about the total transaction cost for a normal investment operation. That is, deposit fiat money at an exchange, buy some amount of cryptocurrency, sell the amount purchased and withdraw the fiat funds of the sale. Table 2.4 shows the total transaction cost of this operation in the case of investing in 1 Bitcoin as a percentage of the Bitcoin price.

Total transaction cost of purchasing and selling in relation to the amount of Bitcoins invested is shown in Figure 2.16 and Figure 2.17. More specifically Figure 2.16 represents the total fee for market order execution and Figure 2.17 illustrates the total fee for limit order execution.

Exchange	Trading fee[1]		Withdrawal fee[1]		Deposit[1]		Round trip transaction cost			Total fee for deposit-buy-sell-withdraw [1] BTC as market taker (%)	Total fee for deposit-buy-sell-withdraw [1] BTC as market maker (%)
	Taker fee	Maker fee	Base currency	Digital coin (BTC)	FIAT money	Digital coin (BTC)	Fix (€)	Variable (%)			
								Market taker	Market maker		
Binance	0,10%	0,10%	$10,80	0,0005	0	0	8,71 €[2]	0,20%	0,20%	0,31%	0,31%
Bitflyer	0,15%	0,15%	756 JPY	0,0004	324 JPY	0	8,24 €[2]	0,30%	0,30%	0,40%	0,40%
Bitfinex	0,20%	0,10%	0,10%	0,0005	0,10%	0	0,00 €	0,60%	0,40%	0,60%	0,40%
Bithumb	0,15%	0,15%	1.000 KRW	0,0005	0	0	0,75 €[2]	0,30%	0,30%	0,31%	0,31%
Bittrex	0,25%	0,25%	0	0,001	0	0	0,00 €	0,50%	0,50%	0,50%	0,50%
Coincheck	0,15%	0,00%	400 JPY	0,0005	1.000 JPY	0	10,69 €[2]	0,30%	0,00%	0,43%	0,13%
HitBTC	0,10%	0,09%	30 €	0,00085	5 €	0	35,00 €	0,20%	0,18%	0,64%	0,62%
Poloniex	0,25%	0,15%	0	0,0001	0	0	0,00 €	0,50%	0,30%	0,50%	0,30%
Coinone	0,10%	0,10%	1.000 KRW	0,0005	0	0	0,75 €[2]	0,20%	0,20%	0,21%	0,21%
Bitstamp	0,25%	0,25%	0,90 €	0	0	0	0,90 €	0,50%	0,50%	0,51%	0,51%
Kraken	0,26%	0,16%	0,09 €	0,001	0	0	0,09 €	0,52%	0,32%	0,51%	0,32%
GDAX	0,25%	0,00%	0,15 €	0	0	0	0,15 €	0,50%	0,00%	0,50%	0,00%
OKCoin	0,20%	0,20%	0,50%	0	0	0	0,00 €	0,90%	0,90%	0,90%	0,90%

[1] Accessed November 2017 at exchanges official site
[2] Currency conversion rate calculated as of 7th March 2018. Source: https://www.bloomberg.com

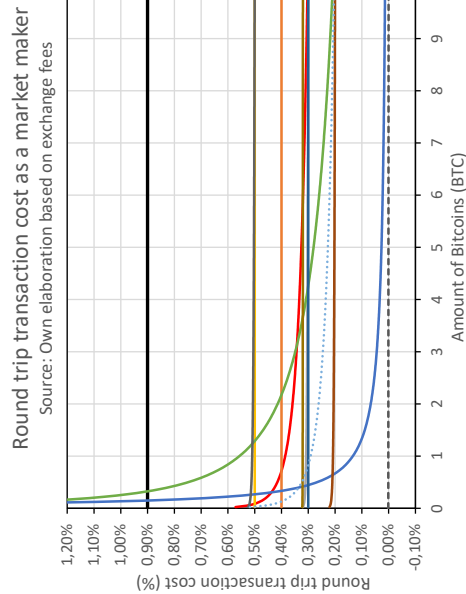**Table 2.4:** Top exchanges fee schedule. Elaborated by the author.

Figure 2.16: Round trip market order transaction fee. Elaborated by the author.
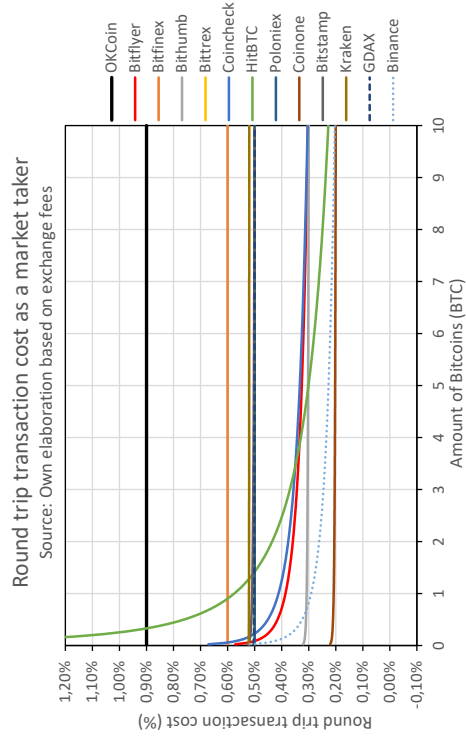

Figure 2.17: Round trip limit order transaction fee. Elaborated by the author.

Trading platform

In the following chapter a description of the software and programs implemented will be provided. In addition, an installation and user's guide will be detailed.

3.1 Base project

As mentioned before, this project will try to use WebSocket technology to connect with exchanges because of its superiority, in terms of communication speed, compared to other kinds of connectivity technologies, such as REST API. Despite this fact, the complexity of establishing a connection with the port or socket using WebSocket is significantly greater. Because of the short period of this project and due to programming and coding difficulty, it has been decided to create a program based on other already designed open-source projects.

The criteria to choose the appropriate existing project is the following:

- The project has to be free and open-source.
- It should have implemented a transaction system as well as a market data gathering system.
- The project should be well documented.
- Ideally, the programming language should be a high level language such as Python or Matlab.

After an extensive search, installation and testing of the existing projects that matched this criteria[23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33], it was decided that the platform more suited to be the basis of this project was the program Tribeca. The reason of choosing Tribeca is that the majority of open-source projects were based on REST APIs instead of WebSocket APIs. The main problem of Tribeca software is that it is a very developed project with almost no coding documentation. Among that, another major problem is that the project has not been properly updated in the last year with the

last exchange APIs upgrades and, because of that, some of the functionalities were not working. Finally, Tribeca is designed as an automated market making bot and it is not programmed to send simple orders such as market orders.

In order to respond to the problems presented above, a new program called Cryptobot will be implemented based on the initial structure of Tribeca.

Tribeca is a high frequency market making cryptocurrency trading platform. Market making is a trading strategy where a trader places two orders at the same time, buying on bid and selling on ask. By placing both orders, the trader expects to earn the bid-ask spread in each double trade execution.

This type of strategy is very attractive for traders because there is no need to predict the direction of the market, they should make profit no matter if the market goes up or down. In practice, this strategy has a risk associate, especially when the market moves in one direction very fast. The major risk in this case is that the trader executes one order but not the other. If this happens, the trader will most probably lose money in the operation.

Market makers are welcomed by exchanges because they provide liquidity to their markets, since they are always ready to buy and sell from other traders. For this reason, many exchanges incentive this kind of operations by charging 0% commission on make orders or even receive a rebate from the trade.

Although Tribeca is designed as a market making trading program, this project aims to complement the platform with the possibility to make individual trades but keeping possibility of operate market making strategies.

3.2 Structure of the program

The code is organized in 3 modules:

- **The engine layer**. The central module of the application. This part of the code is designed to incorporate market data, order status, fees and open positions into the program. It is also responsible of synthesizing trades and security information and transform it into a block to send to the exchange. This part of the program calculates the fair value (`FairValueEngine`), variable used for market making purposes, and generates quotes (`QuotingEngine`). Fair value is an estimation of the price of the digital coin and it is used as a starting point to generate a quote.

- **The adapter layer**. The engine layer should not know about the individual characteristics of every exchange. The engine layer uses the adapter layer to carry out its bidding. The adapter layer also has no idea that it is being used in a manner to make markets. In theory, the adapter layer code and the gateway layer code could be divorced from the Engine layer and we could build a technical analysis

or latency arbitrage bot, instead. The adapter layer also contains all the state reported by the gateways.

- **The gateway layer**. Each exchange has their own API for interacting with the exchange. The connectivity with every API is handled by 4 different interfaces:
  - `IMarketDataGateway`: Handles order book updates and market trade updates.
  - `IOrderEntryGateway`: Send and cancel orders and handle updates to the orders.
  - `IPositionGateway`: Pulls in the latest position information (e.g. USD amount in the exchange account).
  - `IExchangeDetailsGateway`: Read-only information describing naming and exchange fee structure.

The major part of the software is programmed in TypeScript (95%), although some parts are written in HTML (4,5%) and JavaScript (0,5%). HTML and JavaScript are used to design the User Interface. The connection between TypeScript and HTML/JavaScript is made via AngularJS and Socket.io.

AngularJS is a client side JavaScript framework to design dynamic single page web applications by changing static HTML into dynamic HTML. It is commonly used in real time web applications. AngularJS extends HTML vocabulary with directives and attributes, maintaining the semantic and with no need to use external libraries.

Socket.io is a JavaScript library that runs on Node.js and is also designed to develop real time web applications. Socket.io allows two-way communication between client and server because is based on WebSocket technology.

Regarding the connectivity to the different exchanges, the platform connects simultaneously to the REST and WebSocket API from the exchange. WebSocket API is used to gather fundamental market data information from the exchange as well as send trades to the exchange. REST API is used to receive a full report of market data information and present it through the UI. This information is persisted through a MongoDB database.

Figure 3.1 shows how the different program files are interconnected. As can be noted, the main nodes of the program are `models.ts`, `messaging.ts` and `main.ts`. The file `models.ts` defines and declares most of the classes, functions and collections that are used while the program is running. `Messaging.ts` is responsible for the correct registration and understanding of the different messages received by the exchange as well as the messages published in the own program logging files. Finally, `main.ts` is the file where everything is put together. The main is structured in two parts. The first part, `liveTradingSetup` loads and initialize basic configuration such as exchange and pair configured, initialization of MongoDB database, connectivity to the UI... The central part, `runTradingSystem`, loads the setup configuration and maintains the program running in an asynchronous mode.
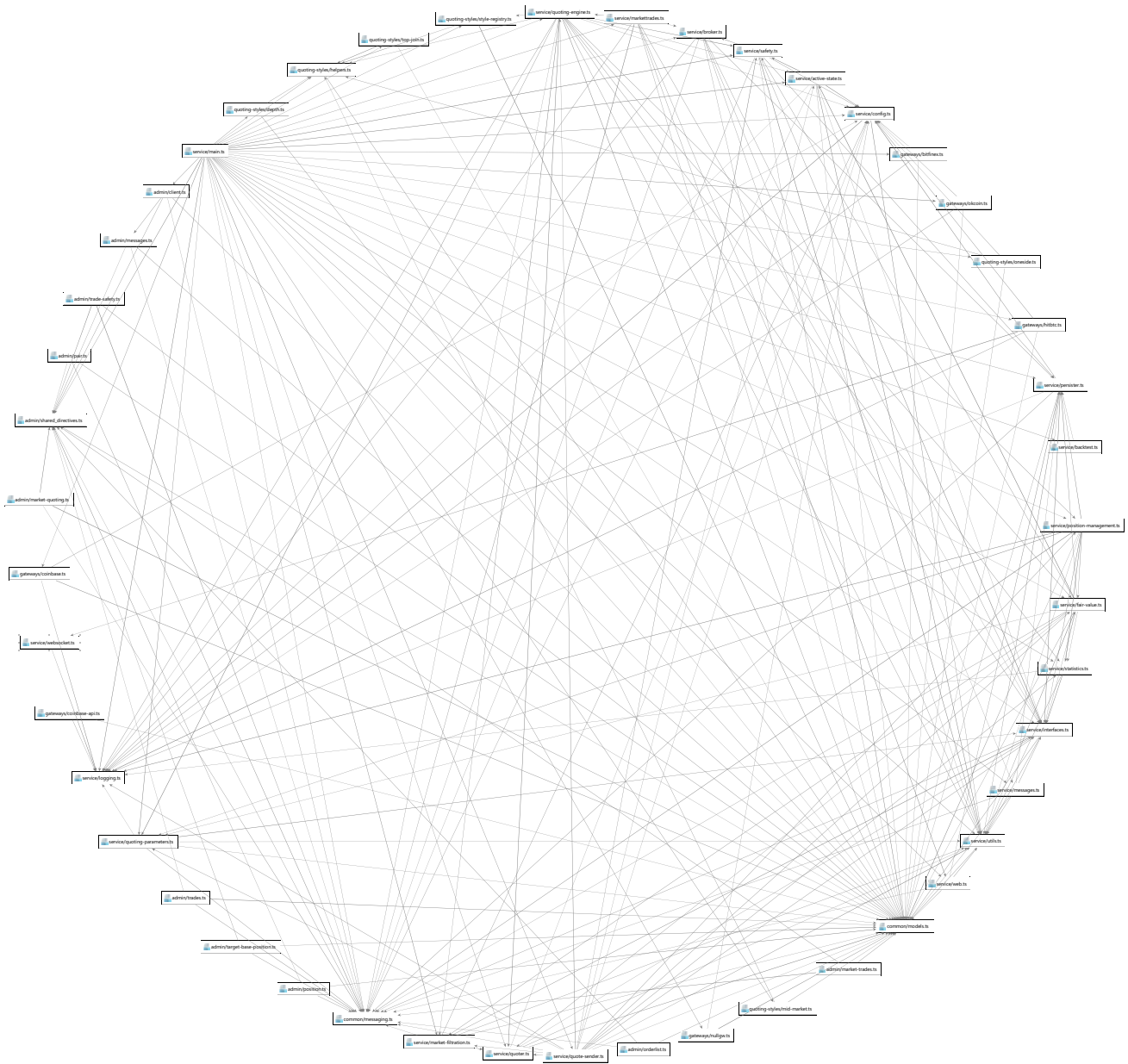
Figure 3.1: Network diagram of the program representing the files that are interacting. Elaborated by the author using JetBrains Webstorm.

Regarding the folder's structure, Cryptobot has all coding files inside `src` folder. In `src`, folders are structured in the following way:

- **admin**. This folder stores all the information related to the use of Browserify. Browserify is an open source tool that creates and manages modules in the client

side (the browser) using the same syntax as Nodejs. With this library it is possible to install modules of third parties or from NPM, in addition to private modules.

- **common**. `Common` contains `messaging.ts` and `models.ts`, both mentioned before. These files are separated from the rest because they are used commonly by most of the other files.

- **service**. In `service` all the files associated to the program's core are stored. Inside `service` there are two additional folders:

  - **gateways**. Contains the files that convert the information received and sent to every exchange into a common scheme understandable for Cryptobot. These files should be updated and modified every time an exchange API is updated. It is also possible to implement new exchanges through this folder making the appropriate changes in the linked files.

  - **quoting-styles**. This folder includes all the trading modes available in Cryptobot. `Top-join.ts` and `Depth.ts` contain the code responsible for the best bid and ask price estimation for the market making trading modes. `OneSide.ts` contains the code responsible for the trading mode OneSide. The rest of the files are auxiliary files.

- **static**. This folder stores the information related to the User Interface. Modifying `index.html` allows to change the data displayed through the UI.

3.3 User guide

Installation manual

To understand better the components involved in the program and to explain further configuration of the trading bot, it is best to install the program first. The installation steps are the following:

0. **System requirements**. A prerequisite for installing the program is to have a 64-bit computer. Installation has been tested on Windows 10, but it should work on a 64-bit system running Windows 7 or higher or macOS 10.8 or higher.

1. **Docker Toolbox installation**. Docker is used as a highly optimized virtual machine that puts libraries and dependencies in the same package using containers (images) without creating a whole virtual operating system. Docker allows many advantages such as an easier installation of programs, images are treated independently of the platform of execution, the applications are isolated so every container runs independently from the rest and the applications are centrally managed. Windows installation is available through https://docs.docker.com/toolbox/toolbox_install_windows/. Docker 1.7.1 or higher is required. During

installation process, uncheck "Kitematic for Windows" (visual container manager), since the program does not need it.

2. **Program folders**. Download, unzip and save folders Cryptobot1 and Cryptobot2 in an empty directory. Cryptobot1 and Cryptobot2 are the same program with different exchange configurations to allow connection of 2 exchanges at the same time. Cryptobot1 is connected by default to pair BTC/USD of GDAX and Cryptobot2 to pair BTC/USD of Bitfinex.

3. **Program configuration**. Environmental variables are configurable through the file `env` in $\sim/Cryptobot1$.

   - Variable `EXCHANGE` configures the exchange at which Cryptobot will be connected. The program has been updated with the latest API configuration from exchanges textttCoinbase (GDAX) and textttBitfinex.

   - Variable `TradedPair` configures the trading pair. If the connected exchange supports it, any of the following combination is possible: USD, BTC, LTC, EUR, GBP, CNY, ETH, DASH, DOGE, LSK, XMR. Example: LTC/EUR.

   - `MongoDbUrl` defines the name of the mongoDB database associated with the program. By default mongoDB database is called `mycryptobot1` for Cryptobot1 and `mycryptobot2` for Cryptobot2. If the program is running on Windows or macOS, "192.168.99.100" in `MongoDbUrl` should be changed to the output of $docker-machine\ ip$ from your docker terminal.

   - Variable `WebClientUsername` and `WebClientPassword` allow to set a username and password to access the UI through the browser. If kept as NULL, the web client will not require authentication.

   - Modify API key, secret and passphrase of your desired exchange to your personal configuration.

4. **Compilation**. Open Docker Quickstart Terminal and navigate to the folder Cryptobot1. Example: if it is saved in your Desktop run $cd\ Desktop/Cryptobot1$. Compile the program with $./build$. If the program is correctly compiled, running $./docker\ ps$ should present information about the cryptobot running containers.

5. **User Interface**. Once the program is compiled, you can access to the User Interface by visiting http://192.168.99.100:3001/ through your browser address bar. It should look something like this:
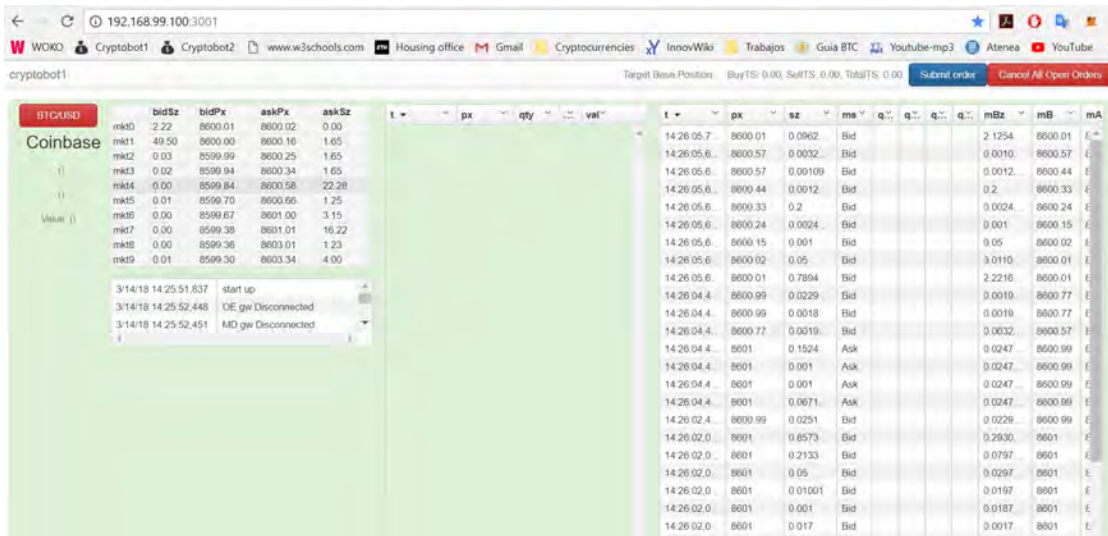
Figure 3.2: Cryptobot user interface

If you want to run in parallel Cryptobot2, go back to Step 3. UI for Cryptobot2 is available through `http://192.168.99.100:3002/`.

6. **Plotter**. A Matlab script has been programmed to plot data collected from both platforms. `Plotter.m` connects to mongoDB databases `mycryptobot1` and `mycryptobot2` and creates plots of the price difference between exchanges, arbitrage opportunities and spread evolution. If you are interested in plotting the data collected by both programs, you can download the script `Plotter.m` to your computer and run it through Matlab. `Plotter.m` only works when both programs are running. `Plotter.m` uses the add-on Database Interface for MongoDB. This Add-On can be installed through the Matlab tab Add-Ons>Get Add-Ons. "Database Interface for MongoDB" also needs Add-On Database Toolbox 8.0 for running. In the chapter "Results", a detailed description of the script will be made. After running the script different plots should appear:

39
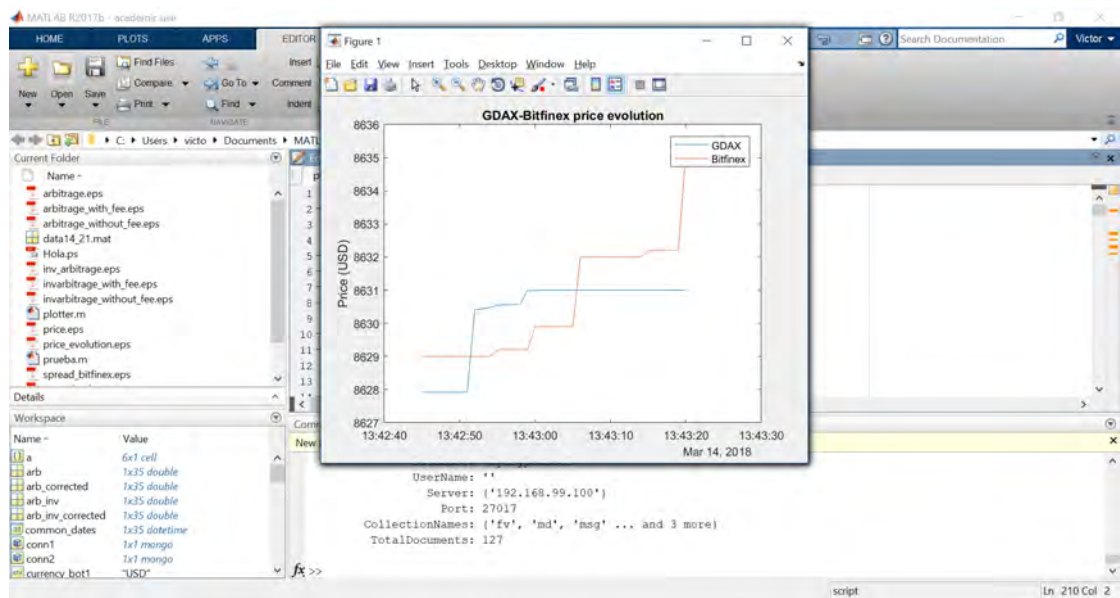
Figure 3.3: Plots created with the Matlab script

7. **Trading strategy**. Cryptobot1 and Cryptobot2 are configured by default to place an order, that should not be filled (limit order at 20.000 USD), when the exchange spread margin is greater than 5 cents of the pair base currency to which it is connected. In order to change this order placement configuration, a different trading strategy can be programmed in file `OneSide` stored in `Cryptobot1/src/service/quoting-styles`. From `OneSide` it is possible to gather information such as price and size of orders in the different levels of the order book as well as getting information from the exchange account such as current position of BTC. Finally, it is also possible to configure the type of orders Cryptobot is sending to the exchange.

8. **Close the application**. In order to stop the running containers Cryptobot1 and Cryptobot2 execute `$ ./stop` from Cryptobot1 or Cryptobot2 through the Docker Terminal. To close the terminal execute `$ exit`.

User Interface

Once Cryptobot1 is up and running, visit HTTP port 3001 (3002 for Cryptobot2) to access the UI (i.e. http://192.168.99.100:3001/). Through the User Interface plenty of data can be analyzed: order book, market trades, currency positions, exchange and pair connected, order list, trades done, quoting mode and parameters to trade. Figure 3.4 illustrates the information shown by the UI.
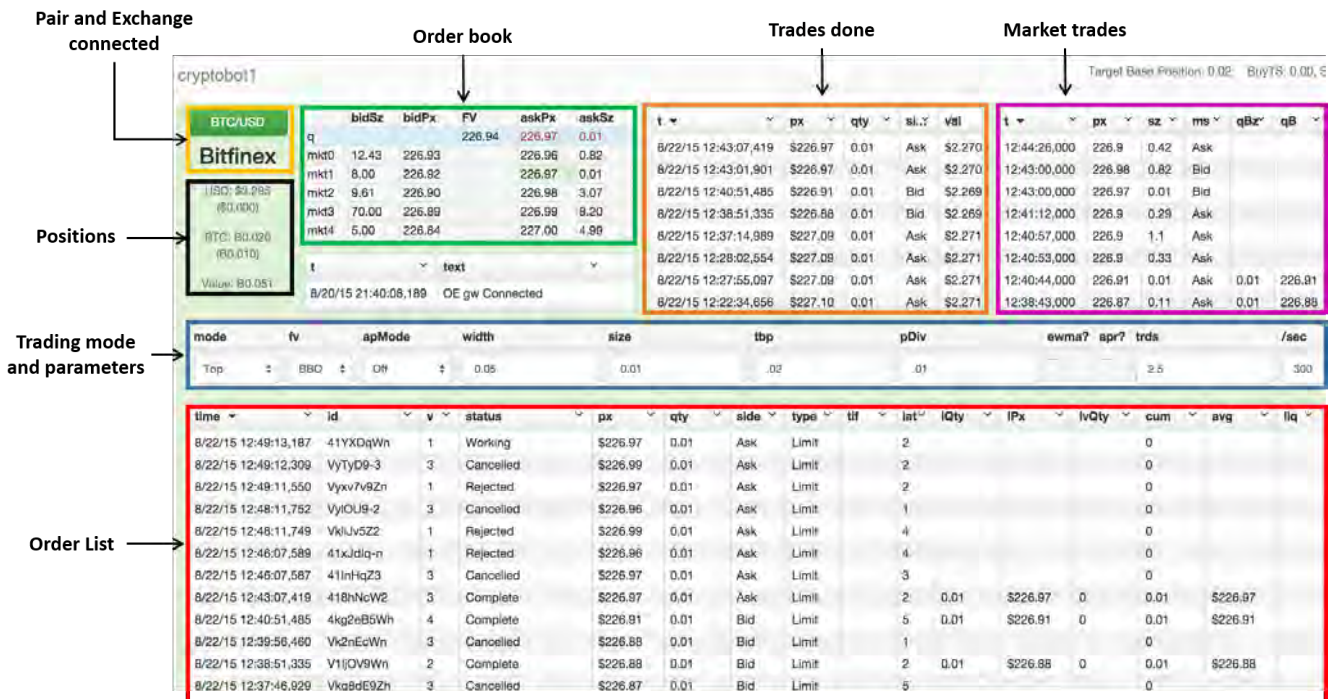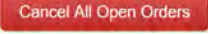
Figure 3.4: Market data information, pending orders and trades done shown by Cryptobot UI

- **Order book**. This is the order book from the exchange for the pair connected. The "mkt" rows are the best bids and offer levels on the exchange you are connected to.

- **Trades done**. Trades done by your exchange account. "side" is the side which the order was sent as. val is the total value of the trade, which is calculated as follows:
$$val = px \cdot qty \pm fee$$
Where px is the price of the order, qty is the size of the order and fee is the total exchange fee applied to the order.

- **Market Trades**. Trades done by all participants in the market. t is the time when the trade is made, px is the price of the order, sz is the size of the trade and ms is the market side of the trade. The columns starting with q are the program quotes at the time of the trade, the columns starting with m are the best bid and offer information at the time of the trade.

- **Trading modes and parameters**. All the trading modes available with the parameters to customize them.

- **Positions**. Shows the amount of currency of the pair connected.

- **Order List**. Shows order statuses of each order sent to the exchange. Cxl button ❌ will attempt to cancel the order.

41

Through the User Interface there is also the possibility of sending single orders manually. When pressing the Submit Order button Submit order a pop-up menu will appear with different order sending options such as side, price, size, time in force and type of the order. In addition, there is also the option of canceling all the existing open orders opened by the program and registered by the UI. Button Cancel All Open Orders Cancel All Open Orders performs this function.

Trading modes

As mentioned before, Cryptobot is based on the existing project Tribeca. In order to not lose the market making trading possibilities offered by Tribeca, Cryptobot includes a new independent trading mode called OneSide. OneSide is designed to be a simple trading mode where the user can define his own trading strategy and send simple orders to the exchange based on recent market data information. OneSide is the default trading mode, but it can be changed through the UI to another trading mode:

Figure 3.5: Selecting the trading mode

Only one trading mode can run at the same time at a running container. OneSide is the trading mode incorporated to the existing modes in Tribeca. In this mode, single orders can be sent based on a defined strategy using recent market data information. If OneSide is selected, the program will automatically execute the code written in `Cryptobot1/src/service/quoting-styles/OneSide.ts`. A description of every market making trading mode with their respective parameters is included in the Appendix. If a market making trading mode is selected, the button showing the connected market pair should be pressed to turn it into green and consequently start making markets. The transition of the button is represented below:

Figure 3.6: Market pair button transition

MongoDB database

As mentioned before, Cryptobot creates a mongoDB database of wealth of data gathered via REST API including market data information and messaging register. The most important collections of the created database are listed below:

- **md**. This collection stores the market data information of the order book. It is separated in 6 different fields: x_id is a unique identifier of each entry, bids are the best 3 bids at the time of the registry, asks are the 3 best asks at the time of the registry, time, exchange is a number that identifies the exchange and pair gives information of the market pair.

- **mt**. This collection stores the market trades done by all the participants in the market. It is separated in 6 different fields: x_id, exchange, pair, price, size, time, quote, bid, ask and make_side. bid and ask are the best bid and ask at the time of the trade. make_side is a binary that represents the side of the trade (Ask or Bid).

- **msg**. This collection stores the connection registry of the trading bot. Field text stores the information related to the connection and disconnection of the program to the network.

Speed

The speed at which Cryptobot is gathering data and placing orders on the order book depends on many factors:

- The network status on the exchange side.

- The exchange connected.

- The network status on the client side.

- The computer processing capacity.

- The complexity of the trading strategy.

Some speed tests are done using the following computer and connectivity characteristics:

- CPU: Intel Core i7 7200U Processor.

- RAM: 8GB.

- Operating system: Windows 10.

- Connectivity: Network cable connected to the ETH network.

In the case of GDAX, the speed at which Cryptobot could place an order on the order book ranged from 0,4 to 0,6 seconds depending on the moment of the connection. On the other hand, the order placement speed in Bitfinex was a bit slower, around 1 second for every order placed.

Trading opportunities

In order to test the performance of the implemented program, an analysis of trading opportunities using the data gathered by the software will be conducted. Taking advantage of the MongoDB database creation done by Cryptobot, a Matlab script called `Plotter` will be designed to plot the data in a useful manner.

The script

In order to use the stored data in `mycryptobot1` and `mycryptobot2` (MongoDB databases created by Cryptobot1 and Cryptobot2), the designed Matlab program uses an Add-on called "Database Toolbox Interface for MongoDB", which allows a connection between Matlab and MongoDB. The script connects to MongoDB through this Add-On using the following parameters:

- **server**. This is the virtual machine host IP address. By default should be "192.168.99.100". The IP can be consulted through the Docker terminal by executing `$ docker-machine ip`.

- **port**. This is the mongo container port used in Cryptobot that is binded to the same port number on the VM's host IP.

- **dbname1 and dbname2**. These are the mongo database names configured in Cryptobot.

By running Cryptobot1 and Cryptobot2 during the same period, the output of the script will be different plots showing: the exchange rate evolution of each pair connected to each bot, the arbitrage opportunities between market pairs and the evolution of the spread on each market. Figures have been created with the following bot parameter configuration:

Other datasets with different exchange and pair configurations are represented in Appendix.

	Cryptobot1	Cryptobot2
Exchange	GDAX	Bitfinex
Market pair	BTC/USD	BTC/USD

Table 4.1: Exchange and pair configuration.

Exchange rate evolution

Figure 4.1 has been generated using the collection `mt`(market trades) and represents the evolution of the market price in each exchange during the analyzed period.
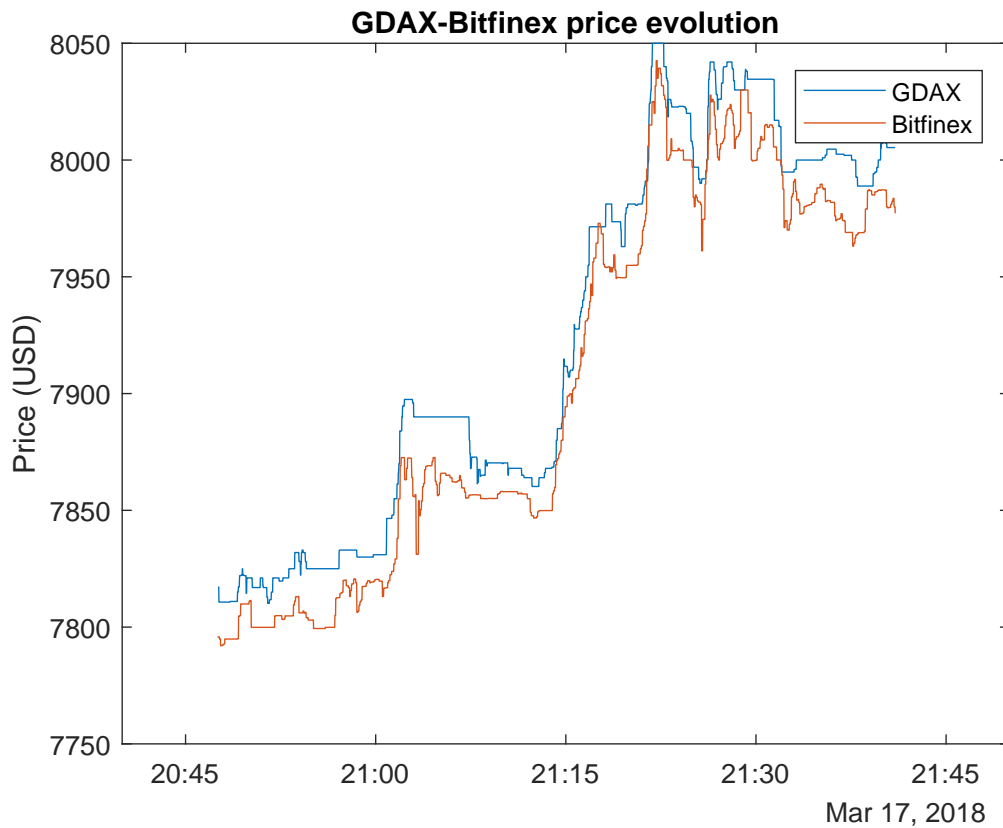

Figure 4.1: BTC/USD comparative rate evolution in GDAX and Bitfinex

Arbitrage opportunities between exchanges

Arbitrage is a trading strategy that consists in taking advantage of the price difference between two or more markets, combining complementary offers (buy and sell) at the same time to capitalize the imbalance, where the profit should be the difference between market prices. This strategy takes advantage of this difference to make a profit free of risk (in theory).

In practice, this operation has some associated risks, such as price fluctuations during the operation (since it is practically impossible to close two or more operations at the same instant) or if there is no counterparty to fulfill one side of a transaction.

Figure 4.2 represents the possible profit of applying this method between GDAX and Bitfinex markets. The represented profit would be obtained by buying 1 Bitcoin in Bitfinex and selling 1 Bitcoin in GDAX. Blue line represent the profit obtained by executing limit orders at the best bid available in Bitfinex and the best ask available in GDAX. Orange line is the same situation but considering the respective maker fee of each exchange. Black line represents the profit obtained by executing market orders on both exchanges considering the respective taker fee of each exchange. In this case, the orders would be executed at the best ask price in the case of Bitfinex and at the best bid price in the case of GDAX, both orders removing liquidity from the exchange.
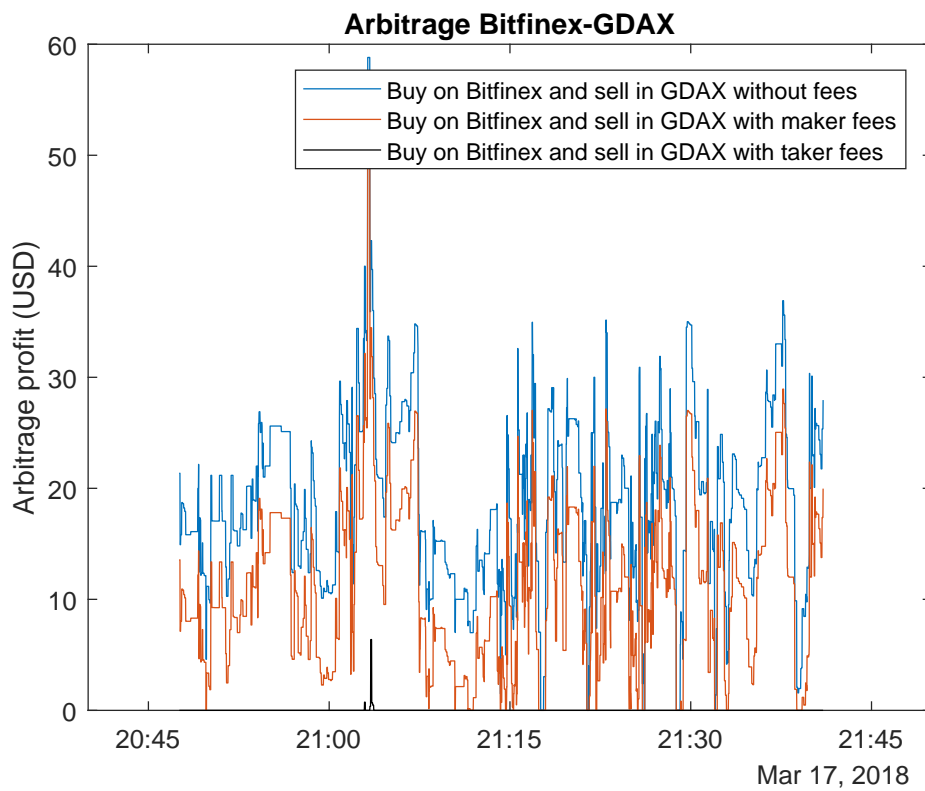
Figure 4.2: Arbitrage opportunities between GDAX and Bitfinex. The represented profit would be obtained by buying 1 Bitcoin in GDAX and selling 1 Bitcoin in Bitfinex simultaneously with or without considering fees.

Figure 4.3 represents the same possibility in case of an inverse execution; buying in Bitfinex and selling in GDAX.
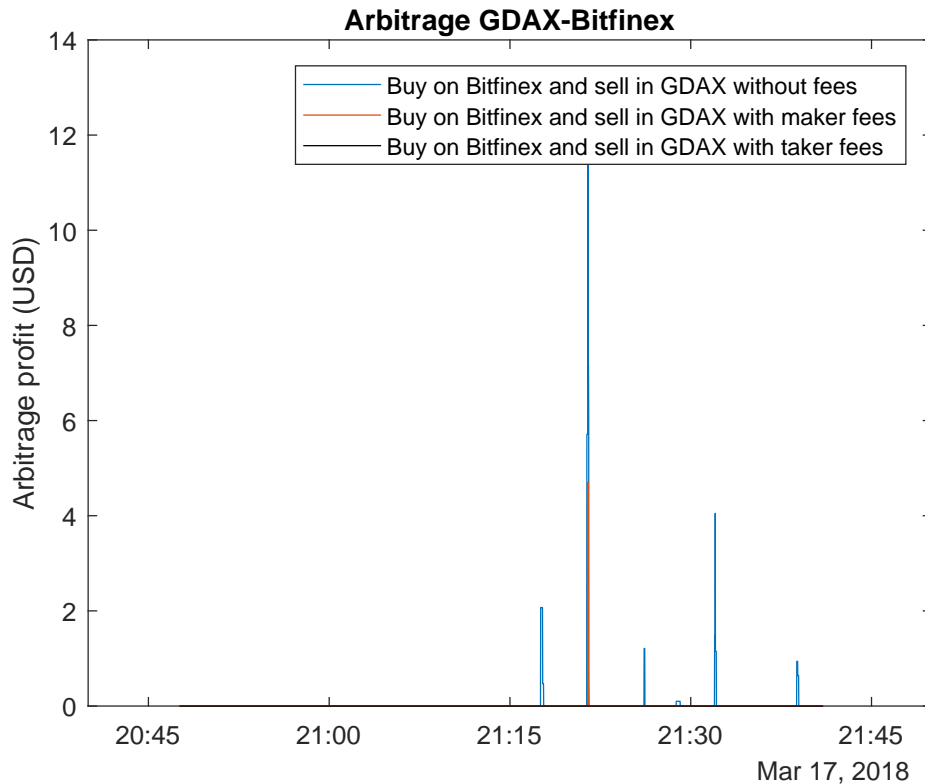
Figure 4.3: Arbitrage opportunities between GDAX and Bitfinex. The represented profit would be obtained by buying 1 Bitcoin in Bitfinex and selling 1 Bitcoin in GDAX simultaneously with or without considering fees.

As can be noted, there are more trading opportunities in the first case, mostly because the price of GDAX was higher than the price of Bitfinex for the analyzed period.

The possible ways to execute this type of trading strategy (buy and sell in different exchanges) indefinitely are the followings:

- Buy 1 Bitcoin in one exchange (assuming that there is already money to exchange), transfer the Bitcoin to the other exchange and sell it. This is the easiest way to perform the strategy. The main problem of this procedure is that in most cases, the cryptocurrency network needs several minutes to complete the transaction. In this case, it could happen that, during this period, the price on the second exchange drops bellow the purchase price, causing a loss.

- Already have some Bitcoins and money in both accounts. By doing so, it would be possible to buy and sell 1 Bitcoin at the same time. It is important to notice that after this type of execution, account positions should be rebalanced to go back to the initial state. In this case, deposit and withdrawal fees should be considered. In the previous case (Figure 4.2 and Figure 4.3), the situations where there was

a possible profit (around 21:05 for Bitfinex-GDAX), would have been a loss after applying these fees.

- Doing the same approach as in the previous case but, in this case, avoid to rebalance the accounts by doing an inverse execution when possible. That is, wait until a new opportunity appears in the other direction: where we sold now buy and where we bought now sell.

- Take advantage of the possibility to short sell offered by some exchanges (e.g. Bitfinex). In this situation, it would be possible to short sell in the exchange that offers the lowest price and buy in the exchange that offers the highest price. When the spread between both exchanges closes (same price), we should exit the market by buying back in the first exchange and selling in the second exchange. In this case, margin funding fees should be considered.

Bid ask spread

As mentioned before, market makers can take advantage of the spread between the best bid price and the best ask price of a market. Figure 4.4 and Figure 4.5 represent the spread evolution (difference between best ask and best bid price) compared to the best ask price for GDAX and Bitfinex.
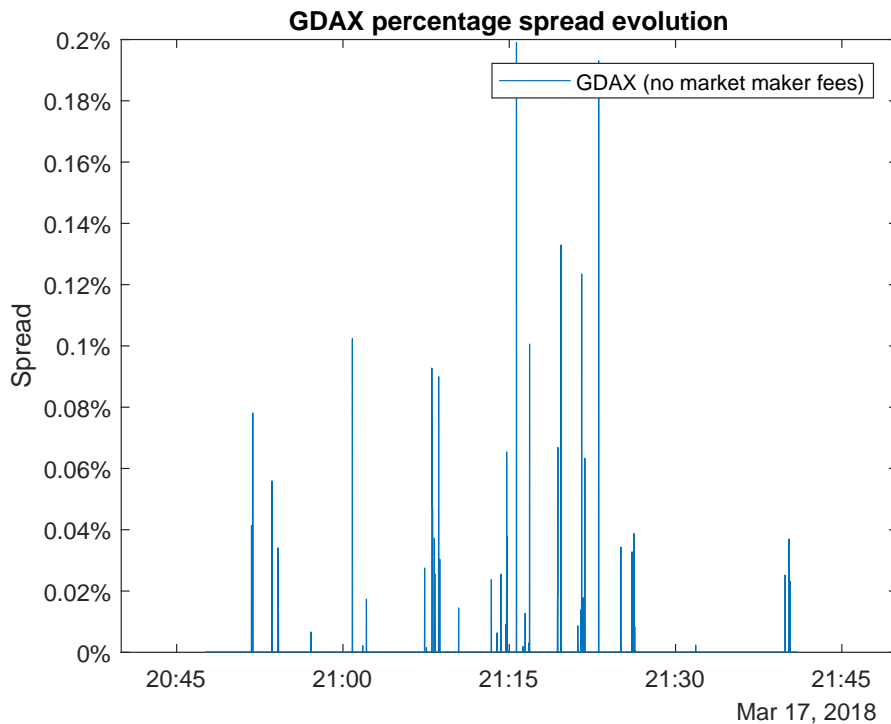

Figure 4.4: Representation of the spread evolution in GDAX compared to the best ask (lowest quoted offer price) for BTC/USD market pair.
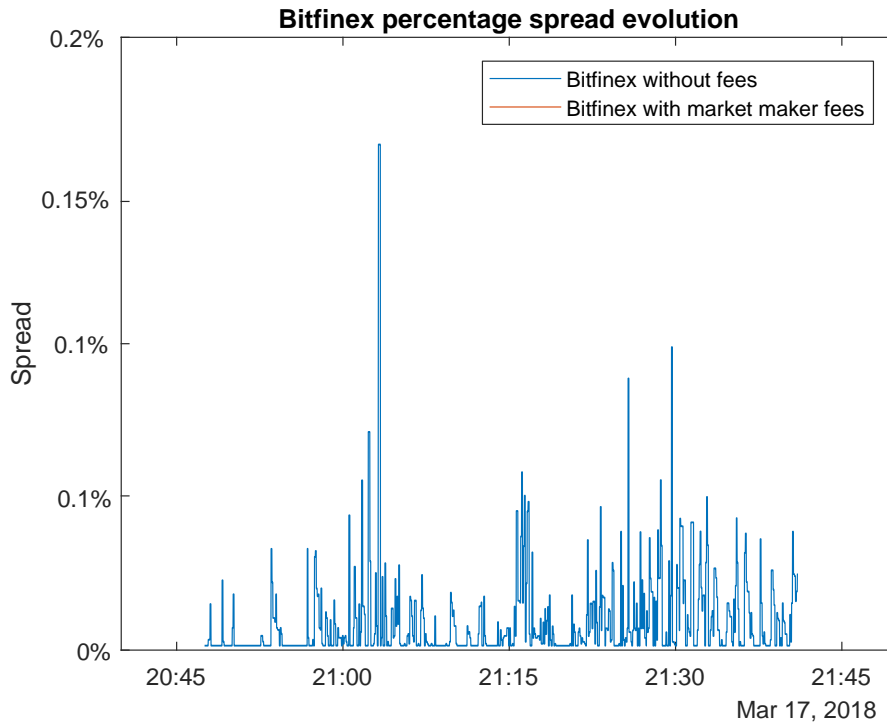
Figure 4.5: Representation of the spread evolution in Bitfinex compared to the best ask (lowest quoted offer price) for BTC/USD market pair.

Results

After analyzing the three different situations (BTC/EUR and LTC/USD represented in Appendix), it can be seen that there are trading opportunities for both strategies. It is important to notice that the window opportunity available for a successfully trading execution is very small, in the order of few seconds.

In the case of arbitrage, the window time is open from tenths of seconds up to 1 minute, depending on the exchange, the moment of the analyzed period and the market pair studied. In the case of the bid ask spread, the window opportunity ranges from tenths of second up to 10 seconds.

Because of this reason, it is very important to exploit these opportunities using a software that reacts to the market on real time. In the case of performing these strategies by placing limit orders, it is crucial to place the orders as fast as possible trying to be on top of the order book at the time of the new opportunity.

If the orders are placed as market orders, it is also important to react as fast as possible in order to get the best bid and ask of the order book. If the program reacts slowly, by gathering data or placing orders, a profit opportunity can be converted into

a loss opportunity. When placing market orders, it is also important to point out that the possible profit depends on the size of candidate orders in the order book at the time of the trading execution.

Conclusion

Over the last years, cryptocurrencies have become more and more popular due to their disruptive innovation and, most importantly, because of their meteoric rising value. The number of cryptocurrencies and exchanges has been growing enormously in the last years. There is an increasing interest for this technologies that requires a continuos monitoring of its evolution.

In this sense, Cryptobot is designed to help address this need by automating a permanent follow-up of the cryptocurrencies. The software is able to connect to different APIs in order to gather market data information as well as placing orders in different exchanges. The software is designed to be easily modified to implement different trading strategies in addition to the possibility of adopting new cryptocurrency exchanges and market pairs.

Furthermore, the analysis and comparison of most important cryptocurrencies and exchanges provide a detailed overview of the market, in order to make informed decisions on the most appropriate exchange and cryptocurrency that could be implemented in future.

It is important to emphasize that the actual markets are still immature to be suitable for institutional investors in terms of stability and liquidity. The natural interplay between regulation, security, anonymity and availability and the distributed nature of cryptocurrencies creates persistent market inefficiencies and high price volatility in the cryptocurrency markets. The constant creation of new exchanges, with their respective staggering competition, and the implementation of new market pairs on each exchange, also contributes to this market inefficiency.

For the very same reason, lots of predictable trading opportunities for small investors emerge due to these market inefficiencies. In this sense, this project offers a great advantage by presenting a new instrument to take profit from these opportunities.

Because of the short duration of the project, it has not been possible to take fully advantage of the program and analysis made. Some of the possible improvement areas are listed below:

- Although the program fulfills the objectives of the project, it could be improved in terms of speed. As mentioned before, in order to perform successfully trading strategies such as arbitrage or market making, it is crucial to gather data and place orders almost instantaneously.

- Take advantage of the analysis made between cryptocurrencies and exchanges. The most appropriate exchanges with their respective digital coins, based on the user's criteria, should be implemented in the progam.

- Perform a more exhaustive study of the trading strategies and opportunities. Again, benefiting from the analysis and comparison made, a more extensive study of trading opportunities could be performed. For instance, making an estimation of the real arbitrage profit considering the different possibilities of position re-balance. In addition, other trading strategies could be implemented such as the log-periodic power law singularity (LPPLS) model or with novel trading strategies as the Bayesian Regression method.

Appendices

A.1 Appendix

Market making trading modes

Below are listed the different market making trading parameters[1]:

- **Mode**. Sets the quoting mode
  - **Join**. Sets a quote to be at the best bid and the best offered price, if the Best-Bid-Offer (BBO) is narrower than $width$, set the bid quote at $FV - width/2$ and ask quote at $FV + width/2$.
  - **Top**. Top - Same as Join, but if the code can better the best bid or offer by a penny while respecting the $width$, set that as the quote so we will then be at the top of the market.
  - **Mid**. Set the bid quote at $FV - width/2$ and ask quote at $FV + width/2$.
  - **Inverse Join**. Set the quote at the BBO if the BBO is narrower than $width$, otherwise make the quote so wide that no one will trade with it.
  - **Inverse Top**. Same as Inverse Join but make our orders jump to the very top of the order book.
  - **PingPong**. Same as Top but always respect the calculated $width$ from the last sold or bought size.
- **FV**. Sets the fair value calculation mode.
  - $BBO - FV = ([bestbidprice] + [bestaskprice])/2.0$
  - $wBBO - FV = ([bestbidprice] \cdot [bestasksize] + [bestaskprice] \cdot [bestbidsize])/([bestasksize] + [bestbidsize])$
- **apMode**.

---

[1]Source: https://github.com/michaelgrosner/tribeca/wiki.

– **Off**. Tribeca will not try to automatically manage positions.

– **EwmaBasic**. Tribeca will use a 200 minute and 100 minute exponential weighted moving average calculation to buy up BTC when the 100 minute line crosses over the 200 minute line, and sell BTC when the reverse happens. The values of 100mins and 200mins are currently not exposed in the UI, but are represented in the code as shortEwma and longEwma.

- **Width**. Minimum width of the quote in USD (ex. a value of .3 is 30 cents). With the exception for when *apr* is checked and the system is aggressively rebalancing positions after they get out of whack, width shall never be violated.

- **size**. Maximum size of our quote in BTC (ex. a value of 1.5 is 1.5 bitcoins). With the exception for when apr is checked and the system is aggressively rebalancing positions after they get out of whack, size shall never be violated.

- **tbp**. Only used when *apMode* is Off. Sets a static "Target Base Position" for Tribeca to stay near. In off auto-position mode, Tribeca will still try to respect *pDiv* and not make your position fluctuate by more than that value. Example: with 10 BTC to trade, and setting $tbp = 3$, $apMode = $ Off and $pDiv = 1$, the holding of BTC will never be less than 2 or greater than 4.

- **pDiv**. If the "Target Base Position" diverges more from this value, Tribeca will stop sending orders to stop too much directional trading. Example: with 10 BTC to trade, "Target Base Position" is reporting 5, and *pDiv* is set to 3, your holding of BTC will never be less than 2 or greater than 8.

- **ewma?**. Use a 100 minute EWMA smoothed line of the price.

- **apr?**. If Tribeca is in a state where has stopped sending orders because the position has diverged too far from Target Base Position, this setting will much more aggressively try to fix that discrepancy by placing orders much larger than size and at prices much more aggressive than width normally allows. It's a bit risky to use this setting.

- **trds and /sec**. Often, only buying or selling many times in a short timeframe indicates that there is going to be a price swing. *trds* and */sec* are highly related: If more than *trds* buy trades in /sec seconds are done, Tribeca will stop sending more buy orders until either */sec* seconds has passed, or enough is sold at a higher cost to make all those buy orders profitable. The number of trades is reported by side in the UI; "BuyTS", "SellTS", and "TotTS". If "BuyTS" goes above trds, Tribeca will stop sending buy orders, and the same for sells.

Trading opportunities with different datasets

The following figures have been created with the following bot parameter configuration:

	Cryptobot1	Cryptobot2
Exchange	GDAX	Bitfinex
Market pair	BTC/EUR	BTC/EUR

Table A.1: Exchange and pair configuration.
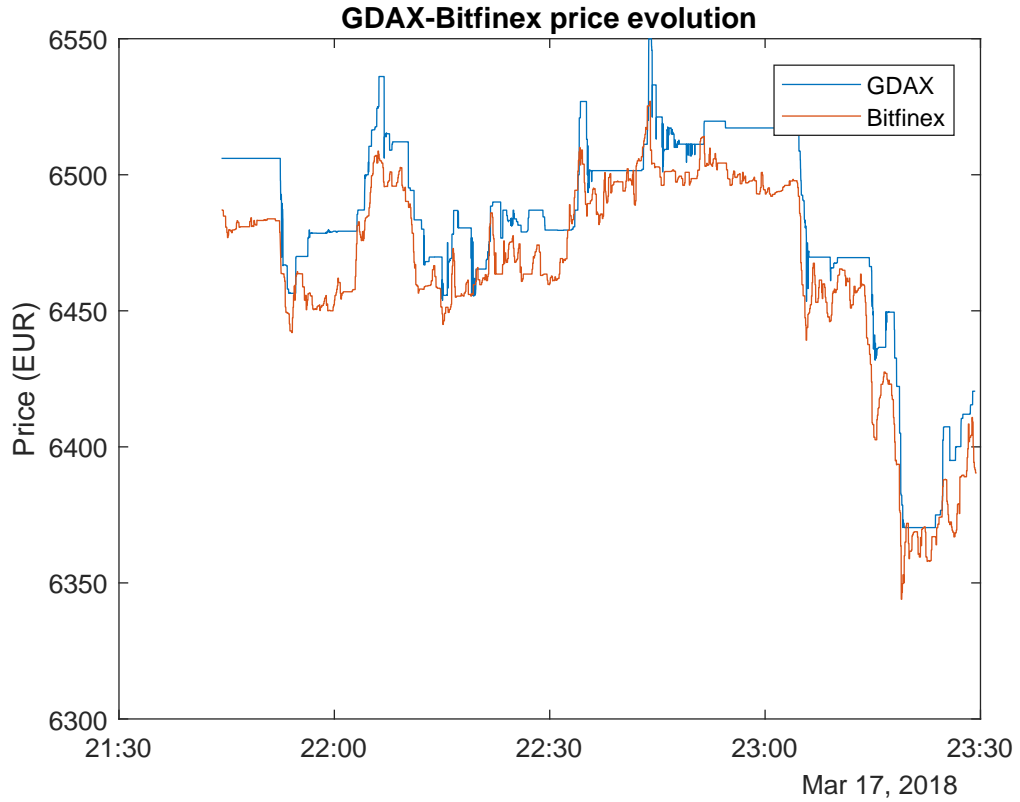
Exchange rate evolution

Figure A.1: BTC/EUR comparative rate evolution in GDAX and Bitfinex
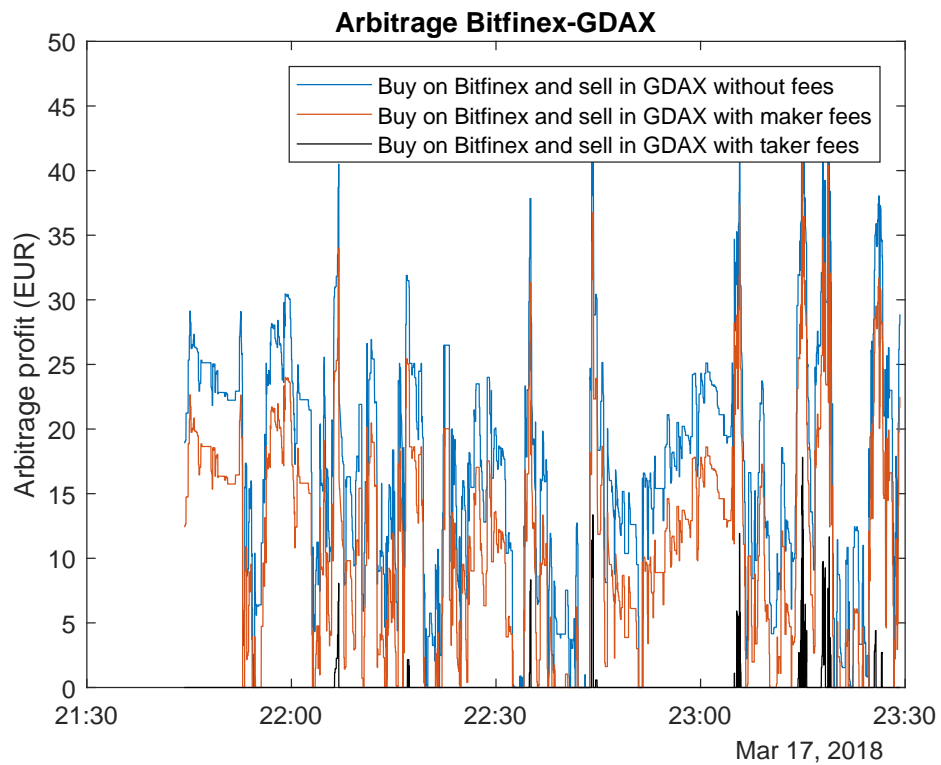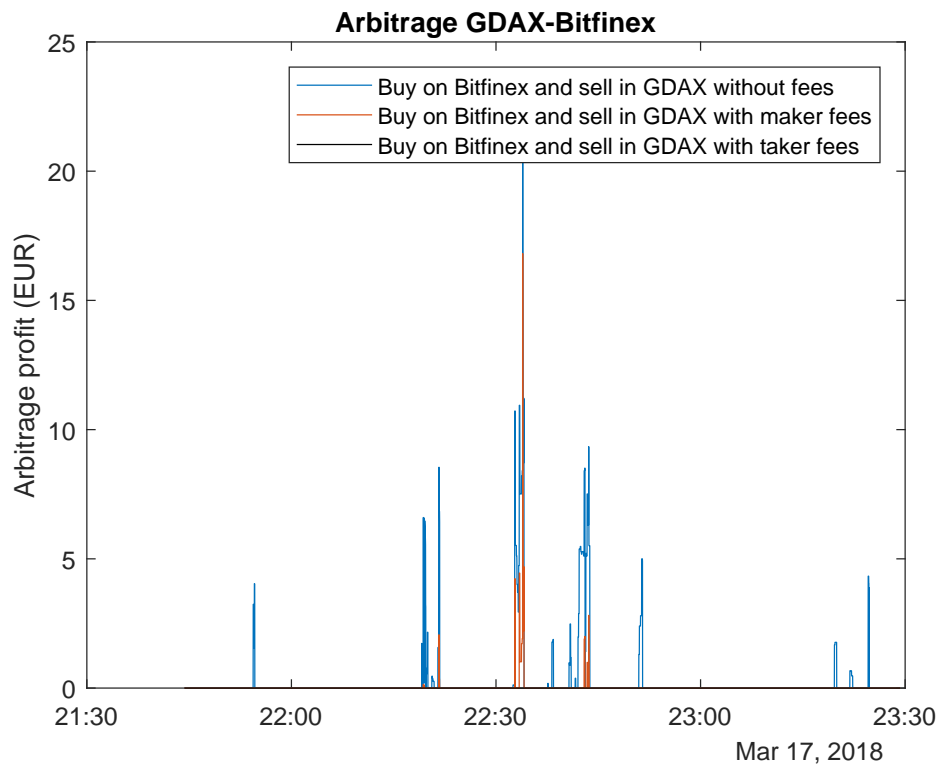
Arbitrage opportunities between exchanges

Figure A.2: Arbitrage opportunities between GDAX and Bitfinex. The represented profit would be obtained by buying 1 Bitcoin in GDAX and selling 1 Bitcoin in Bitfinex simultaneously with or without considering fees.
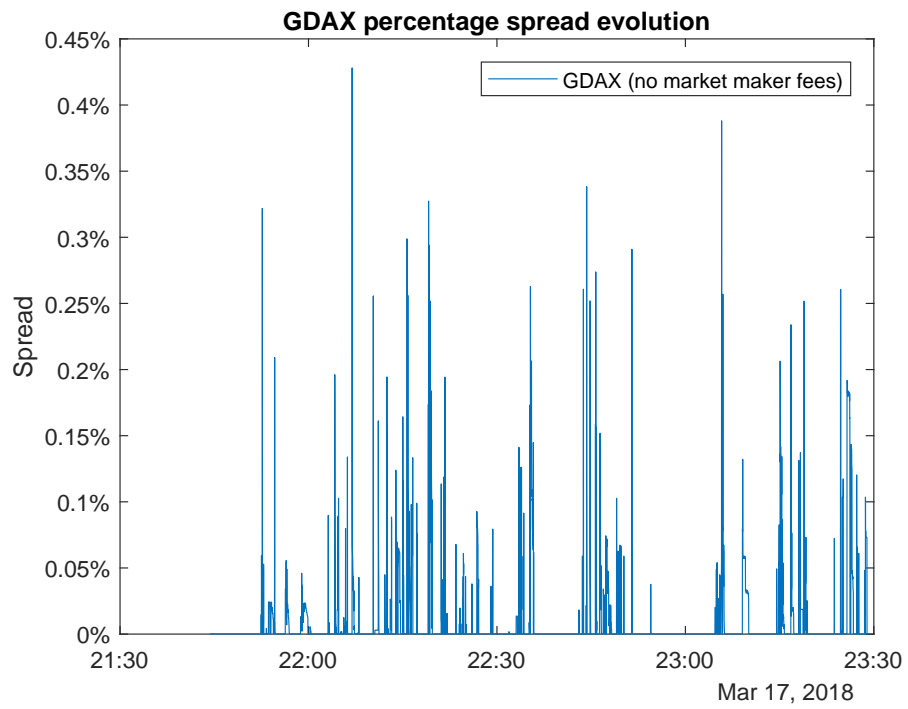
Figure A.3 represents the same possibility in case of an inverse execution; buying in Bitfinex and selling in GDAX.

Figure A.3: Arbitrage opportunities between GDAX and Bitfinex. The represented profit would be obtained by buying 1 Bitcoin in Bitfinex and selling 1 Bitcoin in GDAX simultaneously with or without considering fees.

Bid ask spread

Figure A.4: Representation of the spread evolution in GDAX compared to the best ask (lowest quoted offer price) for BTC/EUR market pair.
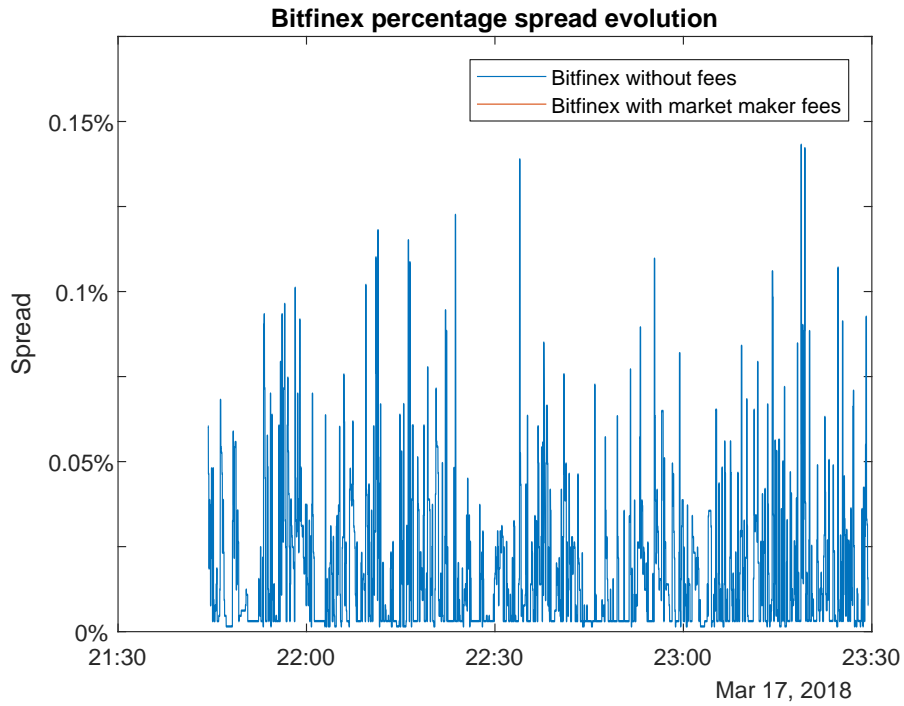
Figure A.5: Representation of the spread evolution in Bitfinex compared to the best ask (lowest quoted offer price) for BTC/EUR market pair.

The following figures have been created with the following bot parameter configuration:

	Cryptobot1	Cryptobot2
Exchange	GDAX	Bitfinex
Market pair	LTC/USD	LTC/USD

Table A.2: Exchange and pair configuration.

Exchange rate evolution
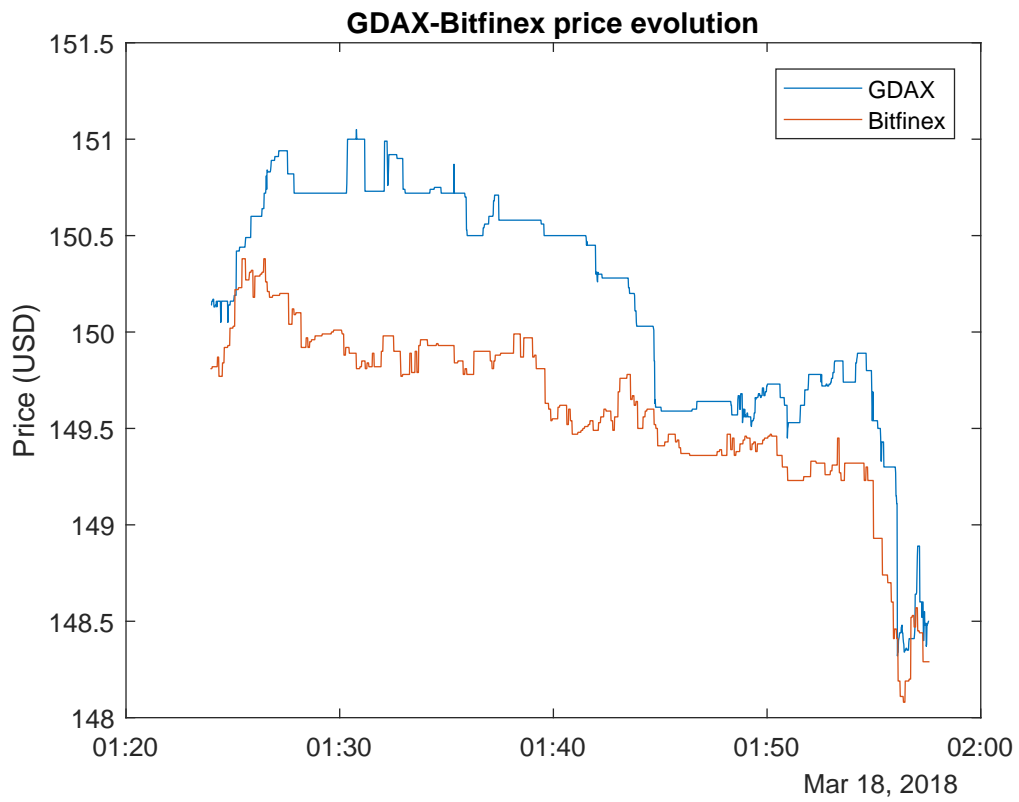
Figure A.6: LTC/USD comparative rate evolution in GDAX and Bitfinex

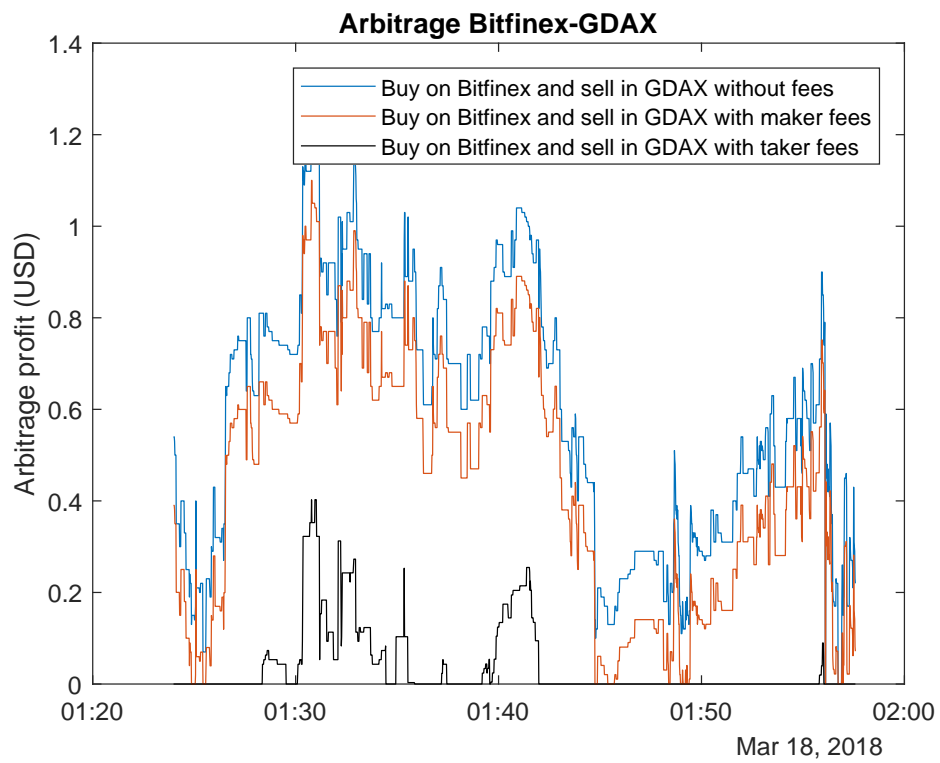**Arbitrage opportunities between exchanges**

Figure A.7: Arbitrage opportunities between GDAX and Bitfinex. The represented profit would be obtained by buying 1 Litecoin in GDAX and selling 1 Litecoin in Bitfinex simultaneously with or without considering fees.

Figure A.8 represents the same possibility in case of an inverse execution; buying in Bitfinex and selling in GDAX.
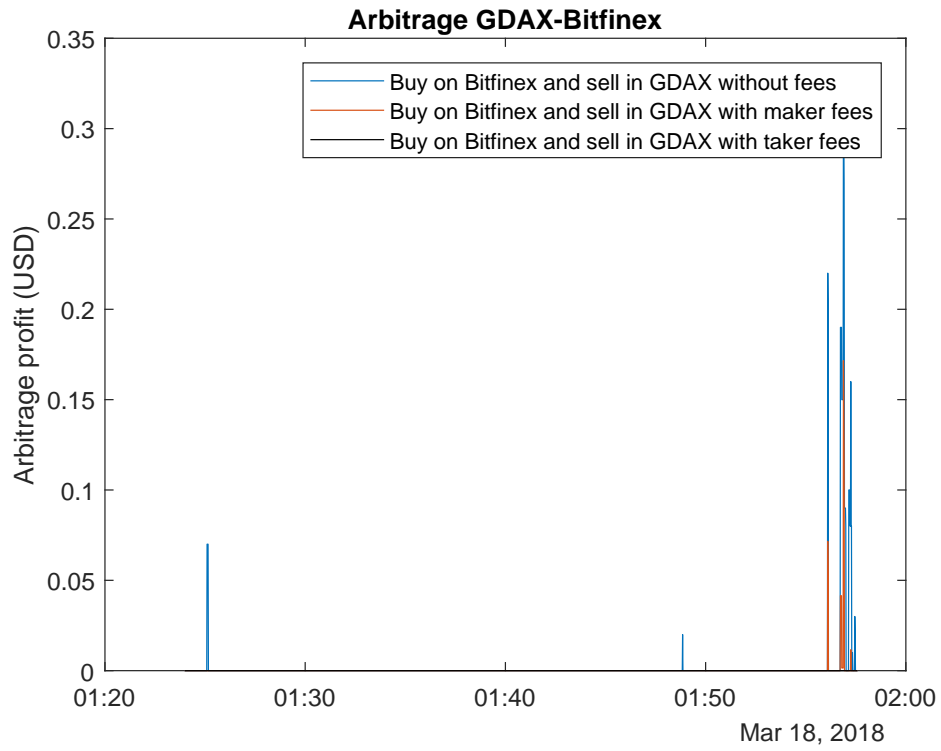
Figure A.8: Arbitrage opportunities between GDAX and Bitfinex. The represented profit would be obtained by buying 1 Litecoin in Bitfinex and selling 1 Litecoin in GDAX simultaneously with or without considering fees.
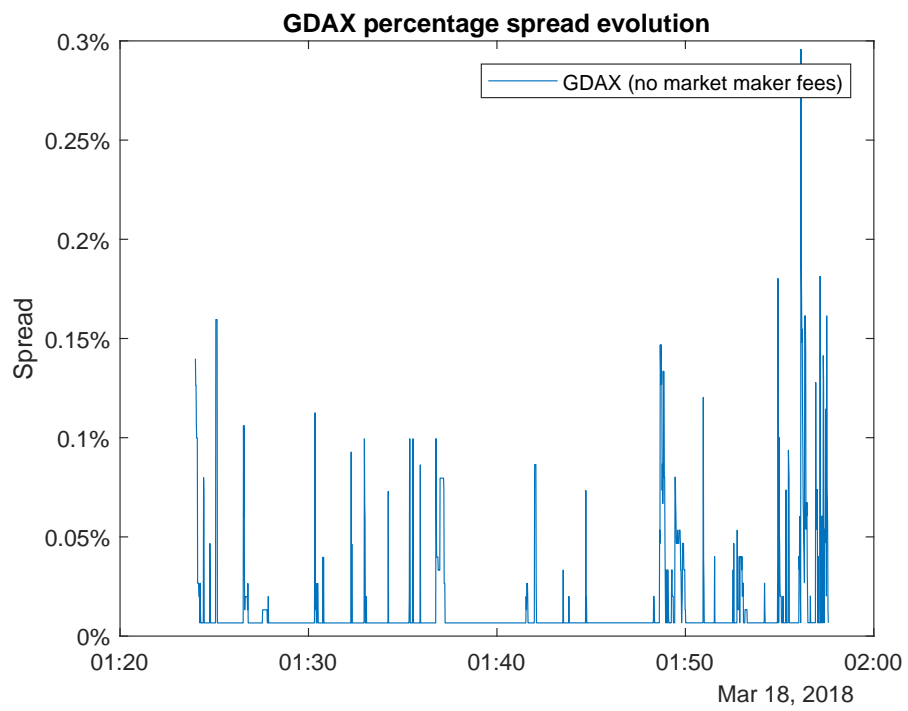
Bid ask spread

Figure A.9: Representation of the spread evolution in GDAX compared to the best ask (lowest quoted offer price) for LTC/USD market pair.
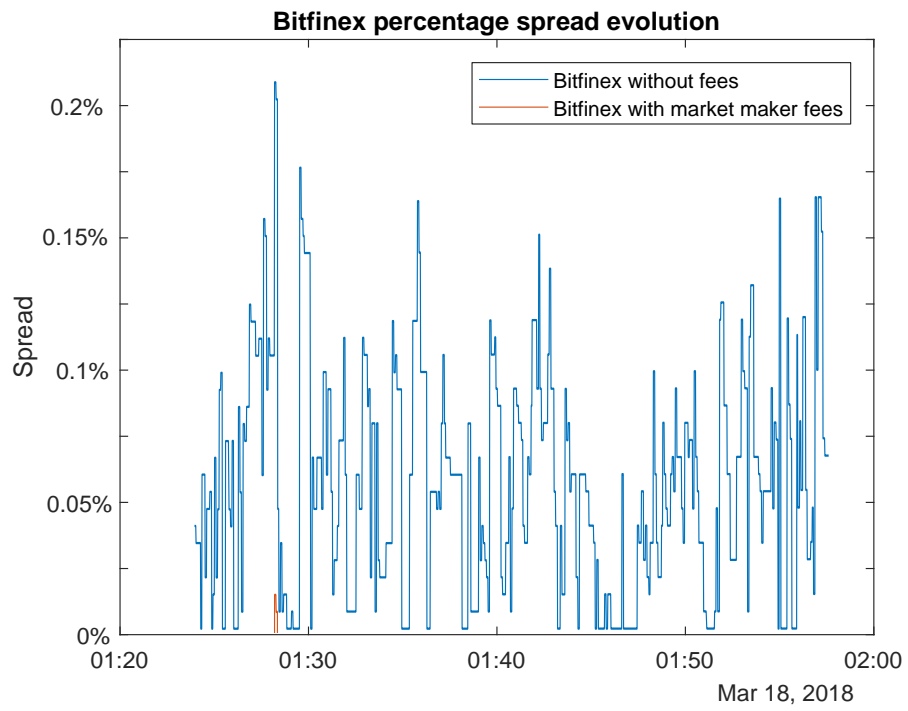
Figure A.10: Representation of the spread evolution in Bitfinex compared to the best ask (lowest quoted offer price) for LTC/USD market pair.

Bibliography

[1] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction.* Princeton University Press, 2016.

[2] Usman W Chohan. Cryptocurrencies: A brief thematic review. 2017.

[3] Don Tapscott and Alex Tapscott. *Blockchain Revolution: How the technology behind Bitcoin is changing money, business, and the world.* Penguin, 2016.

[4] Joshua A Kroll, Ian C Davey, and Edward W Felten. The economics of bitcoin mining, or bitcoin in the presence of adversaries. In *Proceedings of WEIS*, volume 2013, 2013.

[5] Bruce Schneier. Cryptanalysis of md5 and sha: Time for a new standard. *Computer World,(August, 2004)*, 2004.

[6] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008, 2012.

[7] Karl J O'Dwyer and David Malone. Bitcoin mining and its energy footprint. 2014.

[8] Iddo Bentov, Charles Lee, Alex Mizrahi, and Meni Rosenfeld. Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y. *ACM SIGMETRICS Performance Evaluation Review*, 42(3):34–37, 2014.

[9] Jae Kwon. Tendermint: Consensus without mining. *Retrieved May*, 18:2017, 2014.

[10] Elli Androulaki, Ghassan O Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating user privacy in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 34–51. Springer, 2013.

[11] Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In *International Workshop on Fast Software Encryption*, pages 371–388. Springer, 2004.

[12] Jerry Brito and Andrea Castillo. *Bitcoin: A primer for policymakers.* Mercatus Center at George Mason University, 2013.

[13] Andreas M Antonopoulos. *Mastering Bitcoin: unlocking digital cryptocurrencies.* " O'Reilly Media, Inc.", 2014.

[14] Eric Wall and Gustaf Malm. Using blockchain technology and smart contracts to create a distributed securities depository. 2016.

[15] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151:1–32, 2014.

[16] Jega Anish Dev. Bitcoin mining acceleration and performance quantification. In *Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on*, pages 1–6. IEEE, 2014.

[17] David Schwartz, Noah Youngs, Arthur Britto, et al. The ripple protocol consensus algorithm. *Ripple Labs Inc White Paper*, 5, 2014.

[18] Frederik Armknecht, Ghassan O Karame, Avikarsha Mandal, Franck Youssef, and Erik Zenner. Ripple: Overview and outlook. In *International Conference on Trust and Trustworthy Computing*, pages 163–180. Springer, 2015.

[19] Leonard Richardson and Sam Ruby. *RESTful web services.* " O'Reilly Media, Inc.", 2008.

[20] Ian Fette. The websocket protocol. 2011.

[21] Dierk Reuter and Eddie Wen. Foreign exchange trading system, January 22 2008. US Patent 7,321,873.

[22] Jay Palmer Fawcett. *Bitcoin regulations and investigations: A proposal for US policies.* PhD thesis, Utica College, 2017.

[23] Adrian Gallagher. A cryptocurrency trading bot and framework supporting multiple exchanges written in golang, 2015. URL https://github.com/thrasher-/gocryptotrader.

[24] Julien Hamilton. Blackbird bitcoin arbitrage: a long/short market-neutral strategy, 2015. URL https://github.com/butor/blackbird.

[25] Igor Kroitor. A javascript / python / php cryptocurrency trading library with support for more than 90 bitcoin/altcoin exchanges, 2017. Accessed on 7 March 2018. URL https://github.com/ccxt/ccxt.

[26] Gavin Chan. Cryptocurrency exchange market data feed handler, 2016. Accessed on 7 March 2018. URL https://github.com/Aurora-Team/BitcoinExchangeFH.

[27] Nils Diefenbach. Crypto-currency exchange api framework, 2016. Accessed on 7 March 2018. URL https://github.com/Crypto-toolbox/bitex.

[28] Mike van Rossum. A bitcoin trading bot written in node, 2013. Accessed on 7 March 2018. URL https://github.com/askmike/gekko.

[29] Tim Molter. A java library providing a streamlined api for interacting with 60+ bitcoin and altcoin exchanges, 2012. Accessed on 7 March 2018. URL https://github.com/timmolter/XChange.

[30] Maksim Stepanenko. Realtime cryptocurrency api, 2017. Accessed on 7 March 2018. URL https://github.com/lionsharecapital/lionshare-api.

[31] Chase. A command-line cryptocurrency trading bot using node.js and mongodb, 2016. Accessed on 7 March 2018. URL https://github.com/DeviaVir/zenbot.

[32] Christopher Bynum. Bitcoin arbitrage trading system, 2015. Accessed on 7 March 2018. URL https://github.com/cbyn/bitfx.

[33] Carles Tubio. Self-hosted crypto trading bot (automated high frequency market making) in node.js, angular, typescript and c++, 2014. Accessed on 7 March 2018. URL https://github.com/ctubio/Krypto-trading-bot.