



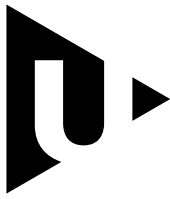
FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

KATEDRA INFORMATIKY
A VÝPOČETNÍ TECHNIKY

Diplomová práce

Burzovní simulátor pro trh řízený limitními objednávkami

Vladimíra Kimlová



**FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI**

**KATEDRA INFORMATIKY
A VÝPOČETNÍ TECHNIKY**

Diplomová práce

Burzovní simulátor pro trh řízený limitními objednávkami

Bc. Vladimíra Kimlová

Vedoucí práce

doc. Ing. Jan Pospíšil, Ph.D.

© Vladimíra Kimlová, 2025.

Všechna práva vyhrazena. Žádná část tohoto dokumentu nesmí být reprodukována ani rozšiřována jakoukoli formou, elektronicky či mechanicky, fotokopírováním, nahráváním nebo jiným způsobem, nebo uložena v systému pro ukládání a vyhledávání informací bez písemného souhlasu držitelů autorských práv.

Citace v seznamu literatury:

KIMLOVÁ, Vladimíra. *Burzovní simulátor pro trh řízený limitními objednávkami*. Plzeň, 2025. Diplomová práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky. Vedoucí práce doc. Ing. Jan Pospíšil, Ph.D.

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2024/2025

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	Bc. Vladimíra KIMLOVÁ
Osobní číslo:	A23N0102P
Studijní program:	N0613A140037 Informatika a její specializace
Specializace:	Zpracování přirozeného jazyka
Téma práce:	Burzovní simulátor pro trh řízený limitními objednávkami
Zadávací katedra:	Katedra informatiky a výpočetní techniky

Zásady pro vypracování

1. Seznamte se s možnostmi simulace obchodování na burze řízené knihou objednávek a zpracujte řešerši existujících open-source simulátorů burzy fungujících na principu odbavování limitních objednávek.
2. Navrhněte a pomocí vhodných vývojových nástrojů implementujte vlastní burzovní simulátor, který bude umožňovat simulaci obchodování prostřednictvím autonomních i manuálních obchodních agentů, jejich snadné přidávání, odebrání a modifikaci jejich chování.
3. Dále implementujte alespoň pět odlišných agentů s různými obchodními algoritmy, a z toho jeden "podvodný" (tedy takový, který provádí tzv. spoofing). Agenti by měli pracovat na principech strojového učení.
4. Realizujte různé velké simulace na vhodných platformách (e-INFRA CZ, MetaCentrum). Připravte nástroj pro generování přehledného reportu se statistickými údaji z realizovaných simulací a následně vhodným způsobem detailně vizualizujte srovnání úspěšnosti jednotlivých obchodních strategií.
5. Navržené postupy a dosažené výsledky důkladně popište v průvodním dokumentu práce.

Rozsah diplomové práce: **doporuč. 50 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

dodá vedoucí diplomové práce

Vedoucí diplomové práce: **Doc. Ing. Jan Pospíšil, Ph.D.**
Katedra matematiky

Datum zadání diplomové práce: **9. září 2024**
Termín odevzdání diplomové práce: **15. května 2025**

L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 30. září 2024

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného akademického titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., zákon o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění, a zejména skutečnost, že Západočeská univerzita v Plzni má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Plzni dne 15. května 2025

.....

Vladimíra Kimlová

V textu jsou použity názvy produktů, technologií, služeb, aplikací, společností apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Abstrakt

Rychlost, anonymita a komplexní interakce účastníků činí z finančních trhů extrémně náročné prostředí pro analýzu i regulaci. Tato práce představuje simulační platformu pro modelování trhu řízeného limitními objednávkami, s důrazem na realističnost, modulární rozšiřitelnost a systematické testování obchodních strategií. Základem systému je realistický párovací algoritmus a implementace obchodních agentů včetně modelu manipulativního chování. Architektura simulátoru je optimalizována na nízkou latenci, vysokou propustnost a flexibilní konfiguraci tržních podmínek. Experimentální simulace potvrzují schopnost systému věrně reprodukovat dynamiku trhu a odhalovat dopady manipulace na jeho stabilitu a efektivitu. Výsledné řešení poskytuje efektivní nástroj pro výzkum mikrostruktury finančních trhů, algoritmického obchodování a detekce podvodných praktik.

Abstract

The velocity, anonymity, and intricate interactions among participants render financial markets an exceptionally challenging domain for analysis and regulation. This thesis introduces a comprehensive simulation platform for modeling market driven by limit order books, emphasizing realism, modular extensibility, and the systematic evaluation of trading strategies. At its core, the system integrates an advanced matching engine and a suite of trading agents, including models designed to replicate manipulative behavior. The simulator architecture is meticulously optimized for low latency, high throughput, and flexible parameterization of market conditions. Experimental simulations validate the platform's capability to faithfully replicate key aspects of market microstructure and to expose the negative effects of manipulation on market stability and efficiency. The developed solution constitutes a robust research instrument for the study of market dynamics, algorithmic trading methodologies, and the detection of malicious trading practices.

Klíčová slova

burzovní simulátor • obchodní agenti • agentové modelování • limitní kniha objednávek • strojové učení • simulace obchodování • spoofing

Poděkování

Ráda bych nejprve poděkovala panu doc. Ing. Janu Pospíšilovi, Ph.D., vedoucímu mé diplomové práce, za jeho odborné vedení, cenné rady a podporu, které mi poskytoval během celého procesu. Jeho ochota věnovat mi svůj čas, energii a sdílet své znalosti byla neocenitelná a měla zásadní vliv na kvalitu a dokončení této práce.

Také bych chtěla poděkovat Davidu Rodriguezovi Merovi, dennímu obchodníkovi, jehož konzultace mi umožnily hlubší pochopení dynamiky finančních trhů a zásadně přispěly k validitě a realističnosti simulace vyvinuté v rámci mé práce.

Ráda bych také poděkovala Ing. Kamilu Ekšteinoovi, Ph.D., za poskytnutou příležitost podílet se na projektu a za odbornou podporu při vstupu do řešené tematiky.

Mé upřímné poděkování patří rovněž mé rodině, speciálně mému příteli, za jejich trpělivost, povzbuzení a neustálou podporu po celou dobu mého studia. Jejich důvěra ve mě, láska a laskavá slova mi dodávala sílu a motivaci překonat všechny výzvy a úspěšně dokončit tuto náročnou etapu mého života.

Vladimíra Kimlová

Výpočetní a úložné zdroje byly poskytnuty v rámci projektu **e-INFRA CZ (ID:90254)**, podpořeného Ministerstvem školství, mládeže a tělovýchovy České republiky.

Tato práce byla také částečně podpořena z Fondu rozvoje **CESNET**, projekt č. 734/2023 Ukládání, přenos a zpracování velkých vědecko-výzkumných finančních dat v prostředí **e-INFRA CZ**.

Obsah

1	Úvod	1
1.1	Současný stav problematiky	2
1.2	Cíle práce	4
1.3	Struktura práce	5
2	Teoretické pozadí a metodologie	7
2.1	Limitní kniha objednávek (LOB)	7
2.1.1	Klíčové charakteristiky a parametry limitní knihy objednávek	8
2.1.2	Typologie objednávek	11
2.1.3	Algoritmy pro vypořádání objednávek	11
2.1.4	Strategie obchodování	14
2.1.5	Manipulativní strategie a tržní zneužití	18
2.1.6	Modelování limitní knihy objednávek	19
2.2	Financial Information eXchange protokol	20
2.2.1	Struktura zpráv FIX	20
2.3	Agentové modelování	22
2.3.1	Definice agenta	22
2.3.2	Typy agentů	23
2.3.3	Vlastnosti agentů	23
2.3.4	Multiagentní systémy (MAS)	25
2.3.5	Klasifikace multiagentních systémů	26
2.3.6	Komunikace v multiagentních systémech	26
2.3.7	Agentové modelování v burzovních simulátorech	27
3	Implementovaná řešení	29
3.1	Architektura simulátoru	29
3.2	Obchodní server	30
3.2.1	Limitní kniha objednávek	31
3.2.2	Vypořádávání objednávek	32
3.2.3	Integrovaná správa produktů a uživatelských financí	32

3.2.4	Struktura serveru	33
3.2.5	Databáze uživatelů	34
3.3	Klientská část aplikace	34
3.3.1	Hlavní komponenty klientské aplikace	34
3.3.2	Administrátorští agenti	37
3.3.3	Algoritmickí obchodní agenti	39
3.3.4	Manuální obchodní agent	48
3.4	Testování systému	50
3.4.1	Testované scénáře	51
4	Experimenty a Diskuze	53
4.1	Nastavení experimentů	53
4.2	Test výkonnosti serveru	55
4.3	Velikonoční burzovní hra	56
4.4	Testování algoritmických obchodníků	60
4.4.1	Experiment s přítomností spoofingu	63
5	Závěr	65
A	Instalace a spuštění simulátoru	67
B	Uživatelská příručka pro použití webového rozhraní	71
C	Specifikace modifikovaného FIX protokolu	81
	Bibliografie	85
	Seznam obrázků	89
	Seznam tabulek	91

One of the funny things about the stock market is that every time one person buys, another sells, and both think they are astute.

*William Feather, American publisher and author
(1889–1981)*

Burzovní trhy řízené limitními objednávkami představují páteř současného finančního systému. Tyto trhy, kde poptávka a nabídka interagují prostřednictvím přesně definovaných obchodních příkazů, se vyznačují značnou dynamikou, komplexní strukturou a vysokou mírou konkurence. Díky své robustnosti jsou nejen důležitým nástrojem pro profesionální obchodníky a investory, ale také předmětem intenzivního výzkumu v akademické sféře. Studium trhů tohoto typu poskytuje cenný vhled do jejich vnitřních mechanismů, vlivu různých obchodních strategií na tržní dynamiku a strukturální odolnost vůči nepředvídatelným událostem.

Jedním z klíčových důvodů pro zkoumání těchto trhů prostřednictvím simulací je skutečnost, že reálná data z burzovního obchodování jsou často nedostupná nebo mají omezenou vypovídací hodnotu. Reálné transakční záznamy bývají anonymizovány, což může znemožnit analýzu individuálního chování účastníků trhu. Navíc jsou tato data často neúplná nebo poskytována s významnými časovými zpožděními, což znesnadňuje přesnou rekonstrukci historického vývoje trhu. Simulační prostředí proto představuje neocenitelný nástroj, který umožňuje detailní analýzu obchodních strategií, experimentování s inovativními přístupy, a to vše v kontrolovaném prostředí, kde lze libovolně upravovat podmínky a parametry trhu.

Tato diplomová práce se zaměřuje na návrh a implementaci komplexního simulačního prostředí pro modelování burzovního trhu řízeného objednávkami. Základem systému je párovací algoritmus objednávek využívající pravidla

Price-Time Priority, který věrně odráží mechanismus párování objednávek běžný v reálných obchodních systémech. Simulátor je postaven na architektuře *klient-server*, přičemž pro komunikaci mezi jednotlivými komponentami je využit protokol FIX (Financial Information eXchange), široce rozšířený standard ve finančním sektoru.

Součástí práce je také implementace různých algoritmických obchodních agentů, kteří reprezentují odlišné přístupy k automatizovanému obchodování, včetně agenta simulujícího manipulativní chování. Pro potřeby manuálního obchodování a interaktivního testování byl vytvořen webový klient, který poskytuje uživatelsky přívětivé rozhraní pro zadávání objednávek a sledování vývoje trhu v reálném čase. Nedílnou součástí řešení je rovněž nástroj pro systematické vyhodnocování obchodních strategií, který umožňuje analyzovat dosažené výsledky na základě definovaných metrik a zohlednit tak různé aspekty tržního chování.

1.1 Současný stav problematiky

Výzkum v oblasti *tržní mikrostruktury* a *simulace obchodování* zaznamenal v posledních letech dynamický rozvoj. Se vzestupem *vysokofrekvenčního obchodování (High Frequency Trading, HFT)* a *algoritmických obchodních strategií* se stále intenzivněji analyzuje dynamika finančních trhů, chování obchodních subjektů a efektivita různých přístupů k tvorbě trhu. Tento vývoj je podporován jak akademickým výzkumem, tak praktickými simulačními modely, které umožňují testování a validaci obchodních strategií v kontrolovaném prostředí bez expozice finančnímu riziku.

Přehled aktuálních metod simulace limitní objednávkové knihy (*Limit Order Book, LOB*) a jejich aplikací v *algoritmickém obchodování* nabízí článek [1]. Autoři porovnávají přístupy *bodových procesů (point process)*, *agent-based modelů (ABMs)*, *hlubokých neuronových sítí* a *stochastických parciálních diferenciálních rovnic (SPDEs)*, přičemž hodnotí jejich schopnost replikovat *stylizovaná fakta* a jejich využitelnost pro simulaci tržní dynamiky. Zvláštní pozornost věnují modelování *cenového dopadu* a zkoumají citlivost jednotlivých přístupů na *vnější obchodní zásahy*. Studie dále provádí srovnávací analýzu přesnosti simulací ve vztahu k *empirickým datům*.

Zajímavý příspěvek přináší studie [2], která využívá *agentový model* pro analýzu LOB a zkoumá vliv chování obchodníků na vývoj cen. Autoři identifikují fenomén *vnitřní bubliny*, jež vzniká i navzdory absenci reálných ekonomických faktorů, jako je inflace. Tento jev, způsobený běžným chováním obchodníků, naznačuje, že běžná činnost obchodníků, jejichž cílem je dosažení zisku, způsobuje falešný nárůst ceny.

Dalším významným příspěvkem k modelování finančních trhů je studie [3], která se zaměřuje na simulaci manipulace trhem prostřednictvím **spoofingu**, tedy techniky, kdy obchodníci vkládají falešné objednávky za účelem manipulace cenami. Tento model rozlišuje dva typy obchodníků: **zero intelligence (ZI)** ignorující objednávkovou knihu a **heuristic belief learning (HBL)**, kteří ji využívají k predikci cenových změn. Analýza ukazuje, že trhy s HBL obchodníky jsou zranitelné vůči spoofingu, což vede k poklesu tržního přebytku a podílu HBL. Model kvantifikuje dopady spoofingu na efektivitu trhu a chování obchodníků.

Práce [4] rozšiřuje **bottom-up** modely trhu zavedením modulárního obchodního prostředí v reálném čase nazvané **ABSTRACTE**, které umožňuje autonomní reprezentaci účastníkům trhu. Výzkum porovnává reálné a **umělé akciové trhy (Artificial Stock Markets, ASMs)**, **ABSTRACTE** si klade za cíl zachytit klíčové aspekty reálných trhů, jež jsou v **umělých simulacích** často opomíjeny, a poskytuje flexibilní platformu pro testování, hodnocení a porovnávání tržních modelů a chování obchodníků.

Simulátor **SHIFT** [5] přináší další krok v realistickém modelování tržního chování a testování obchodních strategií. Tento nástroj by měl umožnit simultánní připojení více automatizovaných obchodníků k tržnímu prostředí a je navržen pro zkoumání **tržních mikrostruktur**. Při vykonávání objednávek simuluje systém pravidlo *first-in-first-out* (FIFO). **SHIFT** implementuje pravidla amerických akciových trhů, podporuje jak tržní, tak limitní objednávky, a je schopen simulovat **tržní stresové situace**, které mohou vést ke krachům. Simulátor je k dispozici prostřednictvím API a nabízí open-source Python klienta, který je dostupný na GitHubu (<https://github.com/hanlonlab/shift-python>). **SHIFT** byl aktivně vyvíjen v letech 2017 až 2019 na Stevens Institute of Technology, současný stav a dostupnost nástroje nejsou v tuto chvíli veřejně známy.

Simulátor **Bristol Stock Exchange (BSE)** [6] představuje minimalistický model centralizovaného trhu využívajícího limitní objednávkovou knihu. BSE reaguje na potřebu dostupného prostředí pro výzkum **automatizovaného obchodování**, které nelze realizovat na reálných burzách kvůli jejich uzavřenosti vůči experimentům. Systém umožňuje zkoumat tržní mechanismy a poskytuje realistickou platformu pro výuku **algoritmického obchodování**. Od roku 2012 je využíván pro vzdělávací i výzkumné účely a je volně dostupný na GitHubu (<https://github.com/davecliff/BristolStockExchange>).

V neposlední řadě, studie [7] představuje **TRADES** (TRAnsformer-based Denoising Diffusion Probabilistic Engine for LOB Simulations), model pro generování realistických simulací limitní objednávkové knihy. **TRADES** využívá **transformer** architekturu k modelování toků objednávek, což umožňuje realistické a reaktivní

simulace trhů. Model zlepšuje kalibraci obchodních strategií a testování tržního dopadu. Vyhodnocení pomocí prediktivního skóre (Mean Absolute Error, MAE) ukazuje zlepšení o faktor 3,27–3,47 oproti současným metodám. **TRADES** je implementován v open-source frameworku DeepMarket a je dostupný na GitHubu (<https://github.com/LeonardoBerti00/DeepMarket>).

1.2 Cíle práce

Hlavním cílem této diplomové práce je návrh a implementace burzovního simulátoru pro trh řízený limitními objednávkami. Tento simulátor umožní simulaci obchodování prostřednictvím autonomních i manuálních obchodních agentů a poskytne nástroje pro analýzu a vizualizaci výsledků simulací. Dílčí cíle práce jsou následující:

1. Prostudovat možnosti simulace obchodování na trzích řízených knihou objednávek a provést rešerši existujících open-source burzovních simulátorů fungujících na principu odbavování limitních objednávek.
2. Navrhnout a implementovat burzovní simulátor, který bude umožňovat:
 - Interakci s trhem prostřednictvím autonomních i manuálně ovládaných obchodních entit.
 - Modulární správu obchodních agentů.
3. Implementovat reprezentativní sadu obchodních agentů využívajících odlišné strategie (včetně strojového učení) a podvodného agenta využívajícího manipulativní chování, s cílem analyzovat dopad takových praktik na dynamiku trhu.
4. Ověřit škálovatelnost systému prostřednictvím simulačních experimentů různých velikostí realizovaných na vhodných výpočetních platformách.
5. Vyvinout nástroj pro generování přehledných reportů obsahujících statistické údaje o realizovaných simulacích. A dále vizualizovat výsledky simulací s důrazem na srovnání úspěšnosti jednotlivých obchodních strategií.

1.3 Struktura práce

Tato diplomová práce je rozdělena do čtyř hlavních kapitol, které reflektují logickou strukturu postupu od teoretického ukotvení problematiky přes návrh a realizaci systému až po experimentální vyhodnocení.

První kapitola uvádí čtenáře do kontextu práce. Představuje základní charakteristiku burzovních trhů řízených limitními objednávkami a zdůrazňuje význam jejich simulace v akademickém a vývojovém prostředí. Následně je diskutován současný stav problematiky v dané oblasti a formulovány konkrétní cíle, kterých má být v rámci práce dosaženo.

Druhá kapitola poskytuje teoretické a metodologické zázemí potřebné pro pochopení fungování navrženého systému. Nejprve je analyzována struktura a dynamika limitní knihy objednávek, včetně klasifikace příkazů, pravidel párování a obchodních strategií, včetně těch manipulativních. Dále je popsán komunikační protokol FIX jakožto klíčový prvek systémové integrace. Závěr kapitoly je věnován principům agentového modelování, typologii agentů a jejich využití v simulaci tržního prostředí.

Třetí kapitola se zaměřuje na návrh a implementaci vlastního simulačního systému. Popsána je jeho softwarová architektura, včetně obchodního serveru, správy dat a klientské aplikace, umožňující jak manuální, tak automatizované obchodování. Součástí kapitoly je i popis jednotlivých implementovaných obchodních agentů. Kapitola dále představuje metodiku testování systému a definuje scénáře, které byly použity k validaci jeho funkčnosti.

Čtvrtá kapitola je věnována experimentální části práce, jsou zde prezentovány výsledky simulací, jejich kvantitativní vyhodnocení a porovnání výkonnosti jednotlivých strategií v různých tržních podmínkách. Tato část poskytuje důležitý vhled do chování obchodníků a efektivity zvolených přístupů.

Teoretické pozadí a metodologie

2

Tato kapitola poskytuje přehled teoretického rámce a metodologií relevantních pro modelování a simulaci finančních trhů. Nejprve se zaměřuje na **limitní knihu objednávek**, která představuje klíčový mechanismus organizace obchodování na moderních burzách. Dále se kapitola věnuje protokolu **Financial Information eXchange (FIX)**, jenž je standardem pro komunikaci mezi obchodními platformami. Poslední část je věnována **agentovému modelování**, které umožňuje simulaci tržní dynamiky prostřednictvím interakce autonomních agentů.

2.1 Limitní kniha objednávek (LOB)

Limitní kniha objednávek (**Limit Order Book**, LOB) je základním prvkem moderních elektronických trhů, jakými jsou například trhy s akciemi, komoditami, měnami a s jejich deriváty. Jedná se o centralizovaný systém, který shromažďuje a organizuje všechny objednávky účastníků trhu podle přísně stanovených pravidel. Každá objednávka v LOB představuje závazek ke koupi nebo prodeji obchodovaného aktiva, je charakterizována souborem parametrů, mezi něž patří typ objednávky (nákupní nebo prodejní), cena, množství, čas zadání objednávky do systému a další údaje specifické pro konkrétní trh [8].

V každém okamžiku obsahuje limitní kniha objednávek úplný seznam všech možných transakcí, které mohou být na daném trhu realizovány. Objednávky v ní jsou obvykle řazeny podle:

1. **Ceny:** Objednávky jsou v první řadě řazeny podle nabízené či požadované ceny. U nákupních objednávek (**Bids**) mají vyšší prioritu vyšší cenové úrovně, zatímco u prodejních objednávek (**Asks**) jsou upřednostňovány nižší ceny.
2. **Času zadání:** Při shodné cenové úrovni jsou objednávky řazeny podle chronologického pořadí jejich přijetí do systému (tzv. pravidlo **FIFO** – First In, First Out).

Dynamika limitní knihy odráží interakce mezi účastníky trhu, kteří svým kolektivním chováním přispívají k tvorbě aktuální tržní ceny. Z tohoto důvodu představují limitní knihy objednávek objekt zájmu nejen teoretiků a akademiků v oblasti financí a ekonometrie, ale také odborníků z příbuzných oborů, jako jsou statistická fyzika, informatika či finanční inženýrství.

2.1.1 Klíčové charakteristiky a parametry limitní knihy objednávek

Základní parametry **LOB** definují způsob, jakým jsou objednávky evidovány a zpracovávány, a zahrnují následující pojmy [9]:

- **Ask cena**, označovaná jako poptávková cena, představuje nejnižší cenu, za kterou je aktuálně prodávající obchodník ochoten prodat.
- **Bid cena**, tedy nabídková cena, je nejvyšší cenou, za kterou lze je kupující obchodník ochoten koupit.
- Rozdíl mezi těmito dvěma cenami, známý jako **Bid-Ask Spread**, je klíčovým ukazatelem efektivity a likvidity trhu. **Bid-Ask Spread** je formálně definován vztahem

$$s(t) := a(t) - b(t), \quad (2.1)$$

kde $a(t)$ je **Ask cena** a $b(t)$ je **Bid cena** v daném čase .

Střední cena, často nazývaná **Mid-Price**, je aritmetickým průměrem **Bid** a **Ask** ceny. Lze ji vyjádřit jako

$$m(t) := \frac{a(t) + b(t)}{2}. \quad (2.2)$$

Tato hodnota je významná pro hodnocení aktuálního tržního sentimentu, neboť poskytuje referenční bod mezi nabídkovou a poptávkovou stranou trhu [9].

Další důležitou charakteristikou **LOB** jsou tzv. parametry rozlišení [9], kam patří:

- **Tick Size** π : nejmenší povolený cenový rozdíl mezi dvěma objednávkami. Určuje přesnost, s níž musí být ceny objednávek zadávány.
- **Lot Size** σ : minimální obchodovatelné množství aktiva. Každá objednávka musí mít velikost ω_x , která splňuje podmínku $\omega_x \in \{\pm k\sigma \mid k = 1, 2, \dots\}$.

Parametry jako nejlepší nabídková cena a nejlepší poptávková cena jsou shrnuty pod pojmem **Top of Book**. Tyto hodnoty mají zásadní význam při realizaci tržních objednávek, neboť představují nejvýhodnější dostupné ceny.

Hloubka knihy, neboli součet všech objednávek na určité cenové úrovni, se rozlišuje podle **Bid** a **Ask** strany. Hloubka na **Bid** straně $n_b(p, t)$ a na **Ask** straně $n_a(p, t)$ je definována následovně [9]:

$$n_b(p, t) := \sum_{x \in B(t) | p_x = p} \omega_x^{(b)}(t) \quad \text{a} \quad n_a(p, t) := \sum_{x \in A(t) | p_x = p} \omega_x^{(a)}(t), \quad (2.3)$$

kde p označuje konkrétní cenovou úroveň v čase t a ω_x představuje objem dané objednávky. Tyto hodnoty ukazují, jaký objem objednávek je na trhu dostupný při daných cenách.

Šířka knihy, neboli počet aktivních cenových úrovní, na kterých se nachází alespoň jedna objednávka, se podobně jako hloubka rozlišuje podle **Bid** a **Ask** strany. Šířka knihy tak ukazuje, jak rozprostřená je objednávková aktivita na jednotlivých stranách trhu.

Closing price (závěrečná cena) je cena aktiva zaznamenaná na konci obchodního dne na konkrétním trhu nebo burze. Tato cena odráží hodnotu, za kterou bylo aktivum naposledy zobchodováno před oficiálním uzavřením trhu [10]. Je však třeba rozlišovat mezi závěrečnou cenou a **Adjusted Closing Price**, která upravuje původní hodnotu o korporátní události (například výplaty dividend) a lépe tak odráží skutečnou návratnost pro investora [11].

Likvidita

Likvidita označuje schopnost aktiva být rychle a efektivně zakoupeno nebo prodáno na trhu, aniž by došlo k výraznému ovlivnění jeho ceny. Vysoká **likvidita** znamená, že v **objednávkové knize** je velké množství objednávek, což usnadňuje realizaci obchodu za požadovanou cenu. Naopak nízká **likvidita** implikuje menší počet objednávek, což může vést k **cenovému skluzu**, kdy skutečná cena realizovaného obchodu značně odlišuje od očekávané ceny v důsledku změn nabídky a poptávky na trhu. **Likvidita** trhu je klíčovým faktorem pro hodnocení efektivitu a fungování finančních trhů. Existují tři hlavní charakteristiky **likvidity** [12]:

- **Těsnost (Tightness)**, která označuje náklady na realizaci obchodní pozice v krátkém časovém období.
- **Hloubka (Depth)**, která vyjadřuje množství nových objednávek potřebných k významnému ovlivnění tržních cen.
- **Odolnost (Resiliency)**, která popisuje schopnost trhu obnovit rovnováhu po náhodném, neinformativním šoku.

Celkově je **likvidita** komplexním konceptem, jehož charakteristiky přímo ovlivňují efektivitu párovacího algoritmu v **LOB** a schopnost trhu poskytovat konkurenční ceny v dynamickém prostředí.

V případě, že objednávková kniha není dostatečně naplněna objednávkami, jakýkoli významnější obchod může mít zásadní dopad na tržní cenu, což může vést k realizaci méně výhodných obchodních podmínek. Tento jev je obzvláště patrný na trzích s méně populárními kryptoměnami nebo během období s nízkým obchodním objemem. V těchto případech může absence dostatečného množství objednávek v LOB způsobit, že trh nebude schopen absorbovat větší obchodní transakce bez významného vlivu na cenovou hladinu.

V rámci této problematiky se účastníci zadávající limitní objednávky označují jako **poskytovatelé likvidity (Liquidity Providers)**, zatímco ti, kdo zadávají **tržní objednávky**, jsou považováni za **odběratele likvidity (Liquidity Takers)**. Je však třeba upozornit na skutečnost, že v reálných tržních podmínkách tento dichotomický pohled není zcela přesný, neboť v mnoha případech účastníci trhu vykazují chování, které nelze jednoduše zařadit do jedné z těchto kategorií [8].

Imbalance index

Imbalance index představuje kvantitativní míru nerovnováhy mezi agregovanými objemy poptávkových (*Ask*) a nabídkových (*Bid*) objednávek v limitní knize objednávek. Tento indikátor poskytuje hlubší vhled do mikrostruktury trhu a umožňuje analyzovat dynamiku nabídky a poptávky, což má přímé implikace pro očekávaný směr pohybu tržních cen. Formální definice indexu je následující [13]:

$$I(t) = \frac{V^{(b)}(t) - V^{(a)}(t)}{V^{(b)}(t) + V^{(a)}(t)}, \quad (2.4)$$

kde $V^{(b)}(t)$ a $V^{(a)}(t)$ jsou definovány jako vážené průměry objemů na příslušných úrovních. Konkrétně pro $V^{(b)}(t)$ platí:

$$V^{(b)}(t) = \sum_{x=1}^L \omega_x \omega_x^{(b)}(t),$$

kde ω_x představuje váhy, které mohou být voleny různými způsoby, například jako exponenciálně klesající váhy $\omega_x = e^{-\lambda(x-1)}$ [13]. L označuje maximální počet cenových úrovní, které jsou uvažovány, $\omega_x^{(b)}(t)$ je objem nabídky na úrovni x v čase t . Podobně je $V^{(a)}(t)$ definováno pro poptávku. Parametr $\lambda > 0$ je hyperparametr, který řídí vliv jednotlivých úrovní na průměrný objem.

Imbalance index nabývá hodnot v intervalu $[-1, 1]$, přičemž:

- Hodnoty blízké 1 signalizují dominanci nabídky (*Bid*) nad poptávkou (*Ask*), což podporuje růst cen.
- Hodnoty blízké -1 indikují převahu poptávky (*Ask*) nad nabídkou (*Bid*), což obvykle vytváří tlak na pokles tržní ceny.
- Hodnoty v blízkosti 0 značí relativní rovnováhu mezi nabídkovou a poptávkovou stranou trhu.

Imbalance index tak představuje klíčový parametr pro analýzu tržních trendů, neboť odráží okamžitý sentiment a strukturální vlastnosti objednávkové knihy.

2.1.2 Typologie objednávek

Moderní finanční trhy využívající systém knihy objednávek umožňují širokou škálu typů objednávek, které účastníkům poskytují flexibilitu a možnost optimalizace jejich obchodních strategií. Mezi základní typy objednávek patří [8]:

Limitní objednávka (Limit Order) *Limitní objednávka* představuje instrukci k nákupu nebo prodeji aktiva za specifikovanou cenu nebo za cenu výhodnější. Tento typ objednávky nevstupuje do okamžité realizace, pokud aktuální tržní podmínky nesplňují stanovenou cenovou hranici.

Tržní objednávka (Market Order) *Tržní objednávka* je definována jako instrukce k okamžitému nákupu či prodeji aktiva za nejlepší aktuálně dostupnou cenu na opačné straně trhu. Absence požadavku na specifickou cenu zajišťuje rychlou realizaci, což je výhodné zejména v situacích, kdy je prioritou rychlost transakce před kontrolou cenové úrovně.

Zrušení objednávky (Cancellation Order) *Zrušení objednávky* slouží k odstranění existující *limitní objednávky* z knihy objednávek. Tento typ příkazu je zásadní pro správu rizika a přizpůsobení obchodní strategie měnícím se tržním podmínkám.

2.1.3 Algoritmy pro vypořádání objednávek

Algoritmy pro vypořádání objednávek jsou zásadními mechanismy, které zabezpečují efektivní a spravedlivé párování poptávkových a nabídkových

objednávek na finančních trzích. Tyto algoritmy mají rozhodující vliv na likviditu trhu, rychlost transakcí a transparentnost obchodních operací. Mezi nejvýznamnější přístupy k párování objednávek patří **Price-Time Priority** a **Pro Rata Matching**, které jsou využívány v závislosti na charakteristikách konkrétního trhu a potřebách účastníků obchodování.

Price-Time Priority

Price-Time Priority představuje nejrozšířenější algoritmus pro párování objednávek, jenž je implementován především na akciových burzách a jiných likvidních trzích. Tento algoritmus přisuzuje prioritu objednávkám na základě jejich ceny a času zadání. Nejprve jsou párovány objednávky s nejlepšími cenovými podmínkami, přičemž v případě shody cenových hladin má přednost objednávka, která byla zadána dříve – pravidlo FIFO (*First In, First Out*). Tento přístup je fundamentálně navržen tak, aby podporoval konkurenci mezi účastníky trhu, což vede k užším spreadům a zvyšuje efektivitu alokace likvidity [8, 14].

Příklad pro **Price-Time Priority**:

Na trhu jsou aktuálně evidovány následující nákupní objednávky:

- **Objednávka 1:** 10 akcií za 50 Kč (zadáno v 10:00),
- **Objednávka 2:** 20 akcií za 50 Kč (zadáno v 10:05),
- **Objednávka 3:** 50 akcií za 51 Kč (zadáno v 10:02).

Následně je zadána limitní prodejní objednávka na prodej 80 akcií za cenu 50 Kč.

Objednávky jsou zpracovávány v souladu s principem cenové a časové priority, tj. primárně podle nabízené ceny (přednost má vyšší nákupní cena), sekundárně podle času zadání (přednost má dříve zadaná objednávka). Nejvyšší limitní cenu předkládá Objednávka 3 (51 Kč), a je proto realizována jako první. Zbývajících 30 akcií je následně alokován objednávkám na cenové hladině 50 Kč, přičemž přednostní plnění získává Objednávka 1 vzhledem k dřívějšímu času zadání ve srovnání s Objednávkou 2.

Price-Time Priority algoritmus je ceněn pro svou transparentnost a jednoduchost, neboť všechny objednávky jsou pořadovány podle jasně definovaných kritérií, jež minimalizují subjektivitu rozhodování a zajišťují spravedlivý přístup ke kapitálovým trhům.

Pro Rata Matching

Pro Rata Matching představuje sofistikovaný algoritmus pro párování objednávek, jenž je rozšířen především na trzích, které kladou důraz na proporcionální rozdělení likvidity mezi účastníky obchodování. Tento algoritmus páruje objednávky s identickými cenovými podmínkami na základě jejich objemu, přičemž v případě existence více objednávek na téže cenové hladině dochází k jejich vykonání v poměru k velikosti jednotlivých objednávek [14].

Příklad pro **Pro Rata Matching**:

Na trhu jsou zadány tři limitní příkazy na nákup za shodnou cenu 50 Kč:

- **Objednávka A:** 100 akcií za 50 Kč,
- **Objednávka B:** 200 akcií za 50 Kč,
- **Objednávka C:** 300 akcií za 50 Kč.

Následně je na trh zadána objednávka na prodej 60 akcií za 50 Kč. V systému typu **Pro Rata Matching** je dostupná likvidita (60 akcií) rozdělena mezi jednotlivé nákupní objednávky poměrně podle jejich podílu na celkovém požadovaném objemu (600 akcií). Výsledné alokace jsou následující:

- Objednávka A obdrží 16.67% (10 akcií),
- Objednávka B obdrží 33.33% (20 akcií),
- Objednávka C obdrží 50% (30 akcií).

Hlavním přínosem tohoto přístupu je zajištění spravedlivější a rovnoměrnější alokace likvidity, což je zvláště relevantní na trzích, kde se předpokládá přítomnost širokého spektra účastníků soutěžících o realizaci objednávek za identické cenové podmínky, například na komoditních trzích. Na rozdíl od metodiky *Price-Time Priority*, která upřednostňuje objednávky na základě jejich časového pořadí, čímž činí časový moment zadání klíčovým faktorem pro určení priority, **Pro Rata Matching** zaručuje, že všechny objednávky na stejné cenové hladině mají stejný základ pro distribuci likvidity. Tento mechanismus tak minimalizuje riziko fenoménu známého jako *front-running*, kdy účastníci trhu obchodují na základě informací o již zadáných objednávkách ostatních účastníků [14].

Proces párování

Při příchodu nové objednávky do LOB dochází k jejímu zařazení do knihy nebo okamžitému párování s již existujícími objednávkami. Tento proces je popsán následujícími scénáři [9]:

1. **Limitní objednávka v aktivní zóně:** Pokud cena nové objednávky leží mimo nebo na hranici aktuální ceny *Bid* či *Ask*, objednávka je ihned párována s existující protistranou.
2. **Limitní objednávka v neaktivní zóně:** Pokud cena objednávky spadá mezi aktuální cenu *Bid* a *Ask*, objednávka je zařazena do knihy a může ovlivnit vývoj ceny.
3. **Tržní objednávka:** Tržní objednávka je spárována s nejlepší dostupnou objednávkou na opačné straně trhu. Pokud její velikost přesahuje dostupné množství, zbytek objednávky zůstává aktivní.

2.1.4 Strategie obchodování

Obchodní strategie je plán využívající analýzu k určení vhodných tržních podmínek a cenových úrovní pro vstup či výstup z obchodu. Zatímco **fundamentální analýza** slouží k predikci cenových pohybů na základě makroekonomických a finančních faktorů, většina obchodních strategií se primárně opírá o **technické indikátory**, které vycházejí z historických cenových dat a objemů obchodů.

Je důležité rozlišovat mezi **obchodní strategií** (Trading Strategy) a **obchodním stylem** (Trading Style). **Obchodní styl** je širší koncept, který určuje, jak často obchodník obchoduje a jak dlouho drží pozice. Naproti tomu **obchodní strategie** představuje konkrétní metodiku definující, při jakých cenových úrovních zahájit obchod nebo jej ukončit. **Obchodní styl** reflektuje individuální preference obchodníka, například zda obchoduje krátkodobě či dlouhodobě, a může se přizpůsobit měnícím se tržním podmínkám [15].

Klíčové komponenty obchodních strategií

Každá úspěšná **obchodní strategie** se opírá o několik zásadních prvků, které zajišťují její účinnost a udržitelnost:

- **Řízení rizik:** Efektivní **řízení rizik** chrání kapitál obchodníka a minimalizuje ztráty. Zahrnuje stanovení **stop loss limitů**, které určují maximální riziko na jeden obchod.

- **Technická analýza:** *Technická analýza* pomáhá předpovědět cenové pohyby pomocí analýzy grafů, indikátorů a vzorců pro podporu rozhodování.
- **Obchodní plán:** *Obchodní plán* spojuje všechny strategie a určuje vstupy, výstupy, velikost pozic a obchodní cíle.

Typy obchodních strategií

Následující sekce se věnuje nejběžnějším obchodním strategiím, které se používají k dosažení zisků v různých podmínkách a typech trhů.

Day trading *Day trading* je strategie zaměřená na nákup a prodej cenných papírů v rámci jednoho obchodního dne s cílem využít krátkodobých cenových pohybů. Vyžaduje vysokou disciplínu, rychlé rozhodování a zaměření na **likvidní aktiva**, jako jsou měny, akcie velkých společností nebo indexy. Hlavními výhodami jsou vyhnutí se rizikům spojeným s držením pozic přes noc a příležitost ke krátkodobým ziskům. Mezi nevýhody patří vysoké transakční náklady, psychická náročnost a potřeba intenzivního sledování trhu [16].

Swing trading *Swing trading* je obchodní strategie zaměřená na zachycení krátkodobých až střednědobých cenových pohybů. Využívá technickou analýzu k identifikaci příležitostí vznikajících v důsledku dočasných cenových odchylek v rámci tržních trendů. Výhodou této strategie je možnost dosáhnout vyšších zisků díky delšímu držení pozic a nižší časové náročnosti ve srovnání s intradenním obchodováním. Nevýhodou této strategie je vyšší míra rizika spojená s tržní volatilitou a nezbytnost udržování disciplíny při identifikaci a využívání optimálních časových okamžiků pro realizaci obchodních pozic [16].

Scalping *Scalping* představuje krátkodobou obchodní strategii zaměřenou na realizaci četných obchodů během velmi krátkého časového horizontu s cílem profitovat z drobných cenových pohybů. Obchodníci využívající tuto strategii se snaží těžit z **Bid/Ask Spreadu**, tedy rozdílu mezi nabídkovou a poptávkovou cenou, což umožňuje generovat malé, avšak časté zisky. Tato strategie je nejvhodnější pro zkušené obchodníky, kteří disponují pokročilými technologiemi pro **vysokofrekvenční obchodování** a schopností rychle reagovat na tržní pohyby. Hlavní výhodou scalpingu je nízké riziko spojené s držením pozic, neboť obchodníci obvykle vstupují a vystupují z trhu během několika sekund až minut. Naopak nevýhodou jsou výrazné **transakční náklady**, které mohou vznikat

v důsledku vysokého objemu obchodů, a nutnost intenzivního sledování trhu, což klade vysoké nároky na psychickou odolnost obchodníka [17].

Arbitrážní obchodování *Arbitrážní obchodování* je strategie využívající cenových rozdílů mezi trhy či burzami. Obchodník nakupuje aktiva na trhu s nižší cenou a prodává je na trhu s vyšší cenou, čímž generuje zisk. Tato strategie je nízkoriziková, protože cenové disparity bývají krátkodobé a rychle eliminovány. Arbitráž vyžaduje pokročilé technické znalosti, rychlou exekuci a přístup k více trhům, což ji činí vhodnou především pro zkušené obchodníky. Výhodou je univerzálnost této strategie a potenciál zisků bez ohledu na tržní podmínky. Nevýhodou jsou nízké marže, které vyžadují vysoké objemy obchodů, a postupné ubývání příležitostí v důsledku rostoucí tržní efektivity a technologického pokroku [18].

Gap Trading *Gap trading* je obchodní strategie, která se zaměřuje na využívání cenových mezer, kdy cena aktiva náhle vzroste nebo poklesne, aniž by mezi těmito pohyby probíhalo výrazné obchodování. Mezera, neboli **gap**, vzniká například v důsledku neočekávaných zpráv nebo technických faktorů. Hlavní výhodou této strategie je možnost rychlého zisku díky náhlým cenovým pohybům, například po významných novinkách nebo tržních událostech. Tato strategie také poskytuje jasné úrovně pro vstup a výstup z obchodu, což může obchodníkovi pomoci řídit riziko. Naopak, nevýhodou je, že mezery vznikají nečekaně a mohou se rychle vyplnit, což vyžaduje rychlou reakci obchodníka [19].

Trend Trading *Trend trading* je strategie, která se zaměřuje na obchodování v souladu s aktuálním tržním trendem, ať už vzestupným, nebo sestupným. Obchodníci tuto strategii využívají k maximalizaci zisků z dlouhodobých cenových pohybů, přičemž pozice obvykle drží delší dobu. Tato metoda je vhodná především pro **swing tradery** a **position tradery**, kteří se orientují na střednědobé a dlouhodobé trendy. Výhodou **trend tradingu** je jeho jednoduchost a vysoká úspěšnost v trzích, které vykazují jasný a stabilní trend. Nevýhodou je riziko nečekaných obrátů trendů, což může vést k ztrátám [20].

Momentum trading *Momentum trading* je krátkodobá strategie, která se zaměřuje na využívání silných cenových pohybů. Obchodníci identifikují aktiva s výrazným růstem či poklesem ceny a obchodují ve směru tohoto trendu, často s pomocí technických indikátorů, jako jsou například klouzavé průměry. Tato strategie je

oblíbená u krátkodobých obchodníků, kteří sledují rychlé cenové změny. Mezi hlavní výhody patří možnost vysokých zisků během silných tržních trendů a flexibilita vstupu s nízkou počáteční investicí. Nevýhody zahrnují riziko falešných trendů, náhlých obrátů a potřebu rychlé reakce a pokročilých znalostí technické analýzy [21].

Mezi další populární strategie patří [22]:

- **Range trading** je strategie, která využívá pohyby ceny mezi úrovněmi *support*¹ a *resistance*², přičemž se nakupuje blízko *support* a prodává blízko *resistance*, ideální pro trhy v konsolidaci.
- **Position trading** je dlouhodobá strategie, kde obchodník drží pozici týdny až roky, s cílem profitovat z velkých tržních trendů, přičemž se spoléhá na fundamentální analýzu.
- **Breakout trading** je strategie zaměřená na vstup do trhu při průrazu klíčových úrovní *support* nebo *resistance*, přičemž obchodník využívá momentum k zachycení větších cenových pohybů.
- **News trading** se zaměřuje na využívání krátkodobých tržních pohybů vyvolaných významnými zprávami, jako jsou ekonomické údaje, politické události či zprávy publikované na sociálních sítích. Tato strategie je vhodná pro obchodníky s dobrým pochopením fundamentální analýzy.
- **End-of-day trading** je strategie, při které obchodník analyzuje trh na konci obchodního dne, s cílem identifikovat dlouhodobé trendy a minimalizovat šum intradenního obchodování.

Výkon obchodní strategie je tradičně hodnocen na základě **poměru rizika a výnosu**. Nejznámějším používaným ukazatelem pro kvantitativní měření tohoto poměru je **Sharpeho poměrový koeficient (Sharp Ratio)** [23], jež vyjadřuje vztah mezi **průměrným výnosem** a **volatilitou výnosů**. V praxi se nicméně často provádí porovnání **očekávaného výnosu** s **volatilitou** nebo **maximálním poklesem** (tzv. **drawdown**). Obecně platí, že vyšší **očekávaný výnos** bývá spojen s vyšší **volatilitou** a větším rizikem **drawdownu**. Výběr vhodného rizikově-výnosového poměru je silně závislý na individuálních preferencích obchodníka ohledně **rizika**.

Obchodní strategie mohou být realizovány buď manuálně, v rámci **discretionary trading**, nebo automatizovaně (**automated trading**). **Discretionary trading**

¹Support (podpora) označuje cenovou úroveň, při které je větší poptávka po aktivu, což zamezuje dalšímu poklesu ceny.

²Resistance (odolnost) označuje cenovou úroveň, kde se nachází větší nabídka, což brání dalšímu růstu ceny.

vyžaduje vysokou úroveň odborných schopností a disciplíny, přičemž obchodník je často vystaven riziku odchýlení se od stanovené strategie, což negativně ovlivňuje výkon. Naproti tomu **automatizované strategie** zahrnují implementaci obchodních pravidel do automatizovaných systémů, které provádějí obchodní příkazy na základě předem definovaných algoritmů. Využití pokročilých modelů a globálních tržních dat poskytuje obchodníkům s automatizovanými systémy výraznou konkurenční výhodu.

2.1.5 Manipulativní strategie a tržní zneužití

Finanční trhy jsou navrženy tak, aby umožňovaly spravedlivé a transparentní obchodování, avšak existují techniky, které obchodníci využívají k manipulaci trhu ve svůj prospěch. Mezi nejznámější manipulativní strategie patří **quote stuffing**, **spoofing** a **layering**, které narušují přirozenou tvorbu cen.

Quote Stuffing

Quote stuffing je obchodní praktika spočívající v zadávání a následném okamžitém rušení velkého množství příkazů k nákupu či prodeji. Tato strategie je primárně využívána obchodníky na vysokofrekvenčních trzích (**High-Frequency Traders**) za účelem manipulace s tržní cenou finančního instrumentu či akciového indexu a dosažení arbitrážních zisků. Princip této taktiky spočívá v zahlcení objednávkového systému burzy extrémně vysokým počtem příkazů, čímž dochází k dočasnému zkreslení poptávkových a nabídkových cen. Tímto způsobem mohou algoritmické strategie uměle ovlivnit cenotvorbu a reakce ostatních účastníků trhu, což vede k nesprávné interpretaci tržní situace. Důsledkem může být nejen manipulace s cenou aktiva, ale také zpomalení či narušení provozu obchodních platforem a burzovních systémů [24].

Příklad: Obchodník zadá nákupní objednávku na 1000 akcií a ihned ji zruší, mezitím trh zareaguje posunem ceny směrem nahoru.

Spoofing

Spoofing představuje sofistikovanou formu algoritmické manipulace s trhem, při níž dochází ke vkládání velkých falešných objednávek. Tyto příkazy jsou však zrušeny těsně před jejich realizací, což vede k umělému zkreslení tržní nabídky a poptávky. Tento postup generuje falešný tržní sentiment a ovlivňuje rozhodování ostatních účastníků trhu [24].

Příklad: Obchodník zadá prodejní objednávku na 500 akcií, ale před realizací ji zruší.

Layering

Layering je podobný **spoofingu**, avšak zahrnuje postupné umísťování více falešných objednávek na různé cenové úrovně. Tento přístup vytváří iluzi silné podpory *support* nebo rezistence *resistance* na trhu a nutí ostatní obchodníky reagovat, čímž manipulátor dosáhne výhodnější ceny pro své skutečné obchody [24].

Příklad: Pokud je aktuální tržní cena akcie 100, obchodník zadá více objednávek na úrovních 95, 97 a 98, aby vytvořil dojem rostoucí poptávky.

2.1.6 Modelování limitní knihy objednávek

Modelování limitní knihy objednávek představuje klíčovou oblast výzkumu, která zkoumá dynamiku interakcí mezi účastníky trhu a strukturou objednávek. Existují dva základní přístupy k modelování LOB, přičemž každý z nich zdůrazňuje odlišné aspekty této problematiky [8, 9]:

- **Ekonomický přístup:** Tento přístup vychází z teorie racionálních agentů, kteří se snaží maximalizovat svůj užitek v rámci daných tržních podmínek. Obchodní rozhodnutí jsou v tomto kontextu interpretována jako optimální řešení individuálních problémů maximalizace užitku. Příkladem mohou být modely, které analyzují strategické chování obchodníků na základě asymetrie informací nebo cenového očekávání (viz [25]).
- **Přístup nulové inteligence:** Alternativní metodika známá jako **zero-intelligence** přistupuje k modelování agentů statisticky a předpokládá náhodnost jejich rozhodnutí. Tento přístup je považován za zjednodušený, avšak dokáže zachytit mnohé důležité vlastnosti reálných trhů. Agenti zde generují objednávky řízené stochastickými procesy, což umožňuje studium dynamiky trhu bez nutnosti detailního modelování racionálního chování jednotlivců [26].

Každý přístup k modelování LOB vyžaduje určité předpoklady. Tradiční ekonomické modely předpokládají, že objednávky jsou generovány racionálními obchodníky, jejichž cílem je maximalizace užitečnosti v prostředí řízeném informacemi. Tento přístup byl však kritizován, neboť empirické studie často ukazují, že lidské chování se odchyluje od dokonalé racionality [26, 27]

Naopak přístup nulové inteligence se zaměřuje na stochastickou povahu toku objednávek, která je modelována pomocí specifických procesů, jejichž parametry mohou být funkcí dalších proměnných, například likvidity [28, 29]. Tento rámec

umožňuje konstrukci kvantitativních modelů, které jsou schopné generovat falsifikovatelné předpovědi bez dodatečných předpokladů o chování agentů.

Mezi těmito dvěma krajními přístupy — racionální maximalizací užitečnosti a nulovou inteligencí — se nachází široké spektrum modelů, které činí slabší předpoklady o chování obchodníků a jejich interakcích. Tyto modely často kombinují aspekty obou přístupů a přidávají složitější dynamiku toků objednávek, avšak za cenu vyšší náročnosti při jejich analýze.

2.2 Financial Information eXchange protokol

Protokol **Financial Information eXchange** (FIX) [30] představuje veřejně dostupný komunikační framework navržený s cílem standardizovat a optimalizovat elektronickou výměnu informací v oblasti finančních služeb. Tento protokol zajišťuje širokou kompatibilitu mezi různými formáty a typy komunikace mezi účastníky finančních trhů, včetně alokace obchodů, zadávání a modifikace objednávek a reportování realizovaných transakcí.

Správu a vývoj protokolu FIX koordinuje nezávislá organizace **FIX Trading Community**, která zajišťuje jeho pravidelnou aktualizaci a přizpůsobení aktuálním požadavkům trhu. FIX je široce implementován jak na straně institucionálních investorů, tak na straně makléřských a obchodních subjektů, což zahrnuje investiční fondy, investiční banky, burzy, makléřské firmy a elektronické obchodní systémy (ECN).

Díky své robustnosti a flexibilitě se FIX stal globálním standardem pro elektronickou komunikaci v oblasti předobchodní přípravy a samotného vykonávání obchodních transakcí. I když je tento protokol primárně zaměřen na obchodování s akciemi, umožňuje rovněž efektivní komunikaci v rámci obchodů s dluhopisy, deriváty a měnovými transakcemi. V tomto kontextu lze FIX považovat za standard pro komunikaci na front office úrovni, obdobně jako je protokol SWIFT [31] etablovaným standardem pro back office operace.

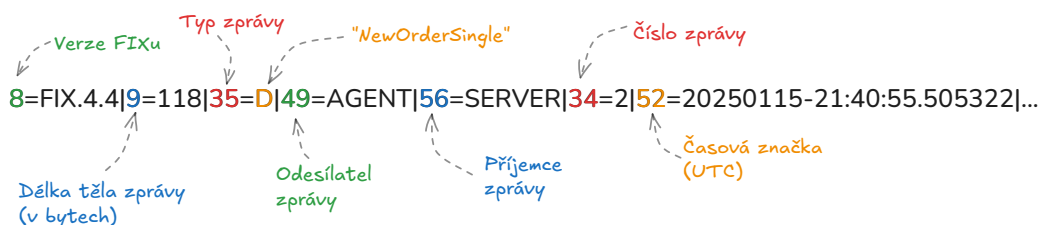
2.2.1 Struktura zpráv FIX

Zprávy protokolu FIX jsou kódovány ve formátu klíč=hodnota (tag=value), přičemž jednotlivé položky jsou odděleny znakem SOH (ASCII kód 0x01). Tento kompaktní formát umožňuje samo-popisné zprávy, které lze snadno dekodovat a interpretovat jak lidsky, tak strojově.

Každá zpráva v protokolu FIX má pevně definovanou strukturu, která se skládá z následujících částí:

- **Hlavička (Header):** Obsahuje klíčové informace potřebné k identifikaci a zpracování zprávy, například:
 - **BeginString** (klíč 8): Verze protokolu FIX.
 - **BodyLength** (klíč 9): Udává délku těla zprávy (Body) v bytech.
 - **MsgType** (klíč 35): Identifikuje typ zprávy (například objednávka, potvrzení obchodu apod.).
- **Tělo zprávy (Body):** Obsahuje podrobné informace týkající se dané transakce nebo komunikace. Každé pole je reprezentováno dvojicí klíč=hodnota, kde klíč je číselný identifikátor specifikující význam pole. Struktura a obsah této části zprávy jsou značně variabilní v závislosti na typu zprávy. Mezi často používaná pole patří:
 - **Klíč 11 (ClOrdID):** Unikátní identifikátor objednávky.
 - **Klíč 54 (Side):** Strana obchodu, například nákup (1) nebo prodej (2).
 - **Klíč 38 (OrderQty):** Počet jednotek nebo objem obchodované položky.
 - **Klíč 44 (Price):** Cena obchodované položky.
- **Patička (Trailer):** Označuje konec zprávy a obsahuje kontrolní mechanismus zajišťující integritu přenášených dat:
 - **Klíč 10 (Checksum):** Kontrolní součet vypočítaný z celé zprávy pro ověření její správnosti.

Pro lepší ilustraci struktury a fungování zpráv v protokolu FIX následuje Obrázek 2.1 s ukázkou zprávy ve výše zmíněném formátu.

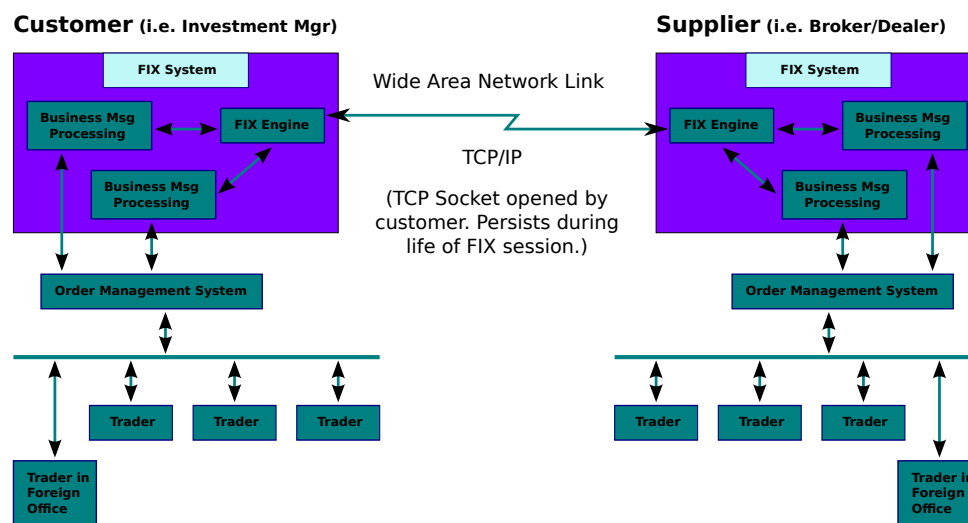


Obrázek 2.1: Ukázková zpráva ve FIX protokolu

FIX protokol pracuje na aplikační vrstvě a využívá protokol **TCP** na vrstvě transportní. Komunikace je organizována na principu klient-server, kde klient

(označovaný jako „Initiator“) zahajuje spojení a server (označovaný jako „Acceptor“) jej přijímá.

Níže je uveden Diagram 2.2 zobrazující strukturu výměny zpráv v rámci protokolu *FIX* mezi **kupující stranou** (zákazníkem) a **prodávající stranou** (dodavatelem/makléřem).



Obrázek 2.2: Diagram znázorňující systém FIX (převzato z [32])

2.3 Agentové modelování

Agentové modelování (AM) je metodický přístup, který umožňuje simulaci a analýzu složitých systémů prostřednictvím interakcí autonomních agentů. Tento přístup se používá k modelování dynamických systémů, kde jednotliví agenti samostatně reagují na změny v prostředí a vzájemně interagují mezi sebou. V kontextu burzovních simulátorů představuje agent model obchodníka, investora nebo jiného účastníka trhu, který se rozhoduje na základě určitých pravidel a strategií.

2.3.1 Definice agenta

Agent je autonomní entita, která vykonává akce na základě předprogramovaných vlastností a pravidel, přičemž není dále řízena externími příkazy. Je schopen interagovat jak s ostatními **agenty**, tak s **prostředím**, což umožňuje jeho **adaptaci** na změny v okolí. **Interakce** mezi agenty jsou většinou **lokální**, tj. probíhají pouze

při vzájemném přiblížení nebo prostřednictvím **informační vazby**, jako je například propojení v rámci komunikační sítě. Agenti mohou mít integrovaný **adaptační mechanismus**, který jim umožňuje přizpůsobit své chování v reakci na změny podmínek nebo na zkušenosti z předchozích interakcí [33].

2.3.2 Typy agentů

V agentovém modelování se často rozlišují různé typy agentů podle jejich chování, schopnosti adaptace a rozhodovacích procesů. Dle [34] lze rozlišit čtyři základní typy agentů, přičemž hranice mezi nimi nejsou zcela ostré:

- **Reaktivní (reflexní) agenti** (*Reactive/Reflex Agents*): Tito agenti vykonávají akce na základě aktuálních podnětů z prostředí podle předem definovaných pravidel. Jejich chování je deterministické a přímo závislé na okamžitých vstupech, bez jakékoliv interní strategie nebo plánování.
- **Agenti zaměřeni na dosahování cílů** (*Goal-Based Agents*): Agenti této kategorie se od reaktivních agentů liší tím, že nejen reagují na podněty, ale aktivně usilují o dosažení specifických cílů. Pro dosahování těchto cílů vyhodnocují možné akce a volí tu, která je nejefektivnější pro dosažení požadovaného stavu.
- **Agenti maximalizující svůj užitek** (*Utility-Based Agents*): Tento typ agentů vyhodnocuje různé možné akce a volí tu, která maximalizuje jejich **užitek**. Užitek je měřítko, které agentovi umožňuje kvantifikovat kvalitu výsledku každé akce. Tito agenti tedy provádějí optimalizační rozhodování na základě výpočtu, jaká akce přinese nejlepší výsledek.
- **Adaptivní (učící se) agenti** (*Adaptive/Learning Agents*): Agenti této kategorie mají schopnost se adaptivně přizpůsobovat změnám v prostředí. Na základě minulých interakcí s prostředím upravují své rozhodovací procesy tak, aby zlepšili své chování a dosahovali lepších výsledků v dlouhodobém horizontu.

2.3.3 Vlastnosti agentů

Každý agent v systému může vykazovat různé vlastnosti, které jej odlišují od běžných řídicích systémů a které určují jeho chování a schopnost interagovat s okolním prostředím. Mezi klíčové vlastnosti agentů patří [35]:

- **Situovanost** (*Situatedness*): Každý agent je úzce spjat se svým prostředím, se kterým interaguje prostřednictvím senzorů a ovlivňuje ho pomocí aktuátorů.

Veškeré vstupy získává přímou interakcí s okolím a jeho činnost se neomezuje pouze na poskytování rozhodovací podpory jako u expertních systémů, které působí skrze prostředníka.

- **Autonomie** (*Autonomy*): Agent disponuje schopností nezávislého rozhodování a volby akcí bez zásahu externích entit, ať už ze strany jiných agentů v systému nebo lidského operátora. Tato vlastnost zajišťuje ochranu vnitřních stavů agenta před vnějšími vlivy a chrání jej před nestabilitou způsobenou externími zásahy.
- **Reaktivita** (*Responsiveness*): Schopnost agenta vnímat stav prostředí a adekvátně na něj reagovat v reálném čase. Tato vlastnost je klíčová zejména v aplikacích, kde je nutná okamžitá adaptace na změny v prostředí.
- **Inferenční schopnost** (*Inferential Capability*): Agent je schopen pracovat s abstraktními cíli, například vyvozovat závěry na základě dostupných informací generalizací relevantních poznatků. Tato schopnost mu umožňuje efektivní rozhodování i v situacích charakterizovaných neúplností, neurčitostí či nedostupností relevantních dat.
- **Proaktivita** (*Pro-Activeness*): Chování agentů není omezeno pouze na reakce na vnější podněty, ale zahrnuje i aktivní vyhledávání příležitostí k dosažení cílů. Namísto pasivního přizpůsobování změnám v prostředí dochází k cílené iniciativě směřující k optimalizaci budoucího vývoje systému.
- **Sociální chování** (*Social Behaviour*): Navzdory své autonomii musí agent disponovat schopností interakce s dalšími entitami, pokud je to nutné pro dosažení stanovených cílů. To zahrnuje nejen přímou kooperaci s jinými agenty v rámci některých multiagentních systémů (MAS), ale i výměnu informací s člověkem či jinými řídicími mechanismy, čímž je umožněna kolektivní adaptace a koordinace.

Kromě výše uvedených vlastností mohou agenti disponovat dalšími atributy, jako je **mobilita** (schopnost přesouvat se mezi různými prostředími), **časová kontinuita** (perzistence existence v čase) nebo **kolaborativní chování** (spolupráce s dalšími entitami). Na základě míry naplnění těchto vlastností lze agenty dále kategorizovat jako **slabé** či **silné**, přičemž silní agenti vykazují pokročilejší schopnosti v rámci uvedených dimenzí [35].

2.3.4 Multiagentní systémy (MAS)

Multiagentní systém (*Multi-Agent System*, MAS) představuje distribuovanou výpočetní architekturu, v níž soubor autonomních agentů operuje v sdíleném prostředí s cílem dosáhnout společného nebo individuálního cíle. Interakce mezi agenty mohou být založeny na kooperaci či konkurenci, přičemž sdílení informací mezi jednotlivými entitami může být buď zcela otevřené, nebo naopak omezené [35].

Multiagentní systémy nacházejí široké uplatnění v různých aplikačních doménách, a to především díky řadě výhod, které tato technologie nabízí. Mezi klíčové přínosy MAS v rozsáhlých systémech patří [35]:

- **Výpočetní efektivita** — paralelizace a asynchronní zpracování zvyšují výkonnost systému.
- **Robustnost** — decentralizace umožňuje pokračování operací i při selhání jednotlivých agentů.
- **Škálovatelnost** — snadná expanze systému přidáváním nových agentů.
- **Ekonomická efektivita** — distribuované řízení snižuje náklady oproti centralizovaným strukturám.
- **Modularita** — agenty lze snadno upravovat, nahrazovat či integrovat do jiných systémů.

Přestože multiagentní systémy (MAS) přinášejí řadu výhod, jejich implementace s sebou nese i specifické výzvy. Jednou z nich je potřeba predikce chování ostatních agentů, protože jejich akce ovlivňují nejen jejich vlastní stav, ale i okolí. Tento problém se dále zhoršuje v dynamických prostředích, kde je nutné rozlišovat změny způsobené agenty od externích vlivů. Další problém představuje omezená percepce agentů, která může vést k suboptimálním rozhodnutím, což komplikuje dosažení globálního optima. Konflikty mezi agenty, vznikající z nedostatku globálního přehledu, vyžadují koordinaci při zachování jejich autonomie [35].

MAS proto nejsou optimální volbou pro každou aplikaci. Jejich nasazení je nejpřínosnější v oblastech, kde dochází k interakcím mezi různými subjekty s rozdílnými či společnými cíli, jako jsou distribuované řízení, obchodní systémy nebo simulace socioekonomických procesů.

2.3.5 Klasifikace multiagentních systémů

Klasifikace **MAS** může být založena na různých charakteristikách, jako jsou **architektura**, **učení**, **komunikace** nebo **koordinace**. Lze proto rozlišovat několik základních typů **MAS** podle těchto vlastností [35].

Interní architektura

MAS lze rozdělit na **homogenní** a **heterogenní** struktury podle vnitřní architektury jednotlivých agentů [35].

- **Homogenní struktura:** Všichni agenti v systému mají stejnou vnitřní architekturu. Rozdíly mezi agenty spočívají pouze v jejich fyzické lokalizaci a oblasti prostředí, ve které provádějí akce.
- **Heterogenní struktura:** Agenti se liší schopnostmi, strukturou a funkcemi. To umožňuje modelovat složitější reálné aplikace, kde mohou agenti mít různé lokální cíle, které mohou být v konfliktu s cíli jiných agentů.

2.3.6 Komunikace v multiagentních systémech

Komunikace je klíčová pro koordinaci agentů v **multiagentních systémech (MAS)**. Nadměrná nebo redundantní komunikace zvyšuje výpočetní náklady a latenci, zatímco omezená komunikace může vést k suboptimálním rozhodnutím. Hlavní komunikační modely zahrnují **lokální komunikaci** a **blackboardovou architekturu** [35].

Lokální komunikace

Lokální komunikace probíhá přímým předáváním zpráv mezi agenty bez centrálního úložiště či prostředníka. Tento model umožňuje **bidirekcionální** tok informací, čímž podporuje distribuovanou architekturu snižující riziko selhání centrálních entit. Absence centrálního koordinátora vede ke zlepšené škálovatelnosti, avšak zároveň vyžaduje robustní mechanismy pro zajištění konzistence sdílených dat.

Komunikace prostřednictvím blackboardů

Alternativním přístupem k výměně informací je **komunikace prostřednictvím blackboardů**, kde agenti sdílí centrální **datové úložiště**. Toto úložiště umožňuje

efektivní **asynchronní přístup** k datům a podporuje jejich řízenou distribuci. Přístup k datům je řízen pomocí **kontrolní vrstvy**, která může definovat přístupová práva a notifikace o dostupnosti relevantních dat. Tento model umožňuje efektivní sdílení znalostí, avšak představuje riziko selhání centrálního bodu, které lze zmírnit redundancí.

2.3.7 Agentové modelování v burzovních simulátorech

Agentové modelování v burzovních simulátorech se zaměřuje na simulaci interakcí mezi různými typy agentů, kteří reprezentují účastníky finančních trhů, jako jsou investoři, spekulanti, makléři a regulátoři. Cílem je analyzovat, jak tyto interakce ovlivňují tržní dynamiku a rozhodovací procesy v podmínkách vysoké nejistoty a komplexnosti. Agentové modely využívají algoritmy, které umožňují agentům přijímat rozhodnutí na základě historických dat, tržních trendů a behaviorálních vzorců, čímž se snaží napodobit reálné obchodní rozhodování. V simulacích mohou být agenti vybaveni různými strategiemi, od jednoduchých pravidel až po sofistikované prediktivní modely, které zahrnují faktory jako risk management a analýzu sentimentu trhu. Tento přístup umožňuje testovat různé scénáře a analyzovat dopady specifických politik nebo krizí na trhy, což je klíčové pro vývoj efektivních regulačních rámců a obchodních strategií.

Významnou součástí agentového modelování burzovních simulací jsou **multiagentní systémy**, v nichž agenti nejen autonomně rozhodují, ale zároveň vzájemně interagují prostřednictvím sdíleného burzovního prostředí. Tržní dynamika v těchto systémech nevzniká řízením z centrální úrovně, nýbrž emergentně jako důsledek souhry individuálních strategií jednotlivých agentů. Ti mohou být modelováni s různou úrovní komplexity – od **zero-intelligence traders**, kteří zadávají příkazy náhodně, až po pokročilé prediktivní agenty využívající algoritmy strojového učení. Multiagentní přístup umožňuje realistickou simulaci komplexních tržních jevů, jako je vznik cenových bublin, nedostatek likvidity či přenos tržních signálů v rámci finančního systému. Multiagentní modelování tak představuje klíčový nástroj pro analýzu adaptivního chování tržních účastníků a jejich interakcí v prostředí s omezenou dostupností informací.

Implementovaná řešení

3

Tato kapitola se věnuje popisu struktury a fungování implementovaného simulátoru burzovního obchodování. Simulátor je realizován jako vícevrstvá aplikace, v níž jednotlivé komponenty plní specifické úlohy, jako je správa tržních dat, řízení obchodních agentů či zprostředkování komunikace mezi účastníky simulace. Představeny jsou jak základní stavební části systému, tak principy jejich vzájemné interakce, včetně způsobu, jakým jsou v rámci simulace modelováni jednotliví účastníci trhu.

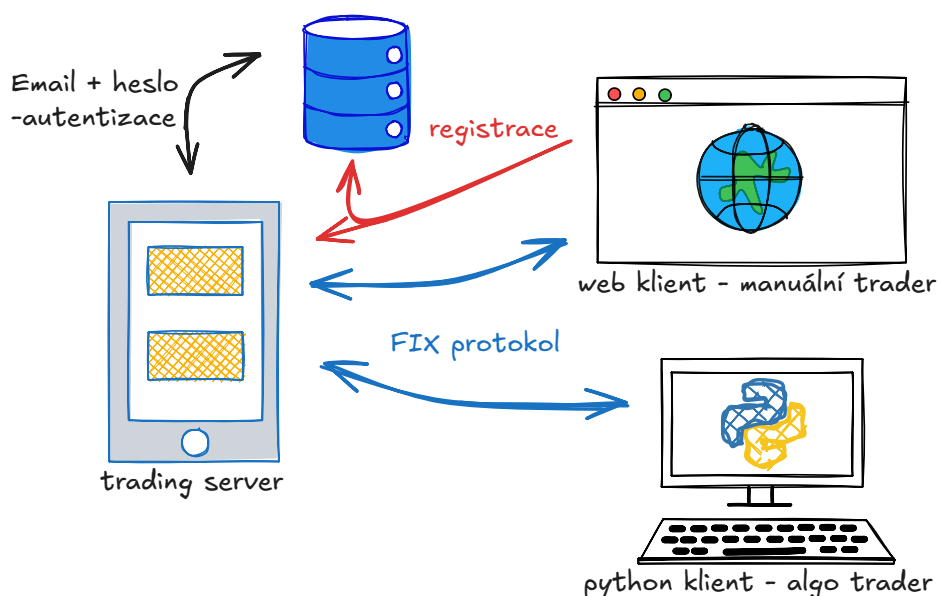
Podrobnosti o instalaci, konfiguraci a spuštění simulátoru jsou uvedeny v uživatelské příručce, která je součástí příloh této práce (Příloha A).

3.1 Architektura simulátoru

Simulátor je navržen jako *klient-server* aplikace, implementovaná v jazyce **Python**. Základní architektura systému vychází z oddělení aplikační logiky na straně serveru a rozhraní pro interakci s uživateli na straně klienta. Klientská část umožňuje jak automatizované obchodování prostřednictvím softwarových agentů, tak manuální zadávání příkazů pomocí webového uživatelského rozhraní.

Serverová komponenta spravuje stav trhu, zpracovává příchozí objednávky a řídí párování příkazů v rámci limitní knihy. Součástí systému je také databáze uživatelů, která umožňuje uchovávat a spravovat informace o jednotlivých účastnících simulace.

Zvolená architektura poskytuje flexibilní rámec pro experimentování s různými typy obchodních strategií a umožňuje analyzovat jejich dopady v kontrolovaném, avšak realisticky strukturovaném tržním prostředí. Obecná struktura aplikace je zobrazena na Obrázku 3.1. Jednotlivé komponenty systému a jejich implementační aspekty jsou pak detailněji popsány v následujících sekcích.



Obrázek 3.1: Schéma architektury simulátoru

3.2 Obchodní server

Serverová část simulátoru je implementována pomocí frameworku **Tornado**¹, který byl zvolen pro jeho schopnost efektivně zpracovávat velké množství současných připojení s nízkou latencí. Díky asynchronnímu zpracování požadavků umožňuje **Tornado** obsluhovat velké množství simultánních požadavků bez blokování vláken, což je klíčové pro simulaci burzovního obchodování. Další výhodou **Tornado** je podpora **WebSocketů**, což umožňuje efektivní obousměrnou komunikaci v reálném čase mezi serverem a klienty.

Pro komunikaci mezi klientem a serverem byl implementován vlastní komunikační protokol, který funguje nad HTTP (respektive nad WebSokety) a definuje jednotný formát pro serializaci a deserializaci zpráv. Tento protokol je navržen jako obecné rozhraní, od kterého mohou být odvozeny konkrétní implementace. V případě simulátoru byla vytvořena implementace založená na upravené verzi standardu FIX (viz Kapitola 2.2), přizpůsobené specifickým potřebám simulovaného prostředí. Tato varianta protokolu zachovává vybrané prvky struktury a principů FIXu, zejména použití standardizovaných polí s číselnými tagy, identifikaci stran komunikace a formát hlavičky zpráv, přičemž jednotlivé typy zpráv jsou enkódovány pomocí knihovny **simplefix**². Tímto způsobem je zajištěna srozumitelná a efektivní výměna informací mezi

¹<https://www.tornadoweb.org/en/stable/>

²<https://pypi.org/project/simplefix/>

komponentami systému v reálném čase. Podrobné informace o konkrétní implementaci tohoto protokolu lze nalézt v Příloze C.

Celková architektura serverové části simulátoru byla navržena s cílem zajistit efektivní správu tržních dat a obchodních objednávek v reálném čase. Hlavním úkolem serveru je koordinace interakcí mezi účastníky simulace, správa jejich obchodních objednávek a zajištění správného párování objednávek v souladu s předem definovanými pravidly.

3.2.1 Limitní kniha objednávek

Základní komponentou serverové části simulátoru je třída reprezentující limitní knihu objednávek, která spravuje aktuální stav objednávkové knihy a poskytuje metody pro manipulaci s objednávkami, včetně jejich přidávání, mazání a modifikace. Dále uchovává informace o uživateli, kteří objednávky zadali, včetně jejich vlastního objemu a celkového výdělku na daném aktivu.

Objednávková kniha je rozdělena na dvě části: **Bids** (kupní nabídky) a **Asks** (prodejní nabídky), přičemž každá strana je implementována pomocí `SortedDict` z knihovny `sortedcontainers`³. Tato struktura umožňuje efektivní vkládání, hledání a mazání objednávek podle ceny v logaritmickém čase $O(\log M)$, kde M je počet cenových úrovní. Každá cenová úroveň je dále reprezentována pomocí deque, tedy obousměrné fronty, která zajišťuje rychlé přidávání a odebrání objednávek na začátku i konci seznamu v konstantním čase $O(1)$. Metody jako `peekitem()` navíc umožňují okamžitý přístup k nejlepší nabídce a poptávce, rovněž v konstantním čase $O(1)$, což je klíčové pro rychlé tržní operace. Kromě toho je pro účely rychlé identifikace a modifikace kvantity objednávek udržována mapa objednávek, což umožňuje přímý přístup bez nutnosti procházet knihu.

Alternativním přístupem by bylo použití **binárního stromu**, kde každý uzel reprezentuje cenovou úroveň s dvojité propojeným seznamem objednávek. Každá strana knihy (Bids/Asks) má vlastní strom a cenové úrovně i objednávky jsou zároveň uloženy v mapách pro rychlý přístup. Tato struktura umožňuje efektivní operace: přidání první objednávky na nové úrovni v $O(\log M)$, ostatní operace (vložení, zrušení, provedení, přístup k nejlepší ceně či objemu) v $O(1)$. Nevýhodou je vyšší složitost implementace a nutnost udržovat strom vyvážený, což může být problematické v dynamickém prostředí s častými změnami, kdy by rebalancování stromu mohlo negativně ovlivnit výkon.

³<https://pypi.org/project/sortedcontainers/>

3.2.2 Vypořádávání objednávek

Další klíčovou komponentou systému je párovací algoritmus typu **Price-Time Priority**, který zajišťuje deterministické a transparentní zpracování příchozích objednávek. Tento algoritmus dává přednost objednávkám s výhodnější cenou (vyšší Bid nebo nižší Ask) a v rámci stejné cenové úrovně rozhoduje podle času vložení objednávky (**FIFO**).

Při zpracování nové objednávky dochází ke kontrole, zda je možné provést obchod s nejlepší dostupnou protinabídkou v knize. Pokud je obchod uskutečnitelný (např. kupní cena je rovna nebo vyšší než nejlepší prodejní cena), algoritmus postupně páruje novou objednávku s objednávkami na dané cenové úrovni v pořadí, v jakém byly zadány. Obchody se provádějí po částech, dokud není příchozí objednávka plně realizována nebo dokud nedojdou protinabídky splňující cenové podmínky. V případě částečného naplnění objednávky je zbylé množství uloženo do objednávkové knihy.

3.2.3 Integrovaná správa produktů a uživatelských financí

Simulační server je navržen s důrazem na modularitu a škálovatelnost, což umožňuje současnou správu více obchodovaných instrumentů. Každý produkt je reprezentován samostatnou instancí **limitní knihy objednávek**, která autonomně spravuje veškeré objednávky a transakce související s daným aktivem. Tento přístup eliminuje provázanost mezi jednotlivými produkty a umožňuje paralelní zpracování událostí na úrovni jednotlivých trhů.

Každá kniha rovněž odděleně udržuje stav uživatelských účtů specifických pro daný produkt, tj. aktuální množství vlastněného aktiva a celkový výnos z obchodování. Tím je zajištěna vysoká granularita dat a přesná evidence výsledků v rámci jednotlivých trhů. Naopak **vstupní rozpočet uživatelů** je sdílen napříč všemi produkty a představuje celkový objem prostředků, který má uživatel k dispozici pro obchodování. Tato architektura umožňuje realistickou simulaci prostředí s více produkty, včetně alokace kapitálu mezi jednotlivé z nich, a podporuje testování strategií zaměřených na správu portfolia tvořeného více aktivy.

V rámci implementace je také zavedeno historické ukládání limitních knih včetně detailních informací o uživatelských pozicích. Tento proces je realizován využitím formátu `.pickle`, který umožňuje efektivní persistentní uchovávání kompletního stavu objednávkové knihy. Ukládání dat v tomto formátu zajišťuje, že při opětovném spuštění serveru je možné rekonstruovat poslední známý stav limitní

knihy jako počáteční bod pro pokračování obchodování, čímž se minimalizuje riziko ztráty kritických obchodních dat. Navíc tento mechanismus umožňuje využití historických dat pro generování komplexních analytických reportů o výkonnosti obchodních strategií, což uživatelům poskytuje cenné nástroje pro kvantitativní hodnocení efektivity jejich obchodních rozhodnutí a následnou optimalizaci jejich obchodních strategií.

3.2.4 Struktura serveru

Server je navržen tak, aby poskytoval bezpečné, efektivní a spolehlivé prostředí pro obchodování v reálném čase. Jeho hlavní funkcí je zpracování a párování obchodních objednávek v souladu s předem definovanými pravidly, přičemž klade důraz na validitu a správnost každé objednávky. Před umožněním přístupu k obchodování server ověřuje identitu uživatele, přičemž pouze registrovaní uživatelé mají oprávnění obchodovat. Při každém pokusu o zadání objednávky je aplikována sofistikovaná validace, která zaručuje její správnost a proveditelnost.

Po úspěšném zpracování objednávky, ať už byla spárována s jinou objednávkou, nebo přidána do objednávkové knihy, server aplikuje příslušné transakční poplatky. Tyto poplatky se skládají z pevné částky a procentuálního podílu z hodnoty transakce. Po jejich aplikaci dochází k odpovídajícímu upravení uživatelského účtu.

Hlavní funkční oblasti serveru zahrnují:

- **Trading Handler:** Tato část serveru je zodpovědná za správu obchodních objednávek. Umožňuje registraci uživatelů, správu objednávkových knih, vyhledávání a párování objednávek v rámci jednotlivých produktů a také výpočet transakčních poplatků.
- **Quoting Handler:** Tento handler poskytuje uživatelům informace o aktuálním stavu objednávkových knih, historii objednávek a stavu jejich vlastních obchodních objednávek. Umožňuje také generování historických reportů o stavu objednávek v minulosti.

Tyto funkce jsou dostupné na vyžádání uživatele prostřednictvím specifických požadavků.

Komunikace mezi serverem a klientem probíhá dále pomocí WebSocket připojení, což umožňuje asynchronní výměnu dat a okamžité aktualizace. Při každé změně v objednávkové knize, například po provedení obchodu, server okamžitě „broadcastuje“ aktuální stav objednávkové knihy všem připojeným uživatelům, čímž zajišťuje, že všechny strany mají přístup k aktuálním informacím pro následná obchodní rozhodnutí.

3.2.5 Databáze uživatelů

Součástí implementovaného řešení je i databázový systém, který slouží k registraci a správě uživatelských účtů. Tento systém využívá **SQLite** jako lehkou a efektivní databázovou technologii pro ukládání informací o uživateli. Databáze obsahuje jednu tabulku **users**, která je strukturována do tří sloupců: **id** (primární klíč), **email** (unikátní identifikátor uživatele) a **password** (heslo uživatele, které je uloženo v zašifrované podobě pomocí **bcrypt**⁴).

3.3 Klientská část aplikace

Klientská část aplikace slouží jako rozhraní mezi uživatelem a serverem, umožňuje komunikaci v reálném čase a interakci s obchodním systémem. Zajišťuje správu obchodních objednávek, získávání tržních dat a stabilní připojení k serveru. Komunikace probíhá prostřednictvím asynchronních metod a API volání.

3.3.1 Hlavní komponenty klientské aplikace

Klientská aplikace se skládá z několika klíčových tříd, které zajišťují efektivní komunikaci a interakci s obchodním serverem. Struktura těchto tříd a jejich vzájemné vztahy jsou znázorněny v následujícím UML diagramu na Obrázku 3.2.

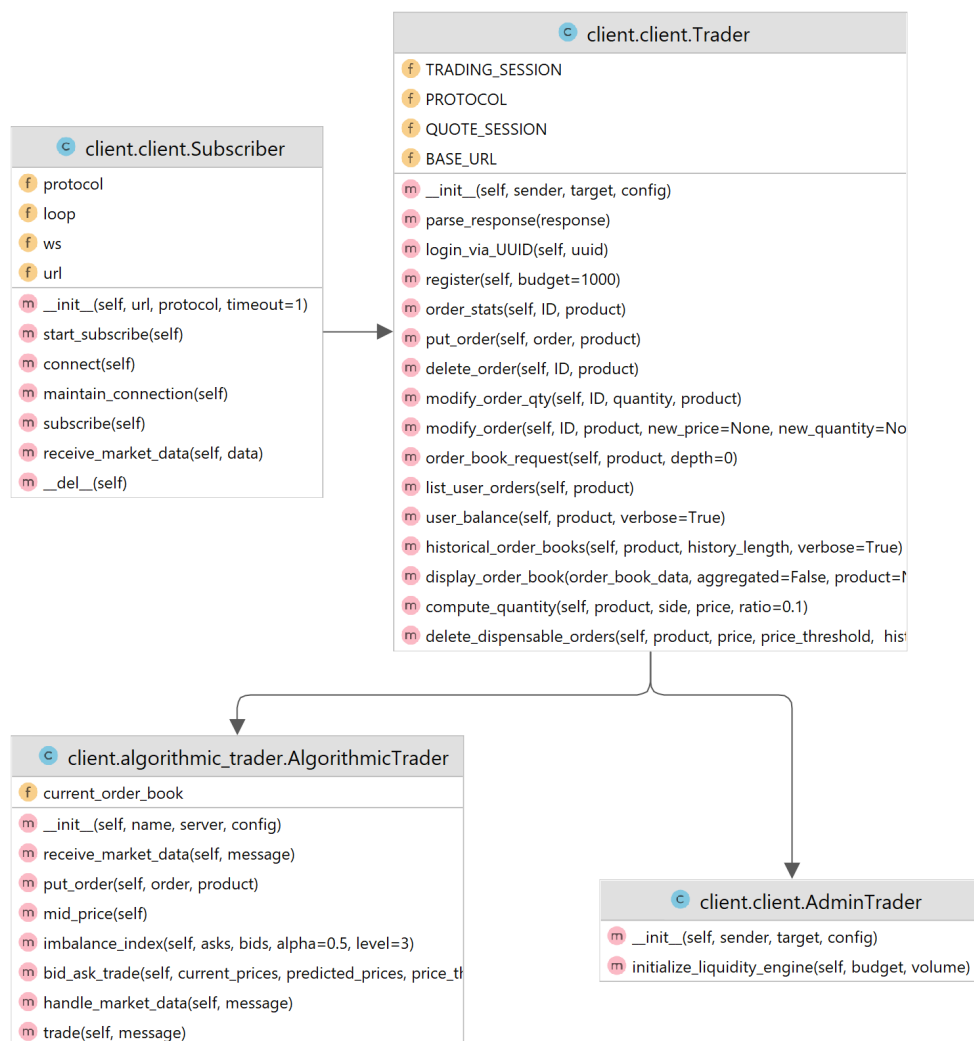
Subscriber

Třída **Subscriber** zajišťuje připojení k obchodnímu serveru pomocí **WebSocketů** a pravidelnou synchronizaci tržních dat. Udržuje stabilní připojení a přijatá data předává metodě **receive_market_data**, kterou musí implementovat potomci této třídy. Tato metoda umožňuje potomkům definovat vlastní způsob zpracování tržních dat.

Trader

Třída **Trader** představuje rozšíření základní funkcionality třídy **Subscriber** o možnost aktivního zasílání požadavků na obchodní server pomocí HTTP rozhraní (REST API). Zatímco **Subscriber** zajišťuje odběr tržních dat přes **WebSocket**, **Trader** umožňuje obousměrnou komunikaci, čímž vytváří plnohodnotného klienta pro algoritmické i manuálně řízené obchodování.

⁴<https://pypi.org/project/bcrypt/>



Obrázek 3.2: UML diagram klientské aplikace

Třída poskytuje nástroje pro správu celého obchodního cyklu – od registrace uživatele, přes zadávání, modifikaci a rušení objednávek, až po dotazy na aktuální i historický stav trhu a zůstatků. Součástí jsou také pokročilé utility pro podporu rozhodování, jako je automatický výpočet vhodné velikosti objednávky na základě aktuálních tržních podmínek nebo detekce a odstranění zastaralých či nevýhodných aktivních objednávek.

AlgorithmicTrader

Třída `AlgorithmicTrader` představuje základní abstraktní třídu pro konstrukci algoritmických obchodních agentů. Navazuje na funkcionalitu poskytovanou třídou `Trader`, kterou rozšiřuje o rámec pro autonomní rozhodování a reakci na aktuální tržní podmínky. Slouží jako architektonický vzor pro návrh obchodních strategií, které využívají online data a automaticky generují obchodní pokyny na základě předem definované logiky nebo adaptivních modelů.

Na rozdíl od obecné třídy `Trader`, která poskytuje základní nástroje pro interakci s obchodním serverem, `AlgorithmicTrader` zavádí mechanismus pro zpracování aktuálních tržních dat a řízení obchodní logiky. Každý potomek této třídy musí definovat dvě klíčové metody:

- `handle_market_data(message)` – zpracování příchozích tržních dat,
- `trade(message)` – generování a odesílání obchodních příkazů.

Interně třída spravuje aktuální stav knihy objednávek a poskytuje nástroje pro odhadování klíčových metrik trhu, jako je například **střední cena** nebo **imbalance index** mezi nabídkovou a poptávkovou stranou. Tyto metriky lze následně využívat jako vstupy pro optimalizaci obchodních rozhodnutí.

Třída `AlgorithmicTrader` tvoří základ pro pokročilé obchodní strategie, které operují autonomně a reagují v reálném čase na změny limitní knihy objednávek. Je navržena tak, aby byla snadno rozšiřitelná jak pro jednoduché pravidlové přístupy, tak i pro komplexní strategie založené na statistických modelech nebo strojovém učení.

AdminTrader

Třída `AdminTrader` představuje speciální variantu obchodního klienta, jejímž primárním účelem je správa a konfigurace obchodního prostředí z pohledu administrátora nebo správce trhu. Slouží jako klientská komponenta pro inicializaci a správu specializovaných agentů, kteří operují na straně serveru a zajišťují udržování stabilní tržní dynamiky. Umožňuje definovat počáteční rozpočet a objem aktiv, s nimiž mohou agenti okamžitě pracovat a emitovat je do trhu, čímž pomáhá udržovat rovnováhu na trhu.

Tato funkcionalita je klíčová pro simulované burzy nebo výzkumné platformy, kde je potřeba řízeně zajistit základní úroveň likvidity v prostředí s omezeným počtem účastníků. `AdminTrader` tak umožňuje zavádět a kontrolovat autonomní tržní

struktury, které fungují nezávisle na běžných klientech, a tvoří základ pro realistické testování obchodních strategií.

Obchodní agenti

V následujících kapitolách je představen návrh a implementace typů obchodních agentů, kteří tvoří nedílnou součást klientské vrstvy burzovního simulátoru. Patří mezi ně administrativní agenti pro správu trhu, algoritmičtí obchodníci a grafické uživatelské rozhraní určené pro manuální obchodování.

Navržení algoritmičtí agenti pokrývají různé koncepty a techniky – od pravidlových systémů, přes statistické metody až po využití metod strojového učení. Tyto přístupy byly zvoleny s cílem ilustrovat, z jakých oblastí lze při návrhu autonomního obchodního chování vycházet. Výsledkem je rozšiřitelný rámec, který slouží nejen jako testovací prostředí pro tržní mechanismy, ale také jako základ pro další vývoj pokročilejších obchodních modelů či pro výuku algoritmického obchodování.

3.3.2 Administrátorští agenti

Administrátorští agenti tvoří specifickou třídu entit navržených pro řízené formování a stabilizaci tržního prostředí. Společným základem těchto agentů je dědičnost od třídy `AdminTrader`, která poskytuje rozhraní pro jejich inicializaci, konfiguraci a správu ze strany serverové infrastruktury.

Mezi tyto agenty patří **tvůrce trhu** a **generátor likvidity**, jejichž primárním cílem je zajistit inicializaci knihy objednávek po spuštění nové obchodní relace a udržovat plynulou tržní aktivitu prostřednictvím strategického vkládání limitních příkazů. Tímto způsobem zajišťují lepší dostupnost obchodních protistran a posilují stabilitu tržního prostředí, což napomáhá plynulému fungování simulovaného trhu.

Tvůrce trhu (Market maker)

Tvůrce trhu je specializovaný obchodní agent navržený pro správu a udržování likvidity na trhu. Jeho hlavní funkcionalitou je inicializace a umístování objednávek do trhu tak, aby zajišťoval stabilitu a likviditu tržního prostředí.

Inicializace trhu je prováděna na základě **exponenciálního rozdělení**, které je aplikováno jak na **ceny**, tak na **objemy** objednávek v limitní knize. Ceny pro nabídku a poptávku jsou generovány exponenciálně kolem počáteční tržní ceny, přičemž většina objednávek je umístěna v bezprostřední blízkosti této ceny,

zatímco pravděpodobnost výskytu objednávek na vzdálenějších cenových úrovních klesá. Tento přístup odráží přirozené tržní chování, kdy je většina likvidity soustředěna v okolí střední hodnoty trhu, zatímco extrémní cenové úrovně mají nižší pravděpodobnost realizace. Objemy objednávek jsou rovněž generovány na základě exponenciálního rozdělení, což umožňuje přirozené rozložení likvidity mezi různé cenové úrovně.

Další funkcionalitou je automatické doplňování likvidity na vyprázdněnou stranu obchodní knihy. Pokud dojde k vyprázdnění objednávkové knihy na jedné straně trhu (poptávka nebo nabídka), systém automaticky generuje nové objednávky na základě aktuální tržní situace. Ceny těchto objednávek jsou určovány pomocí **dynamického spreadu**, který je adaptivně nastaven na základě **volatility historických středních cen**.

Volatilita zde slouží jako měřítko tržní nejistoty a je kvantifikována pomocí směrodatné odchylky cen v rámci klouzavého časového okna délky k . Výpočet směrodatné odchylky v čase n je dán vztahem:

$$\sigma_n = \sqrt{\frac{1}{k} \sum_{i=n-k+1}^n \left(P_i - \frac{1}{k} \sum_{j=n-k+1}^n P_j \right)^2} \quad (3.1)$$

kde P_i (respektive P_j) označuje historické střední ceny. Tento přístup zajišťuje, že nové objednávky jsou umístěny v souladu s aktuálním tržním prostředím a přispívá k udržení rovnováhy mezi nabídkou a poptávkou, čímž podporuje stabilitu a likviditu trhu.

Třída MarketMaker je navržena tak, aby byla vysoce konfigurovatelná. Uživatel má možnost upravit klíčové parametry, jako jsou počáteční ceny, objemy likvidity nebo volatilita, což umožňuje přizpůsobení chování agenta různým obchodním podmínkám. Tato flexibilita poskytuje širokou škálu možností pro ladění tržního prostředí a optimalizaci výkonu podle specifických požadavků.

Generátor likvidity (Liquidity generator)

Generátor likvidity představuje agenta, jehož hlavním cílem je průběžné doplňování syntetické likvidity do trhu. Pracuje nezávisle na konkrétní tržní strategii a jeho úkolem je udržovat aktivitu v obchodní knize prostřednictvím náhodně generovaných objednávek, které napomáhají simulovat realistické tržní prostředí. Z formálního hlediska lze i tohoto agenta klasifikovat jako tvůrce trhu. S ohledem na specifický způsob generování a zpracování objednávek je však v rámci této práce popsán a implementován samostatně.

Objednávky jsou generovány s náhodně volenou stranou trhu (nabídkou či poptávkou), přičemž pravděpodobnost výběru je vážená podle aktuálního objemu na obou stranách knihy. Tento mechanismus upřednostňuje doplňování méně zastoupené strany, čímž napomáhá vyrovnaní tržní nerovnováhy. Ceny objednávek vycházejí z nejlepší dostupné cenové úrovně, na kterou je aplikována malá stochastická odchylka, čímž vzniká přirozený cenový šum. Objem objednávek je omezen s ohledem na dostupnou protistrannou likviditu, což zajišťuje stabilní tok objednávek bez nadměrného zatěžování trhu. Tímto způsobem agent dynamicky a kontinuálně doplňuje likviditu a přispívá k zachování přirozeného tržního toku.

Generátor likvidity slouží jako nástroj pro obohacení tržního prostředí o umělou likviditu, čímž napomáhá udržení kontinuity trhu i za podmínek nízké aktivity. Jeho použití zajišťuje, že v systému vždy existuje dostatek objednávek pro interakci s dalšími agenty, což umožňuje spolehlivější testování, validaci strategií a pozorování chování trhu i v okrajových nebo méně aktivních situacích.

3.3.3 Algoritmickí obchodní agenti

Algoritmickí obchodní agenti jsou navrženi pro automatizované provádění obchodních strategií v simulovaném tržním prostředí. Vycházejí ze společného základu — třídy `AlgorithmicTrader`, která poskytuje nástroje pro zpracování tržních dat, generování obchodních signálů a správu obchodních pozic. Tato třída slouží jako základ pro implementaci různých obchodních strategií, přičemž každý agent má specifický přístup k obchodování a řízení rizik.

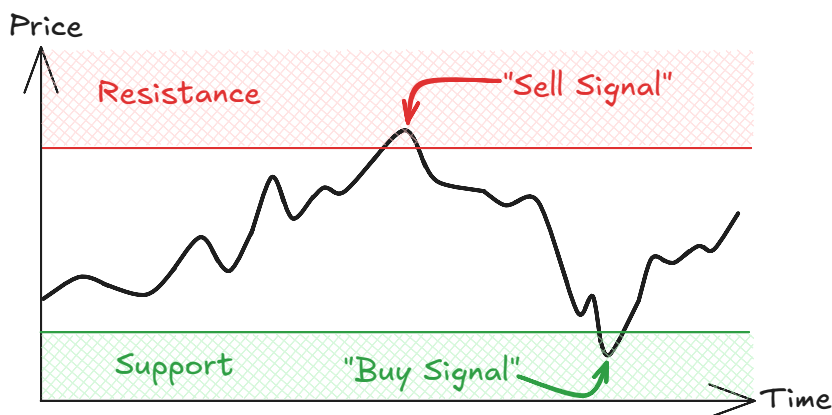
V rámci této práce byly implementovány následující typy algoritmických obchodních agentů:

- **Range trader** – identifikuje cenové pásmo (*Support* a *Resistance*) a otevírá pozice na jeho hranicích s očekáváním, že se cena bude nadále pohybovat uvnitř tohoto rozmezí.
- **Momentum trader** – sleduje výrazné cenové pohyby a otevírá pozice ve směru aktuálního trendu s cílem využít setrvačnosti trhu.
- **Scalping trader** – provádí rychlé krátkodobé obchody s cílem profitovat z drobných cenových rozdílů.
- **Swing trader** – zaměřuje se na využívání krátkodobých cenových fluktuací v rámci širších trendů, kdy identifikuje cenové korekce a čeká na okamžiky, kdy trh vykazuje signály pro návrat k hlavnímu trendu.

- **Regression trader** – využívá regresních metod a statistických modelů ke kvantitativní analýze historických dat a predikci budoucího vývoje cen, přičemž tyto predikce používá k identifikaci obchodních příležitostí na základě očekávaných cenových pohybů.
- **LSTM trader** – uplatňuje hluboké učení a rekurentní neuronové sítě (LSTM) k analýze časových řad a předpovědi budoucích cenových pohybů na základě historických dat, přičemž modeluje sekvenční závislosti v datech pro predikci trendů a změn na trhu.
- **Q-learning trader** – využívá metody posilovaného učení, k optimalizaci obchodních strategií na základě zpětné vazby z interakcí s trhem. Tento přístup umožňuje agentovi adaptivně upravovat své rozhodovací procesy na základě odměn a penalizací, což vede k maximalizaci dlouhodobých zisků a efektivnímu přizpůsobení se dynamice tržního prostředí.
- **Spoofing trader** – jeho strategie spočívá v umístění falešných objednávek s cílem manipulovat s tržními cenami a likviditou, přičemž tyto objednávky nejsou určeny k realizaci, ale slouží pouze k ovlivnění tržního sentimentu.

Range Trader

Range Trader je algoritmičtý obchodní agent zaměřený na obchodování v rámci cenového pásma. Agent využívá úrovně *podpory* a *odporu* k umístění nákupních příkazů pod úroveň podpory a prodejních příkazů nad úroveň odporu s cílem profitovat z cenových fluktuací v tomto pásmu, jak ilustruje Obrázek 3.3.



Obrázek 3.3: Schematické znázornění obchodní strategie Range Trader v rámci cenového pásma.

Na základě aktuálních tržních dat agent pravidelně aktualizuje své interní cenové úrovně poptávky a nabídky a odstraňuje neaktuální obchodní příkazy, tyto

aktualizace jsou realizovány prostřednictvím metody `handle_market_data`. Tato strategie umožňuje efektivní obchodování ve stabilních tržních podmínkách a přispívá k udržování likvidity v rámci vymezených cenových hladin.

Momentum Trader

Momentum Trader je agent obchodující na základě identifikace tržního momentu, tedy směru a síly cenového pohybu v čase. Využívá historická data k výpočtu indikátorů hybnosti a provádí obchodní rozhodnutí v závislosti na síle trendu. Obchodní rozhodnutí vycházejí z hodnoty zvoleného indikátoru hybnosti (percentage change, RSI, SMA, EMA), který je počítán na základě dat z klouzavého okna cen.

Pokud je hybnost pozitivní, agent otevírá nákupní pozici, v opačném případě prodává. Obchodování však probíhá pouze tehdy, pokud je současně splněna podmínka nízké volatility.

Volatilita je posuzována pomocí směrodatné odchylky historických cen, jejíž výpočet je uveden ve vzorci (3.1). Tento ukazatel pomáhá určovat, zda jsou podmínky pro obchodování příznivé. Pokud volatilita překročí stanovenou prahovou hodnotu, agent neprovádí žádné obchodní příkazy, aby se vyhnul obchodování v obdobích s vysokou nestabilitou na trhu.

Dále následují implementované indikátory hybnosti, které agent využívá pro analýzu tržního trendu a rozhodování o obchodech:

- **Procentuální změna (Percentage Change)** – Vyjadřuje relativní změnu ceny v rámci klouzavého okna délky k . Kladná hodnota signalizuje nákup, záporná prodej. Procentuální změnu lze vypočítat jako:

$$M_n = \frac{P_n - P_{n-k}}{P_{n-k}} \cdot 100, \quad (3.2)$$

kde P_n je aktuální cena a P_{n-k} cena před k obdobími.

- **Index relativní síly (Relative Strength Index, RSI)** [36] – Měří relativní sílu růstu vůči poklesu cen v rámci klouzavého okna. Je definován jako:

$$RSI = 100 - \left(\frac{100}{1 + RS} \right), \quad \text{kde } RS = \frac{\text{průměrný zisk}}{\text{průměrná ztráta}}. \quad (3.3)$$

Průměrný zisk a průměrná ztráta se počítají jako aritmetický průměr pouze z kladných (resp. záporných) změn v daném okně délky k . Konkrétně:

- Zisk: pokud $\Delta P_n = P_n - P_{n-1} > 0$, zahrnuje se do výpočtu průměrného zisku.

- Ztráta: pokud $\Delta P_n < 0$, bere se její absolutní hodnota a zahrnuje se do průměrné ztráty.
- Pokud je změna nulová, není započítána ani do zisku, ani do ztráty.

Hodnota RSI nad 70 signalizuje překoupený trh (prodejní signál), pod 30 přeprodaný trh (nákupní signál).

- **Jednoduchý klouzavý průměr (Simple Moving Average, SMA)** – SMA je aritmetický průměr cen v daném časovém období, jak je znázorněno v následující rovnici:

$$SMA_n = \frac{1}{k} \sum_{i=n-k+1}^n P_i. \quad (3.4)$$

Pokud aktuální cena překročí hodnotu SMA, agent to interpretuje jako signál růstu a otevírá nákupní pozici. Pokud cena klesne pod hodnotu SMA, agent považuje tento signál za indikaci k prodeji.

- **Exponenciální klouzavý průměr (Exponential Moving Average, EMA)** [36] – EMA je vážený průměr cen, který klade větší důraz na novější hodnoty, což umožňuje rychlejší reakci na změny cenového pohybu. Výpočet EMA je dán vzorcem:

$$EMA_n = \alpha \cdot P_n + (1 - \alpha) \cdot EMA_{n-1}, \quad \alpha \in (0, 1). \quad (3.5)$$

kde P_n je aktuální cena, EMA_{n-1} je hodnota EMA v předchozím období a α je váhový koeficient, který určuje, jak silný je vliv novějších cen (typicky se používají hodnoty $\alpha = \frac{2}{k+1}$, kde k je délka klouzavého okna).

Pokud cena vzroste nad hodnotu EMA, zahajuje agent nákup. Naopak, pokud cena klesne pod EMA, interpretuje agent tuto situaci jako příležitost k prodeji.

Tento přístup je vhodný pro trhy s jasně definovanými trendy, nejlépe funguje v obdobích, kdy trh vykazuje stabilní růst nebo pokles. Jeho účinnost však může být omezená v obdobích vysoké volatility nebo nízké likvidity, kdy jsou trhy méně předvídatelné.

Scalping Trader

Scalping Trader představuje algoritmického obchodního agenta orientovaného na realizaci krátkodobých transakcí s cílem těžit z malých cenových fluktuací. Limitní příkazy jsou systematicky zadávány v bezprostřední blízkosti aktuální střední ceny, přičemž rozpětí mezi nákupní a prodejní cenou je adaptivně

modifikováno v závislosti na aktuální tržní volatilitě, která je odhadována pomocí směrodatné odchylky historických středních cen v rámci klouzavého okna v souladu se vztahem (3.1).

Obchodní frekvence je regulována minimálním časovým intervalem mezi jednotlivými transakcemi pro daný instrument, čímž se omezuje riziko nadměrné obchodní expozice a přispívá k udržitelné obchodní dynamice.

Tato strategie je nejvhodnější pro trhy s vysokou likviditou a relativně nízkou volatilitou, kde umožňuje časté realizování obchodů s úzkým cenovým rozpětím a minimálními transakčními náklady, aniž by byla výrazně ovlivněna cenovými výkyvy.

Swing Trader

Swing Trader je algoritmický obchodní agent navržený k identifikaci potenciálních obrátů v cenovém vývoji aktiv prostřednictvím kombinace technických indikátorů. Strategie vychází z předpokladu, že trhy se pohybují v cyklech a že změny trendu lze částečně predikovat pomocí historicky ověřených cenových vzorců. Implementace pracuje se třemi klíčovými prvky: **Bollingerova pásma**, **Fibonacciho úrovně zpětných pohybů** a **Imbalance index**.

Pro každé obchodované aktivum je udržována klouzavá časová řada cen o délce k , která slouží jako vstup pro výpočet Bollingerových pásem. Tato technika využívá kombinaci klouzavého průměru a směrodatné odchylky za účelem určení mezních hodnot cenového pásma, viz Obrázek 3.4. Horní a dolní hranice pásma, dále označované jako B_U (Upper Band) a B_L (Lower Band), jsou definovány následujícími vztahy [36]:

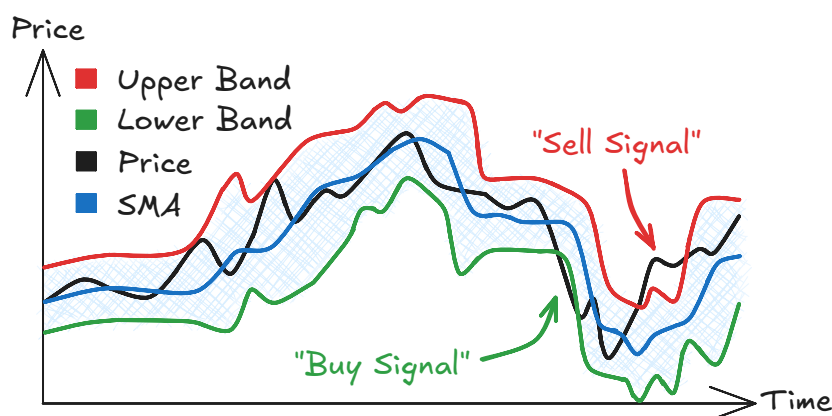
$$\sigma = \sqrt{\frac{1}{k} \sum_{i=n-k+1}^n (P_i - SMA_n)^2}, \quad (3.6)$$

$$B_U = SMA_n + m \cdot \sigma, \quad B_L = SMA_n - m \cdot \sigma, \quad (3.7)$$

kde SMA_n je definován vztahem (3.4), P_i značí jednotlivé ceny v klouzavém okně a m je parametr určující šířku pásma, obvykle nastavený na $m = 2$.

Kromě volatility jsou při rozhodování zohledněny také cenové hladiny odvozené z Fibonacciho úrovní zpětných pohybů (Fibonacci Retracement). Ty jsou vypočteny z historického maxima H a minima L ceny ve zvoleném časovém rámci [36]:

$$\text{Retracement}_{r\%} = H - \frac{r}{100} \cdot (H - L), \quad r \in \{23.6, 38.2, 50.0, 61.8\}. \quad (3.8)$$



Obrázek 3.4: Schematické znázornění obchodní logiky využívající Bollingerova pásma

Výpočet těchto úrovní poskytuje vodítko k identifikaci potenciálních zón podpory a odporu, v nichž může dojít k obratu trendu.

Třetím faktorem je *imbalance index* (viz (2.4)), který kvantifikuje převahu jedné strany trhu (nabídky nebo poptávky) na základě struktury knihy objednávek.

Obchodní rozhodnutí jsou realizována pouze při souběhu více faktorů. Nákupní příkaz je zadán, pokud cena klesne pod dolní Bollingerovo pásmo, nachází se pod hranicí 38,2 % Fibonacciho úrovní zpětných pohybů a současně je zaznamenána převaha nabídky nad poptávkou. Analogicky je prodejní příkaz generován, pokud cena přesáhne horní Bollingerovo pásmo, nachází se nad úrovní 61,8 % a trh vykazuje převažující poptávku.

Tato strategie je vhodná pro trhy s relativně stabilními cenovými pohyby, kde lze identifikovat opakující se vzory a změny trendu. Funguje dobře zejména v obdobích, kdy trh vykazuje nízkou až střední volatilitu.

Regression Trader

Strategie tohoto agenta je založena na prediktivním modelování budoucí nákupní a prodejní ceny aktiva pomocí regresních metod. Vychází z předpokladu, že cenové časové řady nesou určitou míru predikovatelnosti, zejména v krátkodobém horizontu, a že lze tuto informaci využít pro rozhodování o obchodní strategii. Model zohledňuje jak vývoj samotných cen, tak i objemy limitních objednávek, které slouží jako váhové faktory příslušných cen v rámci výpočtu.

V rámci implementace je možné volit mezi několika variantami regresních modelů, včetně *linear*, *ridge*, *lasso*, *bayesian* a *random_forest*. Tyto modely se liší mírou regularizace a schopností zachytit nelineární vztahy, čímž uživateli

umožňují ladit strategii s ohledem na charakter daného trhu. Využitý model je trénován nad klouzavým časovým oknem dat, jehož velikost je dynamicky upravována podle aktuální volatility – konkrétně směrodatné odchylky střední ceny (viz vztah (3.1)). Agent se tímto způsobem adaptivně přizpůsobuje proměnlivé tržní dynamice a optimalizuje délku analyzovaného historického okna.

Obchodní rozhodnutí se zakládá na rozdílu mezi predikovanou a aktuální cenou. Pokud je predikovaná cena nabídky výrazně vyšší než aktuální, agent prodává; pokud je predikovaná cena poptávky nižší, nakupuje. Pro minimalizaci vlivu šumových signálů byl zaveden prahový limit, který omezuje realizaci obchodů v případě nízké predikční přesnosti.

Tato strategie je optimální pro trhy s nízkou až střední volatilitou a krátkodobými trendy, kde prediktivní model přináší výhodu při detekci mírných cenových pohybů. Díky možnosti zohlednění objemů a adaptivní volbě délky okna lze strategii aplikovat i v prostředí s proměnlivou dynamikou.

LSTM Trader

Tato strategie využívá model hlubokého učení, konkrétně **LSTM** (Long Short-Term Memory) model, k predikci budoucích cen aktiv na základě historických dat a dalších tržních ukazatelů. LSTM je specifický typ rekurentní neuronové sítě, který je optimalizován pro modelování časových řad a efektivní zachycování dlouhodobých závislostí v sekvenčních datech, což ho činí ideálním nástrojem pro predikci cenových trendů.

Strategie se skládá z následujících klíčových kroků:

1. **Zpracování tržních dat:** Agent kontinuálně sbírá data o aktuálních cenách nabídky a poptávky, obchodovaných objemech a imbalance indexu z knihy objednávek. Současně vypočítává odvozené tržní indikátory, jako je spread mezi nabídkou a poptávkou či cenové momentum, zde definované jen jako cenová změna mezi po sobě jdoucími časovými okamžiky. Získaná data jsou průběžně aktualizována a uchovávána v klouzavém časovém okně, které slouží jako vstup pro predikční model.
2. **Trénování modelu:** Model je trénován metodou *mini-batch gradientního sestupu*⁵, přičemž agent průběžně shromažďuje tržní data, která jsou ukládána do **replay bufferu**, paměťového zásobníku obsahujícího historické

⁵Mini-batch gradientní sestup je optimalizační metoda, která aktualizuje parametry modelu na základě průměrného gradientu vypočteného z malého náhodně vybraného podmnožiny dat (mini-batch), což kombinuje rychlost stochastického gradientního sestupu a stabilitu batch gradientního sestupu.

vzorky. Tyto vzorky jsou následně využívány při trénování modelu. Optimalizace probíhá pomocí algoritmu *Adam* a ztrátové funkce *Mean Squared Error (MSE)*.

3. **Predikce ceny:** Na základě historických dat a natrénovaného modelu agent predikuje ceny nabídky a poptávky pro nadcházející časový krok.
4. **Realizace obchodů:** Na základě predikovaných cen agent realizuje obchody, přičemž porovnává rozdíl mezi predikovanými cenami nabídky a poptávky a aktuálními tržními cenami. Pokud tento rozdíl překročí předem stanovený prahový limit, agent umístí obchodní příkaz. Tento přístup je analogický jako u agenta využívajícího regresi.

Strategie je vhodná pro trhy s výraznou dynamikou a schopností trendů se měnit v krátkodobém horizontu, kde hluboké učení může pomoci zachytit složité vzory v datech, které nejsou snadno identifikovatelné tradičními analytickými metodami.

Q-learning Trader

Tento obchodní agent staví na principech **Q-learningu**, což je jedna z metod **reinforcement learningu**, tedy učení s posilováním. Místo přímé predikce budoucích cen se zde agent učí *optimální politiku rozhodování* – tedy jaké akce je vhodné provádět v různých tržních situacích, aby maximalizoval svůj dlouhodobý zisk. Učení probíhá interaktivně – skrze zpětnou vazbu z prostředí.

Celý proces lze rozdělit do následujících fází:

1. **Reprezentace stavu:** Agent si vytváří diskrétní reprezentaci prostředí na základě tří kvantitativních rysů aktuální tržní situace: změnou ceny mezi dvěma po sobě jdoucími okamžiky, odchylkou aktuální ceny od klouzavého průměru a volatilitou. Tyto příznaky jsou kvantizovány (zaokrouhleny) za účelem omezení velikosti stavového prostoru, což je nezbytné pro efektivní aplikaci tabulkového Q-learningu. Výsledná trojice slouží jako klíč do Q-tabulek, které uchovávají odhady očekávaných návratností pro jednotlivé akce ve specifických stavech.
2. **Volba akce:** V každém časovém okamžiku agent zvažuje tři možné akce: nákup, prodej, nebo držení pozic. Akce je vybírána pomocí **epsilon-greedy** strategie: s pravděpodobností ϵ volí náhodnou akci (explorace, z angl. Exploration), jinak preferuje tu, která má dle Q-tabulek nejvyšší očekávanou hodnotu (exploatace, z angl. Exploitation).

3. **Realizace obchodu:** Pokud je zvolen nákup nebo prodej, je zadán limitní příkaz s mírnou korekcí ceny vůči aktuální střední ceně a jsou zaznamenány podrobnosti o této transakci pro pozdější vyhodnocení.
4. **Ohodnocení akce:** Ve chvíli, kdy dorazí nové tržní údaje, dochází k vyhodnocení předchozí akce. Odměna r je definována jako změna ceny ve směru zvolené pozice:

$$r = \begin{cases} p_{n+1} - p_n & \text{pro nákup} \\ p_n - p_{n+1} & \text{pro prodej} \\ -|p_{n+1} - p_n| & \text{pro držení pozic} \end{cases}$$

kde p_n je cena v době zadání příkazu a p_{n+1} je cena po doručení nových aktuálních dat. Odměna tak penalizuje i pasivitu v případě volatilního trhu.

5. **Aktualizace Q-hodnot:** Po vyhodnocení odměny následuje aktualizace Q-hodnot prostřednictvím *Bellmanovy rovnice* [37], která umožňuje aproximaci optimální politiky:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \cdot \left[r + \gamma \cdot \max_{a'} Q(s', a') - Q(s, a) \right] \quad (3.9)$$

kde:

- $Q(s, a)$ je očekávaná návratnost pro akci a ve stavu s ,
- $\alpha \in (0, 1)$ je *rychlost učení*, udávající, jak silně se má Q-hodnota aktualizovat,
- $\gamma \in [0, 1)$ je *diskontní faktor*, který určuje důležitost budoucích odměn,
- r je odměna získaná za přechod ze stavu s do stavu s' .

Tato aktualizace směřuje Q-hodnoty k optimálním návratnostem, čímž agent postupně aproximuje optimální politiku.

6. **Adaptace strategie průzkumu:** Aby byla zachována rovnováha mezi průzkumem nových strategií (explorací) a využíváním dosavadních znalostí (exploatací), dochází k postupnému exponenciálnímu snižování parametru ϵ v čase. Tento přístup umožňuje, aby agent v počátečních fázích aktivně zkoumal různé možnosti, zatímco v pozdějších fázích směřoval k ustálené a efektivní obchodní strategii.

Klíčovou vlastností této strategie je učení se *na základě důsledků vlastního chování*. Agent si postupně vytváří představu o tom, jaká rozhodnutí v dané tržní situaci vedou k vyšším ziskům, a tuto strategii dále zdokonaluje. Odměna není určována

okamžitě, ale až se zpožděním při příchodu nových dat, což odráží reálný aspekt obchodování, kde dopad rozhodnutí nelze vyhodnotit ihned.

Tato metoda je vhodná zejména v prostředí, kde je potřeba adaptivně reagovat na měnící se tržní podmínky a kde není k dispozici spolehlivý predikční model, ale spíše potřeba najít účinné chování na základě zkušenosti.

Spoofing Trader

Strategie **spoofingu** představuje formu tržní manipulace, při níž dochází k záměrnému umístování objemově rozsáhlých limitních objednávek mimo aktuální střední cenu za účelem vytvoření falešného dojmu o nabídce či poptávce. Tyto objednávky nejsou určeny k realizaci, ale k manipulaci s limitní knihou a chováním ostatních účastníků trhu.

Agent periodicky analyzuje aktuální střední cenu a volí stranu spoofovacích objednávek podle vlastní pozice – pokud drží aktivum, spoofuje nákupní stranu. Spoofovací příkazy jsou objemově výrazné a umístované s pevnou relativní vzdáleností od střední ceny. Na opačné straně trhu agent zároveň umístí reálný příkaz s nižším objemem a blíže k tržní ceně s cílem profitovat z krátkodobého cenového posunu.

Bezprostředně po zadání reálného příkazu jsou spoofovací objednávky stornovány. Tato strategie využívá asymetrie v tržní reakci na zdánlivé posuny v likviditě, aniž by se vystavovala významnému exekučnímu riziku.

Je třeba zdůraznit, že **spoofing** je ve většině jurisdikcí považován za zakázanou praxi, neboť podkopává integritu a transparentnost finančních trhů. V kontextu simulačního prostředí však může sloužit jako nástroj pro analýzu tržní mikrostruktury, studium dopadů manipulativního chování a testování odolnosti algoritmických strategií vůči nečistým tržním praktikám.

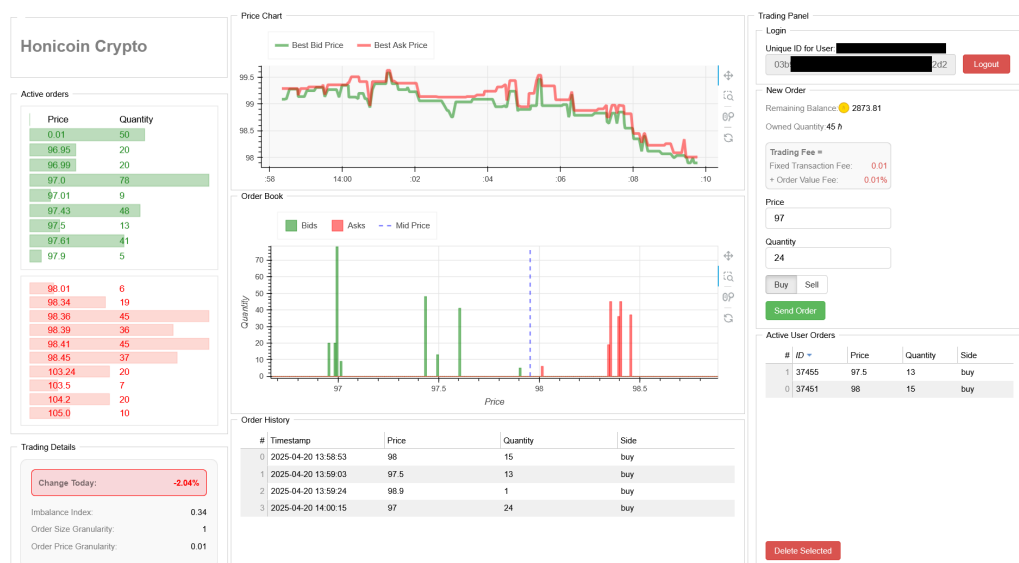
3.3.4 Manuální obchodní agent

Manuální obchodní agent představuje rozhraní pro interaktivní zadávání pokynů a řízení obchodní aktivity ze strany uživatele. Umožňuje přímou participaci na trhu bez nutnosti automatizace obchodní logiky. Interakce může probíhat buď prostřednictvím metod poskytovaných třídou `Trader`, která umožňuje programově řízené zadávání obchodních příkazů a správu pozic, nebo pomocí grafického webového rozhraní, které slouží jako intuitivní nástroj pro obchodování a sledování aktuálního tržního vývoje.

Grafické uživatelské rozhraní

Součástí klientské části systému je grafické uživatelské rozhraní navržené s důrazem na srozumitelnost, jednoduchost a edukativní charakter. Toto rozhraní umožňuje uživatelům ručně zadávat obchodní příkazy, sledovat aktuální stav trhu a vyhodnocovat vývoj obchodních pozic v reálném čase. Využití tohoto rozhraní je ideální zejména pro testování a demonstraci obchodních strategií v simulovaném prostředí bez nutnosti přímé programátorské interakce.

Rozhraní je rozděleno do několika funkčních panelů, které poskytují klíčové informace o stavu trhu a o uživatelském účtu. Přehledné uspořádání těchto komponent je znázorněno na Obrázku 3.5. Patří mezi ně zejména:



Obrázek 3.5: Navržené webové uživatelské rozhraní pro manuální obchodování

- **Panel aktivních příkazů,** zobrazující aktuální přehled deseti nejlepších nákupních a prodejních objednávek v objednávkové knize včetně jejich cen a množství.
- **Cenový graf,** který vizualizuje vývoj nejlepší poptávkové a nabídkové ceny v čase a umožňuje sledovat základní tržní trendy.
- **Kniha objednávek,** poskytující agregovaný pohled na rozložení nákupních a prodejních příkazů v rámci cenového spektra, včetně indikace střední ceny.
- **Panel obchodních statistik,** shrnující základní metriky jako denní změnu ceny, hodnotu imbalance indexu a granularitu obchodování.

- **Historie obchodů**, zobrazující chronologický výpis naposledy realizovaných transakcí.
- **Obchodní panel**, sloužící k zadávání nových příkazů, sledování stavu účtu a správě aktivních objednávek.

Přihlášení uživatele probíhá prostřednictvím e-mailové adresy a hesla, přičemž každému účtu je přiřazen unikátní identifikátor. Ten může být následně využit i pro autorizovanou programově řízenou interakci.

Grafické uživatelské rozhraní bylo realizováno s využitím knihovny Bokeh⁶, která poskytuje robustní nástroje pro tvorbu interaktivních vizualizací v jazyce Python a umožňuje jejich snadnou integraci do webového prostředí formou dynamických aplikací.

Interaktivní komponenty rozhraní umožňují uživateli aktivně pracovat s jednotlivými prvky zobrazení a získávat detailní přehled o tržním dění v reálném čase. Vizualizace jsou aktualizovány průběžně bez nutnosti manuálního obnovení stránky. Díky tomu je možné v reálném čase sledovat vývoj cen, objednávek i dalších klíčových parametrů trhu. Uživatel má k dispozici širokou škálu interaktivních nástrojů, mezi které patří například **posun grafu** (pan), **přiblížení** pomocí výběru oblasti (box zoom) či kolečka myši (wheel zoom), a možnost resetu výchozího zobrazení. Dále jsou integrovány **hover nástroje**, které umožňují zobrazit doplňkové informace o konkrétních vizuálních entitách v grafech. Tabulky s objednávkami i obchodní historií navíc podporují dynamické řazení dle zvoleného sloupce, což dále zvyšuje přehlednost a uživatelský komfort při práci s aplikací.

Rozhraní představuje klíčový nástroj pro interaktivní testování a vizualizaci chování obchodních algoritmů a přispívá k celkovému zpřístupnění systému širšímu okruhu uživatelů. Podrobný popis jeho funkcionality spolu s návodem k použití je uveden v uživatelské příručce, která je přiložena jako Příloha B.

3.4 Testování systému

Za účelem komplexního ověření funkční správnosti vyvíjeného obchodního systému byla implementována sada jednotkových testů využívajících standardní modul unittest⁷ jazyka Python. Testy pokrývají klíčové komponenty systému včetně reakcí na běžné, hraniční i chybové stavy.

⁶<https://docs.bokeh.org/>

⁷<https://docs.python.org/3/library/unittest.html>

3.4.1 Testované scénáře

- **Základní obchodní operace** – ověřuje registraci obchodníka, zadání, modifikaci a rušení objednávky, včetně korektní správy zůstatku.
- **Správa knihy objednávek** – ověřuje správné prioritní řazení objednávek podle ceny a času podání.
- **Validace vstupů** – kontroluje odmítnutí neplatných požadavků (např. záporné ceny, neexistující ID) s adekvátní zpětnou vazbou.
- **Výpočet cílového množství** – prověřuje funkci `compute_quantity` v různých tržních podmínkách a alokačních strategiích.
- **Vyhodnocení stavu objednávek** – testuje klasifikaci stavů (např. *filled*, *partially filled*, *rejected*, *invalid*) na základě výsledku párování.
- **Historie knihy objednávek** – ověřuje uchovávání a zpřístupnění historických snapshotů knihy.
- **Odstraňování neaktuálních objednávek** – kontroluje identifikaci a mazání nerelevantních objednávek na základě cenové a časové odchylky.

Testovací scénáře pokrývají klíčové funkce systému v různých podmínkách. Jejich realizace přispívá ke zvýšení spolehlivosti, bezpečnosti a udržitelnosti systému v produkčním prostředí.

Experimenty a Diskuze

4

V této kapitole jsou popsány provedené experimenty sloužící k validaci funkčních a výkonnostních charakteristik navrženého simulačního systému. Následně jsou analyzovány získané výsledky a formulovány závěry týkající se jejich interpretace, identifikovaných omezení a možností dalšího rozvoje systému.

4.1 Nastavení experimentů

Pro provedení experimentů byl využit **e-INFRA CZ Openstack Cloud**, v rámci něhož byl nasazen virtuální stroj s následujícími hardwarovými a softwarovými specifikacemi:

- **Typ virtuálního stroje:** e1.4core-16ram
- **Specifikace:** 4 vCPU, 16 GB RAM, 80 GB disk
- **Operační systém:** Debian 11 (debian-11-x86_64)

V rámci této infrastruktury byly provedeny následující experimenty:

- **Test zátěže serveru:** Cílem tohoto testu bylo ověřit výkonnost a stabilitu serverové infrastruktury při různých úrovních provozní zátěže. Simulovány byly jak běžné, tak extrémní provozní podmínky, přičemž byly generovány souběžné požadavky na server s cílem sledovat jeho odezvu, propustnost a odolnost vůči zahlcení. Test hodnotil schopnost systému zvládat vysoký počet simultánních relací bez výpadků a s udržením přijatelných latencí.
- **Velikonoční burzovní hra:** Tento experiment probíhal jako týdenní beta testování burzovního simulátoru, následované čtyřmi dny aktivní hry. Hra byla zaměřena na testování interakce s uživatelským rozhraním pro manuální obchodování prostřednictvím webového UI. Kromě manuálního

obchodování byla účastníkům také poskytnuta možnost vyzkoušet algoritmické obchodování pomocí **Pythonu**. Experiment byl součástí reálného provozu, kde účastníci obchodovali s fiktivním produktem a měli za úkol co nejlépe zhodnotit svůj počáteční kapitál 10 000 jednotek.

Cílem experimentu bylo nejen ověření technických aspektů simulátoru a správného vykonávání obchodních příkazů, ale rovněž otestování jeho chování při dlouhodobém provozu s větším počtem současně aktivních uživatelů. Důraz byl kladen na sledování zátěže serverové části, stabilitu systému při kontinuálním zpracování požadavků a jeho celkovou robustnost.

- **Test algoritmických obchodníků:** Tento experiment byl zaměřen na validaci chování autonomních obchodních agentů v rámci simulovaného tržního prostředí. Hlavním cílem bylo ověřit technickou správnost implementace jednotlivých strategií a posoudit jejich schopnost adaptace na proměnlivé tržní podmínky. Experiment nebyl koncipován jako komparativní analýza výkonnosti jednotlivých strategií, neboť výsledky jsou zásadně ovlivněny konkrétní konfigurací trhu a simulovaných scénářů. Důraz byl kladen především na stabilitu, konzistenci chování agentů a jejich schopnost interakce v komplexním prostředí s proměnlivou likviditou a volatilitou.
- **Test algoritmických traderů v přítomnosti spoofingu:** V tomto experimentu byla obchodní aktivita agentů ovlivněna přítomností jediného spoofing agenta — specializovaného účastníka, který vkládal falešné objednávky za účelem manipulace tržní dynamiky. Cílem experimentu bylo nejen ověřit odolnost algoritmických strategií vůči tržní manipulaci, ale také analyzovat, zda spoofing agent skutečně ovlivňuje chování trhu a jaký má dopad na ostatní obchodníky reagující na zavádějící cenové signály.

Poznámka k transakčním poplatkům:

V rámci všech simulací byla na každou realizovanou transakci aplikována následující struktura poplatků:

1. Fixní poplatek ve výši 0,01 jednotky za každou objednávku (nezávisle na její velikosti),
2. Proporcionální poplatek ve výši 0,01 % z hodnoty objednávky, kde hodnota objednávky je definována jako součin obchodované kvantity a ceny specifikované v příslušné objednávce.

4.2 Test výkonnosti serveru

Pro účely ověření stability a výkonnosti serverové infrastruktury pod různými úrovněmi zátěže byly provedeny zátěžové testy s využitím nástroje Locust¹. Simulovaní uživatelé v pravidelných intervalech odesílali obchodní požadavky a dotazy na tržní data. Cílem bylo vyhodnotit schopnost systému efektivně obsluhovat vysoký počet paralelních relací a analyzovat odezvu serveru při různé úrovni zatížení.

Byly provedeny dva samostatné testovací scénáře:

- Standardní zátěž: 100 simulovaných uživatelů s rychlostí připojování 1 uživatel za sekundu.
- Vysoká zátěž: 2000 simulovaných uživatelů s rychlostí připojování 20 uživatelů za sekundu.

Shrnutí dosažených výsledků obou testů je uvedeno v Tabulce 4.1.

Tabulka 4.1: Shrnutí výsledků zátěžových testů serveru při standardní zátěži (100 uživatelů) a vysoké zátěži (2000 uživatelů)

Ukazatel	Standardní zátěž	Vysoká zátěž
Počet požadavků (GET)	7597	31 698
Počet požadavků (POST)	15 119	65 774
Celkový počet požadavků	22 716	97 472
Počet selhání	0	0
Průměrná doba odezvy	9.73 ms	1941 ms
Minimální doba odezvy	2 ms	2 ms
Maximální doba odezvy	92 ms	4606 ms
Průměrná velikost odpovědi	1818 B	1568 B
Počet požadavků za sekundu (RPS)	189.82	492.40
Medián doby odezvy (50% požadavků)	7 ms	2100 ms
90. percentil doby odezvy	18 ms	3300 ms
95. percentil doby odezvy	24 ms	3400 ms
99. percentil doby odezvy	44 ms	3800 ms

Při standardní zátěži vykazoval server extrémně nízké hodnoty latence, s průměrnou odezvou pod 10 ms a mediánem 7 ms. Nebyla zaznamenána žádná selhání a i 99. percentil odezvy se držel hluboko pod 50 ms. To potvrzuje, že za

¹<https://locust.io/>

běžného provozu je systém vysoce optimalizovaný a bezproblémově zvládá požadavky v reálném čase.

Při vysoké zátěži s 2000 souběžnými uživateli nebyla opět zaznamenána žádná selhání požadavků. Průměrná doba odezvy kolem 1.94 sekundy, společně s mediánem 2.1 sekundy, naznačuje, že systém si udržoval konzistentní výkonnost i při současném odbavení vysokého počtu paralelních požadavků. Přestože nejvyšší zaznamenaná odezva dosáhla až 4606 ms, hodnoty vyšších percentilů (90 % při 3300 ms a 99 % při 3800 ms) ukazují, že většina požadavků byla vyřízena v přijatelných mezích bez kritických výkyvů. Zásadní je, že i při extrémním zatížení nedošlo k žádným chybám a systém udržel průměrnou propustnost téměř 500 požadavků za sekundu.

Výsledky tak ukazují, že architektura systému je velmi dobře optimalizována jak pro běžný provoz, tak pro extrémní zatížení. Při standardní zátěži (100 uživatelů) systém vykazuje minimální latenci v jednotkách milisekund a bezchybně zpracování požadavků. Při extrémní zátěži (2000 uživatelů) si systém zachovává stabilitu, ačkoliv s vyšší průměrnou odezvou.

Tyto výsledky potvrzují schopnost systému efektivně horizontálně škálovat a zvládat zátěž v řádu desítek tisíc požadavků během několika minut bez ztráty stability či kritického snížení kvality služeb.

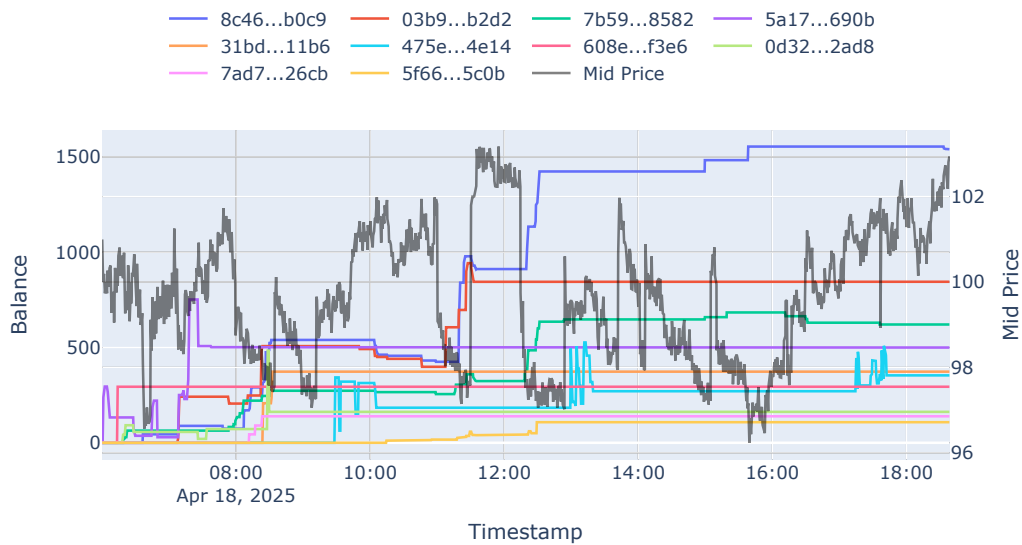
4.3 Velikonoční burzovní hra

Velikonoční burzovní hra představovala klíčový experiment v rámci testování burzovního simulátoru za reálných podmínek. Během čtyřdenního období probíhala interaktivní simulace, do níž se zapojilo celkem 86 zaregistrovaných účastníků. Cílem bylo nejen ověřit funkčnost systému z pohledu uživatele, ale také sledovat jeho chování v delším časovém horizontu při současném zatížení více uživateli.

Nejvyšší aktivita byla zaznamenána v první den soutěže, kdy se do obchodování zapojilo přibližně 20 aktivních účastníků. Tento den poskytl nejhodnotnější datové výstupy, včetně zajímavých výdělečných strategií, které ukázaly rozdíly mezi jednotlivými přístupy k obchodování a schopností reagovat na vývoj simulovaného trhu.

Na Obrázku 4.1 je zobrazen časový vývoj výdělků jednotlivých obchodníků právě během prvního dne hry. Levá osa y udává hodnotu čistého výdělku (bez započítání poplatků za zprávy), který je definován jako součet realizovaného zisku z obchodů a nerealizovaného zisku z aktuálně držných pozic (tj. aktuální střední cena násobená aktuálně držným objemem). Pravá osa zobrazuje vývoj střední ceny

Balance of Best Traders Over Time



Obrázek 4.1: Vývoj stření ceny a výdělků zúčastněných obchodníků během prvního dne velikonoční hry

aktiva v čase (černá křivka), která slouží jako indikátor obecného trendu na trhu.

Každá křivka ve vizualizaci odpovídá jednomu účastníkovi, přičemž jejich anonymizovaná UUID jsou uvedena v legendě. Z obrázku je patrné, že nejúspěšnějším účastníkem prvního dne byl obchodník s identifikátorem 8c46...b0c9, který dosáhl výdělku 1540 jednotek. Jeho výkonnost je charakteristická průběžným obchodováním během celého dne s častými hodinovými úseky držení pozice.

Vizualizace rovněž umožňuje kvalitativní analýzu obchodních strategií. Například u účastníka 5a17...690b lze pozorovat strategii založenou na počátečním aktivním obchodování, po kterém následoval výrazný pokles hodnoty portfolia (přibližně o 250 jednotek), načež obchodník již dále pouze držel pozici, což se projevuje jako horizontální průběh jeho výdělkové křivky.

Tabulka 4.2: Souhrn vybraných statistik obchodníků za první den experimentu

	User	Balance	Ret [%]	AvgVol	MaxVol	AvgBal	Ords
1	8c46...b0c9	11533.42	15.33	66.97	112	-5994.56	50
2	03b9...b2d2	10840.19	8.40	23.00	106	-1757.01	21
3	7b59...8582	10617.28	6.17	65.14	95	-6294.04	38
4	5a17...690b	10494.13	4.94	5.34	104	-94.38	165

pokračování na další straně

	User	Balance	Ret [%]	AvgVol	MaxVol	AvgBal	Ords
5	31bd...11b6	10372.20	3.72	59.44	70	-5801.62	3
6	475e...4e14	10324.89	3.25	8.87	100	-656.98	34
7	608e...f3e6	10292.99	2.93	99.67	100	-9966.53	1
8	0d32...2ad8	10160.53	1.61	30.77	95	-3026.52	9
9	7ad7...26cb	10138.60	1.39	29.70	30	-2919.61	5
10	5f66...5c0b	10107.08	1.07	12.49	18	-1202.42	48
11	d1ea...57fd	10084.68	0.85	46.51	99	-4697.33	12
12	3315...479f	10034.58	0.35	8.27	10	-822.50	1
13	25e5...fc58	10004.92	0.05	0.80	1	-78.68	1
14	50b3...bcaf	10000.00	0.00	0.00	0	0.00	0
15	e955...53cb	10000.00	0.00	0.00	0	0.00	0
16	0e00...85a5	10000.00	0.00	0.00	0	0.00	0
17	80ca...1551	10000.00	0.00	0.00	0	0.00	0
18	d9cb...1ea1	10000.00	0.00	0.00	0	0.00	0
19	0671...2f64	10000.00	0.00	0.00	0	0.00	0
20	51a1...0905	10000.00	0.00	0.00	0	0.00	0
21	990e...370c	10000.00	0.00	0.00	0	0.00	0
22	c2fc...c53e	9999.98	-0.00	0.00	0	0.00	1
23	7061...6e8e	9716.18	-2.84	13.90	95	-1539.39	687
24	b065...611b	4040.71	-59.59	38.73	100	-3943.15	90

Pro doplnění grafické analýzy byla sestavena tabulka s doplňujícími metrikami popisujícími obchodní chování a výkonnost vybraných účastníků (viz Tabulka 4.2). Kromě celkového výdělku (Balance), tj. zhodnocení počátečního kapitálu 10 000 po odečtení poplatků za zprávy, a jeho relativního vyjádření (Ret), zahrnuje tabulka také další ukazatele: průměrný (AvgVol) a maximální (MaxVol) objem držených akcií, průměrný zůstatek bez přepočtu aktuálních pozic (AvgBal) a celkový počet uskutečněných obchodních příkazů (Ords).

Z výsledků prvního dne experimentu lze vyzdvihnout několik obchodníků, jejichž strategie či výkonnost byly z analytického hlediska zvlášť zajímavé:

- **Trader 608e...f3e6** realizoval pouze jediný obchod, přesto dosáhl pozitivního výsledku a umístil se na 7. místě v celkovém pořadí prvního dne. Jeho strategie spočívala ve včasném nákupu a následném pasivním držení pozice po zbytek dne.
- **Účastník 7061...6e8e** vykázal extrémní aktivitu s celkovým počtem 687 obchodů, což z něj činí nejaktivnějšího obchodníka v rámci prvního dne.

Navzdory této aktivitě skončil se ztrátou a návratností -2,84 %, což může naznačovat nízkou efektivitu jeho vysokofrekvenční obchodní strategie.

- **Obchodník** 31bd...11b6 dosáhl zhodnocení 3,72 % s pouhými třemi realizovanými objednávkami. Tento výsledek může být interpretován jako důkaz promyšlené a precizně načasované strategie.
- **Trader** 5a17...690b, jehož chování bylo analyzováno i ve vizualizaci vývoje výdělků, uskutečnil během dne 165 objednávek, avšak jeho konečný zisk činil pouze 4,94 %. Tento příklad ilustruje, že vysoká obchodní aktivita nemusí nutně korelovat s vyšší výkonností.
- **Účastník** 50b3...bcaf se během hry pouze zaregistroval a nezadal žádný obchodní příkaz a setrval pouze v roli pozorovatele. Jeho konečný zůstatek tak zůstal roven výchozímu kapitálu 10 000. Tento případ ilustruje pasivní přístup jakožto legitimní strategii, která může být zvolena v prostředí s vysokou mírou nejistoty nebo nedostatečnou důvěrou v tržní podmínky.
- **Obchodník** c2fc...c53e vykázal téměř nulovou změnu v konečném zůstatku (9999.98), přičemž nikdy nevlastnil žádný objem. Tento výsledek může na první pohled působit neobvykle, avšak je vysvětlitelný. Účastník totiž zadal pouze jednu objednávku, která však nebyla nikdy realizována. Vzhledem k tomu, že obchodní objem nebyl nikdy přidělen, byla vykázaná ztráta způsobena výhradně poplatkem za zadání objednávky.

V prvních dvou dnech experimentu bylo povoleno i algoritmické obchodování. Tato možnost však byla následně omezena, neboť někteří uživatelé zneužili transparentnost *open-source* kódu k získání neférové výhody — například vytvořením AdminTradera, který dokázal obcházet standardní pravidla hry. Na základě těchto událostí byla od třetího dne algoritmická obchodní aktivita zablokována. Tento incident poukázal na nezbytnost zajištění odpovídajících bezpečnostních opatření i v rámci simulačního prostředí, zejména v souvislosti s dostupností zdrojového kódu. Transparentnost systému sice podporuje otevřenost a snadnější testování, avšak zároveň vytváří potenciální prostor pro zneužití, pokud není přístup adekvátně omezen. Experiment tak vedl k jednoznačnému závěru: *nelze spoléhat na férovost všech uživatelů i v neziskovém akademickém prostředí.*

Při vyhodnocování výsledků bylo nutné přistoupit k filtrování podvodných aktivit. Konkrétně byly ze vstupních dat odstraněny úseky, kde došlo k úpravám údajů s cílem získat nelegitimní výhodu, aby nebyly závěry ovlivněny neetickým chováním některých účastníků.

Z pohledu technické infrastruktury lze experiment označit za úspěšný. Serverová část systému zvládla čtyřdenní provoz bez výpadků, přičemž zvládala simultánní zátěž od desítek aktivních uživatelů. Webové uživatelské rozhraní bylo hodnoceno jako přehledné a intuitivní i ze strany uživatelů bez předchozích zkušeností s finančními nástroji nebo burzovním obchodováním. Celkově tak experiment splnil svůj účel jak z hlediska testování robustnosti serveru, tak i validace použitelnosti celé platformy v praxi.

4.4 Testování algoritmických obchodníků

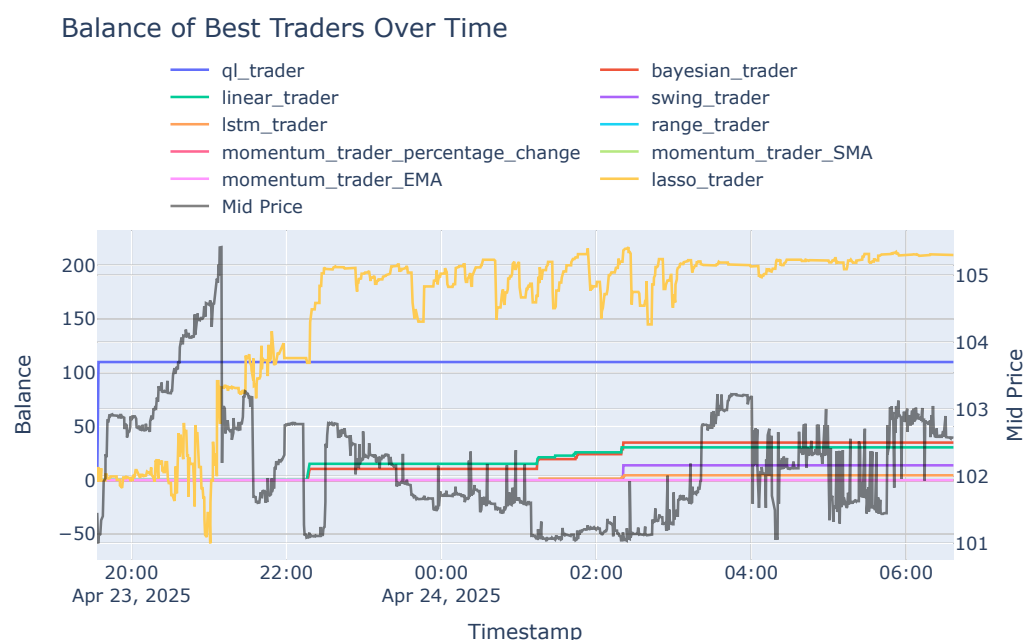
Tento experiment byl zaměřen na ověření funkčnosti a analýzu chování autonomních obchodních agentů v kontrolovaném simulovaném tržním prostředí. Testování se účastnilo celkem 14 unikátních agentů, jejichž typy strategií jsou uvedeny v Tabulce 4.3. Každý agent byl spuštěn se standardními (výchozími) parametry bez individuální optimalizace, s cílem sledovat jejich přirozené reakce na vývoj trhu a získat první náhled na jejich dynamiku v prostředí bez externích zásahů.

Počáteční rozpočet každého agenta byl nastaven na 10 000 jednotek, což umožnilo jednotné podmínky pro porovnání základní výkonnosti jednotlivých strategií. Mezi testovanými agenty byly zahrnuty všechny implementované obchodní strategie s výjimkou podvodné strategie spoofingu:

- Obchodník založený na Q-learningu,
- Obchodník využívající BayesianRidge regresi,
- Obchodník využívající lineární regresi,
- Swingový obchodník,
- Obchodník využívající LSTM neuronovou síť,
- Obchodník obchodující v cenovém rozpětí (Range Trader),
- Momentum obchodník na základě procentuální změny ceny,
- Momentum obchodník využívající jednoduchý klouzavý průměr, SMA,
- Momentum obchodník využívající exponenciální klouzavý průměr, EMA,
- Obchodník využívající LASSO regresi,
- Momentum obchodník na základě indikátoru RSI,
- Obchodník využívající Random Forest regresi,
- Obchodník využívající Ridge regresi,
- Scalping obchodník, realizující vysoký počet krátkodobých obchodů.

Cílem experimentu bylo otestovat chování různých typů algoritmických obchodních agentů v simulovaném tržním prostředí. Experiment nebyl zaměřen na přímé porovnávání výkonnosti jednotlivých agentů, protože výsledky obchodních strategií jsou zásadně ovlivňovány specifickými charakteristikami trhu, jako jsou volatilita, likvidita či dynamika cenového vývoje. Bez systematického testování napříč odlišnými scénáři by vzájemné srovnávání agentů neposkytovalo metodologicky korektní závěry.

Experimentální simulace byla opakovaně realizována s dlouhodobým časovým rozsahem v řádu několika hodin, přičemž analýza vývoje střední ceny odhalila dva převládající dynamické vzorce. Prvním byl scénář, kdy po počátečním růstu ceny následovala její výraznější oscilace v čase. Druhým vzorcem byl opět počáteční růst ceny, který však přešel do fáze stagnace, kdy se cena stabilizovala bez výraznějších změn.



Obrázek 4.2: Vývoj střední ceny a absolutních výtěžků jednotlivých agentů v průběhu simulace bez cenové stagnace

Počáteční nárůst ceny lze v obou případech vysvětlit konfigurací experimentu, v níž všichni agenti vstupovali do simulace bez držených aktiv a většina implementovaných strategií generovala poptávku po nákupu. Tento souhrnný nákupní tlak způsobil růst tržní ceny v úvodní fázi simulace.

V případech, kdy cena přešla do fáze stagnace, došlo k výraznému poklesu obchodní aktivity agentů. Tento jev lze vysvětlit skutečností, že většina testovaných strategií byla koncipována tak, aby reagovala na výraznější cenové pohyby, které

slouží jako signály k realizaci obchodů. V prostředí bez dostatečné volatility strategie negenerovaly obchodní signály, což vedlo k poklesu realizovaných obchodů.

Je třeba zároveň zdůraznit, že simulované tržní prostředí bylo tvořeno výhradně algoritmickými obchodníky a nebylo ovlivněno žádnými externími faktory, jako jsou makroekonomické události, zprávy, či behaviorální reakce reálných investorů, které by v reálném trhu významně ovlivňovaly dynamiku cen.

Scénář s cenovou oscilací

Výzkumně zajímavější se ukázal scénář, v němž se trh neustálil ve fázi stagnace, ale vykazoval dynamický cenový vývoj s výraznými oscilacemi. Z tohoto důvodu jsou dále prezentované výsledky odvozeny právě z tohoto typu průběhu. Obrázek 4.2 znázorňuje vývoj střední ceny a absolutních výdělků jednotlivých obchodních agentů. Přestože v grafu dosáhl nejvyššího absolutního zisku agent využívající lasso regresi (přibližně 209 jednotek), po započtení transakčních poplatků nejlepších výsledků dosáhl agent založený na Q-learningu, který zhodnotil svůj počáteční rozpočet o přibližně 1,09 %. Tento rozdíl je způsoben odlišnými styly obchodování: zatímco agent využívající lasso regresi uskutečnil více než 2 000 obchodů, čímž akumuloval značné transakční náklady, agent s Q-learning strategií přistupoval k obchodování selektivněji, a minimalizoval tak vliv poplatků na konečný výsledek.

Výrazný vliv transakčních poplatků ilustruje případ scalping agenta, který v rámci simulace provedl 18 547 obchodů a v důsledku vysokých nákladů zaznamenal ztrátu téměř 23 % ze svého počátečního kapitálu. Na opačném konci spektra se nachází agent využívající range trading strategii, který vzhledem k absenci dostatečně výrazných cenových pohybů v rámci stanoveného obchodního pásma (`support_level = 99`, `resistance_level = 101`) nerealizoval žádný obchod.

Z výsledků uvedených v Tabulce 4.3 je patrné, že jednotliví agenti vykazovali výrazně odlišné obchodní chování i výsledky. Některé strategie byly charakteristické velmi nízkou obchodní aktivitou, zatímco jiné realizovaly tisíce transakcí během krátkého období. Rozdílné přístupy k obchodování se promítly nejen do výnosnosti, ale zejména do citlivosti na transakční poplatky, které výrazně ovlivnily konečné bilance některých agentů. Tyto výsledky zdůrazňují, že efektivita jednotlivých strategií je silně podmíněna specifiky tržního prostředí a konkrétními charakteristikami poplatkové struktury.

Tabulka 4.3: Souhrn vybraných statistik obchodníků v experimentu bez spoofingu

	Strategie	Balance	Ret [%]	AvgVol	MaxVol	AvgBal	Ords
1	Q-learning	10109.17	1.09	54.88	55	-5519.80	37
2	Bayes. reg.	10034.05	0.34	11.19	23	-1130.97	18
3	Linear reg.	10029.61	0.30	11.09	20	-1120.54	19
4	Swing	10013.61	0.14	3.09	9	-312.67	3
5	LSTM	10003.44	0.03	1.13	3	-114.02	27
6	Range	10000.00	0.00	0.00	0	0.00	0
7	Mom. %	9998.82	-0.01	0.00	0	0.00	27
8	Mom. SMA	9998.82	-0.01	0.00	0	0.00	27
9	Mom. EMA	9998.82	-0.01	0.00	0	0.00	27
10	Lasso reg.	9958.25	-0.42	59.84	89	-5987.86	2180
11	Mom. RSI	9879.97	-1.20	83.30	89	-8541.93	3214
12	RF reg.	9829.06	-1.71	57.13	88	-5788.70	2291
13	Ridge reg.	9791.07	-2.09	65.77	88	-6723.03	2021
14	Scalping	7702.54	-22.97	36.14	82	-3971.76	18547

4.4.1 Experiment s přítomností spoofingu

Tento experiment navazoval na předchozí testování v prostředí bez přítomnosti spoofingu. Podmínky zůstaly identické – byla zachována struktura trhu, nastavení transakčních poplatků i počáteční rozpočty standardních agentů. Jedinou změnou bylo zavedení specializovaného spoofingového agenta, jehož úkolem bylo aktivně manipulovat tržními podmínkami prostřednictvím vkládání a následného rušení velkých falešných objednávek. Aby bylo možné v simulaci účinně aplikovat spoofingové techniky, byl na základě expertního doporučení agentovi přiřazen vyšší počáteční rozpočet 30 000 jednotek a obchodní objem 3 000 jednotek.

Aby bylo možné objektivně porovnat výkonnost jednotlivých agentů, byly všechny hodnoty následně přepočítány, čímž bylo zajištěno objektivní porovnání výkonnosti agentů, nezávisle na rozdílných počátečních podmínkách. Výsledky jsou shrnuty v Tabulce 4.4.

Výsledky experimentu ukazují, že spoofingový agent dosáhl návratnosti 28,24 %, čímž výrazně překonal všechny ostatní obchodníky v simulovaném trhu. Tento výsledek potvrzuje efektivitu agresivního chování založeného na manipulaci tržními signály v testovaném prostředí. Spoofingový agent systematicky vkládal a rušil velké objednávky v těsné blízkosti aktuálních cen, čímž vytvářel iluzorní tlak na růst či pokles ceny a dezorientoval ostatní algoritmické obchodníky.

Tabulka 4.4: Souhrn vybraných statistik obchodníků v experimentu se spoofingem

	Strategie	Balance	Ret [%]	AvgVol	MaxVol	AvgBal	Ords
1	Spoofing	12824.29	28.24	2998.25	3025	188.54	3449
2	Mom. %	10106.28	1.06	14.74	90	-1395.64	2348
3	Ridge reg.	10018.05	0.18	11.19	56	-1081.61	1427
4	Swing	10015.27	0.15	56.10	89	-5656.04	492
5	Lasso reg.	10013.63	0.14	11.92	57	-1168.30	1408
6	Range	10000.00	0.00	0.00	0	0.00	0
7	Mom. RSI	9998.64	-0.01	0.00	0	0.00	36
8	Scalping	9996.57	-0.03	19.95	89	-1889.57	5611
9	LSTM	9987.88	-0.12	6.09	89	-613.63	1612
10	Mom. EMA	9953.34	-0.47	8.39	85	-839.46	2926
11	Bayes. reg.	9948.33	-0.52	12.33	82	-1237.38	1406
12	Mom. SMA	9911.47	-0.89	13.12	89	-1319.04	3456
13	RF reg.	9888.95	-1.11	11.98	75	-1228.40	1383
14	Linear reg.	9849.10	-1.51	11.87	75	-1230.03	1389
15	Q-learning	9216.59	-7.83	28.03	89	-3112.65	2477

Je však nezbytné interpretovat tyto výsledky s vědomím určitých omezení. Simulace probíhala v umělém a značně zjednodušeném prostředí, kde tržní dynamiku určovali výhradně předem definovaní algoritmičtí agenti s omezenou komplexitou strategií. Experiment nezohledňoval řadu klíčových faktorů, které ovlivňují reálné finanční trhy, jako jsou makroekonomické události, regulační zásahy, heterogenita účastníků trhu, behaviorální aspekty lidských obchodníků či komplexní likviditní dynamika. Výsledky proto nelze přímo extrapolovat na reálné finanční trhy, ale spíše je chápat jako ilustraci základních principů manipulativního chování v řízeném experimentálním rámci.

Navzdory těmto omezením má podobné testování významnou hodnotu. Simulace umožňují lépe pochopit, jak mohou tržní manipulace ovlivnit chování běžných obchodních strategií a jaké systémové důsledky může mít i přítomnost relativně malé skupiny manipulujících subjektů. Studium takových fenoménů je klíčové nejen pro predikci potenciálních tržních anomálií, ale i pro návrh opatření zvyšujících robustnost a odolnost burzovních systémů vůči manipulaci.

Tato diplomová práce se zabývala návrhem a implementací *simulačního prostředí* pro modelování obchodování na trzích řízených *limitní knihou objednávek*. Cílem bylo vytvořit *robustní* a *modulární* platformu, která umožní detailní analýzu tržních mechanismů, testování různorodých obchodních strategií a demonstraci specifických fenoménů v prostředí finančních trhů.

Výsledkem je funkční *simulátor* postavený na architektuře klient–server s využitím standardizovaného komunikačního protokolu FIX. Jádrem systému je *párovací algoritmus* založený na prioritě ceny a času (**Price–Time Priority**), který zajišťuje realistické vypořádávání limitních objednávek. Simulátor je doplněn o webové uživatelské rozhraní pro manuální interakci s trhem a reprezentativní sadu obchodních agentů. Speciální důraz byl kladen na implementaci manipulativního agenta, jehož cílem bylo ilustrovat dopady nelegálního chování na tržní prostředí.

Součástí práce bylo dále provedení experimentálních simulací, které demonstrovaly vliv jednotlivých strategií na dynamiku trhu, schopnost agentů adaptovat se na měnící se podmínky a důsledky tržní manipulace. Výsledky ukázaly, že i v kontrolovaném a zjednodušeném prostředí může manipulativní chování, reprezentované *spoofingem*, výrazně narušit predikovatelnost tržního vývoje a snížit výkonnost standardních obchodních strategií. Přestože simulace vykazují určitá omezení vyplývající z abstrakce reálného trhu a jednoduchosti modelovaných agentů, poskytují cenný vhled do základních mechanismů interakce účastníků trhu a možných rizik spojených s manipulací.

Kromě toho byly provedeny zátěžové testy systému, jejichž výsledky potvrdily vysokou optimalizaci architektury jak pro běžný provoz, tak pro extrémní podmínky. Při standardní zátěži systém vykazoval minimální latenci v jednotkách milisekund a bezchybné zpracování požadavků.

Prínosem práce je návrh a realizace plně funkčního a snadno rozšiřitelného *simulačního rámce*, který může sloužit jako výzkumný a výukový nástroj v oblasti agentního modelování, algoritmického obchodování a tržní mikrostruktury. Díky

své flexibilitě umožňuje nejen srovnávání obchodních strategií, ale také simulaci a analýzu vlivu různých typů tržního chování – včetně manipulativních praktik – na stabilitu a efektivitu trhu. Práce tak přispívá k lepšímu porozumění dynamice limitních trhů a poskytuje praktický základ pro další výzkum v oblasti ekonomických simulací, multiagentních systémů a etického rámce algoritmického obchodování.

Do budoucna lze systém rozšiřovat v několika směrech. Jednou z klíčových oblastí je rozšíření podpory o různé typy párovacích mechanismů, které by umožnily simulaci širší škály tržních prostředí a chování. Perspektivní je také rozvoj pokročilých modelů tvorby trhu s cílem realističtější replikace dynamiky reálných trhů a zavedení mechanismů generování externích událostí ovlivňujících tržní vývoj. Dalším přirozeným směrem rozvoje je zvyšování inteligence obchodních agentů, a to zejména prostřednictvím pokročilejších metod strojového učení a adaptivního rozhodování. Významný potenciál představuje také vylepšení grafického uživatelského rozhraní, a to jak z hlediska rozšíření funkcionalit, tak z pohledu zlepšení vizualizace komplexních datových toků a obchodování s více produkty.

Do budoucna může simulátor sloužit nejen jako edukativní platforma, ale také jako nástroj pro podporu výzkumu v oblasti tržní dynamiky a detekce podvodného chování na finančních trzích.

Instalace a spuštění simulátoru



Tato kapitola slouží jako technická příručka pro instalaci, konfiguraci a spuštění softwarového systému pro simulaci obchodování na trhu řízeném knihou limitních objednávek. Celý systém byl implementován v jazyce **Python** ve verzi 3.9.7, přičemž z důvodu kompatibility s použitými knihovnami je doporučeno využívat právě tuto verzi.

Poznámka: Repozitář je dostupný na adrese:

https://github.com/Jivl00/Exchange_simulator_for_the_limit_order-driven_market

Stažení a příprava repozitáře

Zdrojový kód projektu je veřejně dostupný na platformě **GitHub**. Pro získání aktuální verze kódu lze využít následující příkaz:

```
1 uživatel@pc: $ git clone  
https://github.com/Jivl00/Exchange_simulator_for_the_limit_order-driven_market.git  
2 uživatel@pc: $
```

Instalace závislostí

Veškeré softwarové závislosti jsou deklarovány v souboru `requirements.txt`. K jejich instalaci je možné využít správce balíčků `pip`:

```
1 uživatel@pc: /Exchange_simulator_for_the_limit_order-driven_market$ pip  
install -r requirements.txt
```

Struktura projektu a konfigurační soubory

Všechny skripty a moduly je nutné spouštět ze složky `src`, aby bylo zajištěno korektní načítání vnitřních závislostí. Konfigurace serverové části je definována v souboru `config/server_config.json`, kde lze nastavit:

- IP adresu serveru,
- komunikační porty,
- alternativní routy pro přístup k API.

Spuštění serverové části

Obchodní server lze spustit pomocí následujícího skriptu:

```
1 uživatel@pc: /Exchange_simulator_for_the_limit_order-driven_market/src$  
python server/server.py
```

Serverová aplikace po spuštění automaticky vytvoří interní databázi, pokud není dosud přítomna. Po ukončení simulačního běhu dojde k uložení výsledků do složky `data`.

Načtení serverem uloženého stavu a pokračování v simulaci lze zajistit pomocí přepínače `-l`:

```
1 uživatel@pc: /Exchange_simulator_for_the_limit_order-driven_market/src$  
python server/server.py -l
```

Pro účely rozšíření funkcionality simulačního prostředí je možné využít volitelné pomocné moduly dostupné ve složce `server/agents`. Tyto komponenty umožňují dynamické ovlivnění trhu v průběhu simulace:

- **market maker** – zajišťuje předvyplnění knihy limitních objednávek, čímž simuluje počáteční likviditu trhu,
- **liquidity generator** – periodicky generuje nové objednávky, čímž udržuje stabilní úroveň tržní aktivity.

Spuštění těchto modulů se provádí ze základního adresáře `src`, například:

```
1 uživatel@pc: /Exchange_simulator_for_the_limit_order-driven_market/src$  
python server/agents/market_maker.py
```


Spuštění obchodních agentů

Obchodní agenti jsou definováni jako samostatné třídy ve složce `client/agents`, přičemž každý soubor reprezentuje nezávislou implementaci obchodní strategie. Při inicializaci jednotlivých agentů je možné měnit parametry konkrétní strategie, jako například velikost klouzavého okna nebo frekvenci zadávání objednávek.

Agenta lze spustit následujícím způsobem:

```
1 uživatel@pc: /Exchange_simulator_for_the_limit_order-driven_market/src$
  python client/agents/nazev_agenta.py
```

Spuštění webového rozhraní

Grafické uživatelské rozhraní systému je realizováno jako samostatná komponenta. Lze jej spustit pomocí skriptu:

```
1 uživatel@pc: /Exchange_simulator_for_the_limit_order-driven_market/src$
  python viz/main_page.py
```

Analýza výsledků simulace

Pro komplexní analýzu výstupních dat byl připraven **Jupyter Notebook**, který umožňuje generování přehledných vizualizací a statistických souhrnů. Notebook se nachází ve složce:

`viz/report/report.ipynb`

Notebook využívá výstupní data ze složky `data` a umožňuje například:

- porovnávat výkonnost jednotlivých strategií,
- analyzovat objemy a četnost obchodů,
- hodnotit úspěšnost obchodních strategií.

Testování systému

Součástí systému je i základní sada testů ověřujících funkčnost jednotlivých komponent. Testy lze spustit z adresáře `tests` pomocí následujícího příkazu:

```
1 uživatel@pc: /Exchange_simulator_for_the_limit_order-driven_market/tests$
  python -m unittest tests.py
```


Uživatelská příručka pro použití webového rozhraní

B

V rámci testování vyvíjeného systému a jeho prezentace širší skupině uživatelů byla vytvořena uživatelská příručka v anglickém jazyce, jejímž cílem bylo usnadnit orientaci v grafickém rozhraní a zpřístupnit základní principy obchodování i méně technicky orientovaným účastníkům. Tato příručka vznikla jako součást tematicky laděné velikonoční burzovní soutěže, jež sloužila nejen k demonstraci funkcionality systému v reálném čase, ale také jako prostředek pro sběr uživatelské zpětné vazby, která následně přispěla k dalšímu vylepšení aplikace.

Navzdory svému částečně neformálnímu tónu a popularizačnímu zaměření představuje tento dokument plnohodnotný a přehledný popis jednotlivých komponent uživatelského rozhraní, obchodních funkcí i základních principů interakce se systémem. Z tohoto důvodu je originální znění této příručky zařazeno do příloh této práce jako doplňkový dokument, který poskytuje čtenáři praktický pohled na konkrétní podobu rozhraní a způsob jeho využití v simulovaném prostředí.

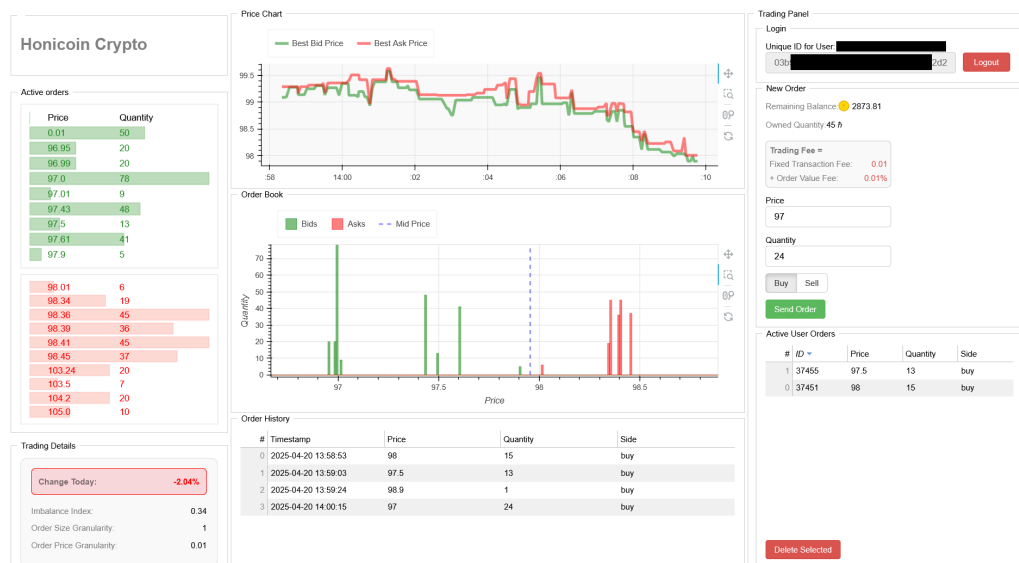
Welcome to Honicoín Crypto

Welcome to **Honicoín Crypto**, your hands-on trading simulator designed to give you a realistic experience of market mechanics. This guide will walk you through the different components of the interface and how to place, manage, and understand trades.

Before diving into the exciting world of stock trading, you'll need to register with us. To begin, simply provide your email and set up a password. Make sure to choose a password you can remember, as there is no password recovery option available. Also, remember that the email you provide is important—your trading results will be sent to that email. So, please ensure that it is a valid email address that you have access to!

Interface Overview

The simulator interface is divided into several panels, each offering critical trading information, as shown in Figure B.1.



Obrázek B.1: Overview of the Honicoín Crypto simulator interface, showing the various panels containing trading information.

Active Orders (Left Panel)

This panel displays all current **buy (green)** and **sell (red)** orders. It provides an overview of the market's ongoing activity, showing the prices and quantities at

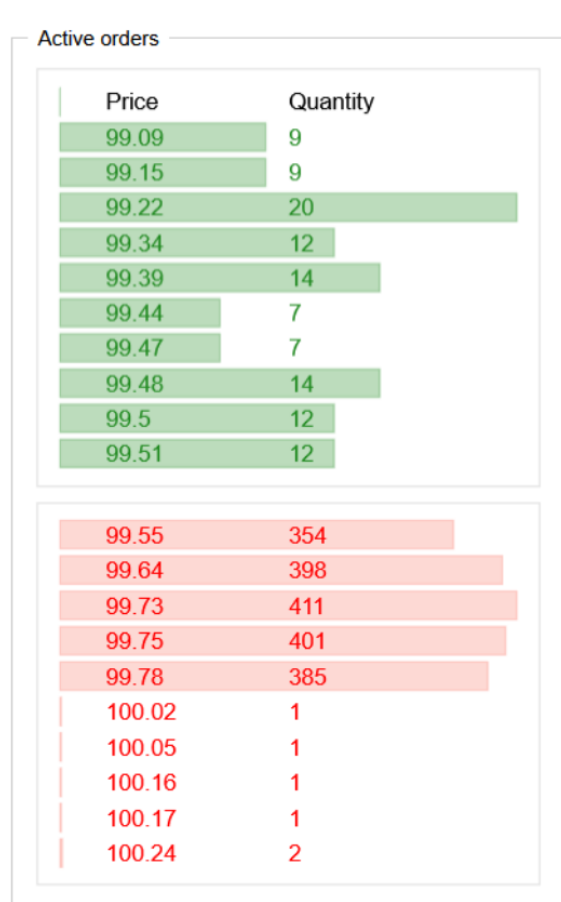
which buyers and sellers are willing to transact. An example of this is shown in Figure B.2.

- **Price:** The price per stock unit that is being offered by buyers and sellers.
- **Quantity:** The number of shares available at the specified price.

Buy orders (green) are the prices at which buyers are willing to purchase stocks.

Sell orders (red) are the prices at which sellers are willing to sell their stocks.

Bid price refers to the highest price that buyers are willing to pay for a stock. **Ask price** refers to the lowest price that sellers are willing to accept for a stock.



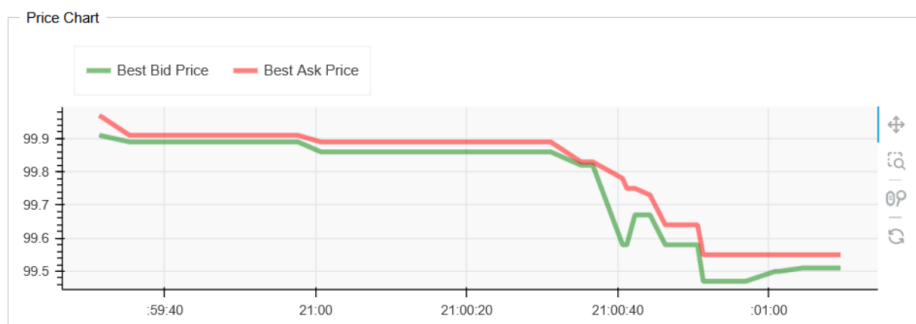
Obrázek B.2: Active Orders

Price Chart (Top-Center Panel)

The price chart displays real-time market data, showing the movement of stock prices over time. The chart includes:

- **Green Line:** The Best Bid Price, which is the highest price buyers are offering for the stock.
- **Red Line:** The Best Ask Price, which is the lowest price sellers are willing to accept for the stock.

An example of the price chart is shown in Figure B.3.



Obrázek B.3: Price Chart

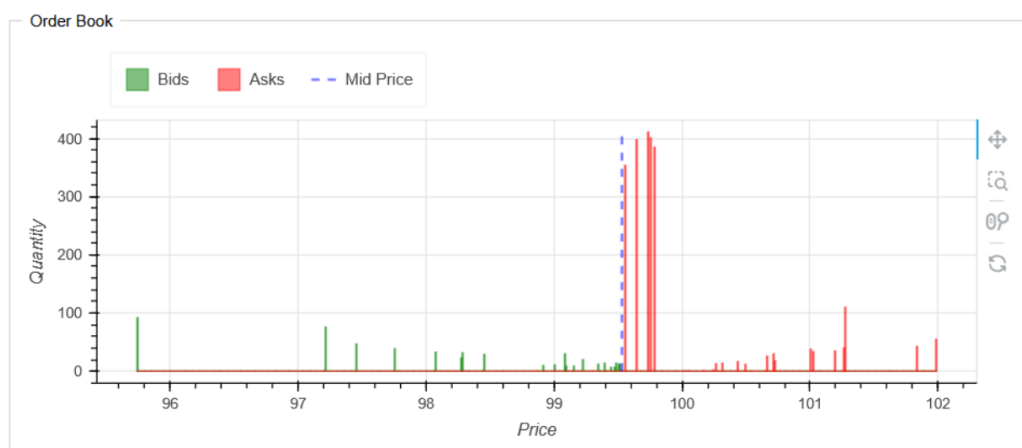
Use this chart to track market trends and to time your entries and exits into the market. The closer the bid and ask prices are to each other, the higher the liquidity in the market, meaning it is easier to buy and sell shares quickly.

Order Book (Center-Lower Panel)

The Order Book shows a visual representation of all current buy and sell orders:

- **Green Bars:** Represent Buy Orders (Bids), showing the volume and price at which buyers want to purchase.
- **Red Bars:** Represent Sell Orders (Asks), showing the volume and price at which sellers want to sell.
- **Blue Dashed Line:** Represents the Mid Price, which is the average price between the Best Bid and Best Ask.

Figure B.4 shows the layout of the Order Book, highlighting the buy and sell orders along with the Mid Price.

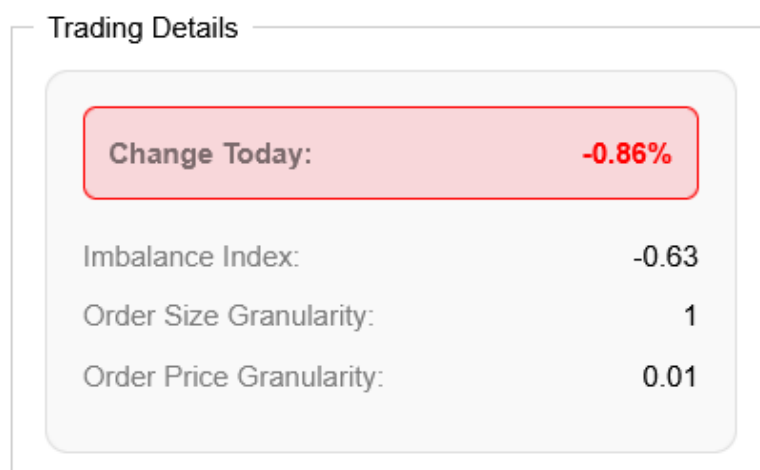


Obrázek B.4: Order Book

The Order Book helps you understand the market depth, indicating how much stock is available at different price levels.

Trading Details (Bottom-Left Panel)

This section provides quick stats that help you assess the current market conditions, as shown in Figure B.5:



Obrázek B.5: Trading Details

- **Change Today:** This shows the percentage change in the stock price from the start of the day. It reflects how much the price has increased or decreased.
- **Imbalance Index:** This metric measures the imbalance between buy and sell orders. A positive imbalance indicates that there are more buy orders than sell orders, while a negative imbalance indicates more sell orders.

- **Granularity Settings:** This defines the level of precision with which prices and order quantities can be set within the simulator. It determines how finely you can adjust prices and quantities when placing orders.

Order History (Bottom-Center Panel)

This section logs the details of recent trades that have been executed, allowing you to review past transactions and gain insight into recent market activity. As shown in Figure B.6, the Order History displays the most recent trades, including the time of execution, price, and quantity of shares traded.

Note that the Order History is only available for the current session and will be cleared upon page refresh. It is not stored permanently, so if you refresh or close the page, the history will be reset.

Order History				
#	Timestamp	Price	Quantity	Side
0	2025-04-09 21:00:36	100	10	buy
1	2025-04-09 21:00:40	100	10	buy
2	2025-04-09 21:00:42	100	10	buy
3	2025-04-09 21:00:50	99	11	buy
4	2025-04-09 21:00:56	99.8	12	buy
5	2025-04-09 21:01:00	99.5	12	buy

Obrázek B.6: Order History

Trading Panel (Right Panel)

This section contains the tools needed to interact with the market, including your account details, balance, and the ability to place new orders. As shown in Figure B.7, you can use this panel to manage your trades, track your balance, and enter new orders.

Trading Panel

Login


Unique ID for User:

03b

2d2

Logout

New Order

Remaining Balance:  2873.81

Owned Quantity: 45 ħ

Trading Fee =

Fixed Transaction Fee: 0.01

+ Order Value Fee: 0.01%

Price

97

Quantity

24

Buy

Sell

Send Order

Active User Orders

#	ID ▾	Price	Quantity	Side
1	37455	97.5	13	buy
0	37451	98	15	buy

Delete Selected

Obrázek B.7: Trading Panel

Login

Displays your **Unique User ID**, which is used to track your trading activity. It can be used for algorithmic trading via the **Python** client (which will be explained later). Keep your ID secure to ensure the safety of your trading activities and account data.

Account Stats

This panel shows:

- **Balance:** The amount of funds remaining in your account.
- **Owned Quantity:** The number of stocks you currently hold in your portfolio. Remember, you cannot sell stocks you do not own. Ensure that you have sufficient shares in your portfolio before attempting to place a sell order.

New Order Section

To place a trade, follow these steps:

1. Enter the **Price** at which you wish to buy or sell.
2. Enter the **Quantity** (number of shares) you wish to buy or sell.
3. Select whether you want to **Buy** or **Sell**.
4. Click **Send Order** to place the trade.

Fees: Please note that each trade incurs fees, which are calculated based on various factors such as the transaction amount and the number of shares involved. Be sure to review the fee structure before placing trades.

Active User Orders

This section displays all your currently open orders that have not yet been executed. Each order is shown with the following details:

- **ID:** A unique identifier assigned to each order for tracking purposes. This helps you distinguish between multiple orders in case you need to make adjustments or cancellations.

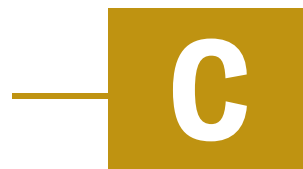
- **Price:** The price at which you have set the order to be executed. This is the price you are willing to buy or sell the stock for. Orders will remain open until the market price matches your set price, or until you decide to cancel them.
- **Quantity:** The number of shares you wish to buy or sell at the specified price. This represents the amount of stock being offered or sought at that price. If the order is partially filled, the remaining quantity will stay open and continue to be active until it is fully executed or cancelled. The quantity will decrease as parts of the order are filled, but the order will remain active as long as there is any remaining quantity.
- **Side:** Indicates whether the order is a **Buy** or a **Sell** order. A buy order means you are looking to purchase stock at the specified price, while a sell order means you are offering your stock for sale at that price.

You have the ability to manage your open orders directly from this section. If you wish to cancel an order, simply select the order from the list and click on the **Delete Selected** button. This will remove the order from the system, and it will no longer be active in the market. It is important to review your open orders regularly to ensure that they align with your trading strategy and current market conditions.

Tips for Using the Simulator

- **Set Realistic Price Targets:** Establish clear price targets based on technical analysis or market sentiment. Set buy orders slightly below the current price in anticipation of a dip, or place sell orders above the current price if you expect a rally. Avoid chasing prices too aggressively, as this can lead to unfavourable entry points.
- **Start Small and Learn Gradually:** Begin with small trades and gradually increase your exposure as you gain experience. The simulator is a great way to practice, but real success comes with understanding the mechanics of trading over time. Patience is key.
- **Stay Calm and Patient:** The market can be volatile, and price movements can happen quickly. Stay calm and avoid making impulsive decisions based on short-term fluctuations. Sometimes, it's best to wait for confirmation before acting, rather than rushing into trades.

Specifikace modifikovaného FIX protokolu



V následující Tabulce C.1 jsou uvedeny všechny podporované zprávy, které jsou součástí implementovaného protokolu. Každá zpráva je popsána s ohledem na její typ, směr komunikace (klient → server nebo server → klient) a stručný popis jejího účelu a struktury. Tento seznam slouží jako přehled základních typů zpráv.

Tabulka C.1: Seznam podporovaných typů zpráv

Název zprávy	MsgType	Směr	Popis
RegisterRequest	A	Klient → Server	Registrace uživatele
InitializeLiquidityEngine	A	Klient → Server	Založení admin providera likvidity
OrderStatusRequest	H	Klient → Server	Dotaz na stav konkrétní objednávky
NewOrderSingle	D	Klient → Server	Zadání nové objednávky
OrderCancelRequest	F	Klient → Server	Požadavek na zrušení objednávky
OrderModifyRequestQty	G	Klient → Server	Požadavek na úpravu množství
MarketDataRequest	V	Klient → Server	Požadavek aktuální limitní knihy
UserOrderStatusRequest	AF	Klient → Server	Dotaz na všechny objednávky uživatele
UserBalanceRequest	BB	Klient → Server	Dotaz na zůstatek
CaptureReportRequest	AD	Klient → Server	Dotaz na historii obchodů

RegisterResponse	A	Server → Klient	Potvrzení registrace
OrderStatus	8	Server → Klient	Stav konkrétní objednávky
ExecutionReport	8	Server → Klient	Výsledek zpracování objednávky
ExecutionReportCancel	8/9	Server → Klient	Potvrzení/zamítnutí zrušení objednávky
ExecutionReportModify	8/9	Server → Klient	Potvrzení/zamítnutí úpravy objednávky
MarketDataSnapshot	W	Server → Klient	Stav trhu
UserOrderStatus	8	Server → Klient	Přehled objednávek uživatele
UserBalance	BA	Server → Klient	Zůstatek na účtu
CaptureReport	AE	Server → Klient	Historie obchodů

B. Detailní popis vybraných zpráv

Název zprávy: NewOrderSingle

Strana komunikace: Klient → Server

Popis: Zpráva pro zadání nové objednávky. Použité FIX tagy:

- 35=D (MsgType)
- 54=1/2 (buy/sell)
- 38=quantity
- 44=price
- 55=product (Symbol)

Název zprávy: OrderCancelRequest

Strana komunikace: Klient → Server

Popis: Zpráva pro zrušení existující objednávky. Použité FIX tagy:

- 35=F (MsgType)
- 41=OrigClOrdID
- 55=product (Symbol)

Název zprávy: **MarketDataRequest**

Strana komunikace: Klient → Server

Popis: Zpráva slouží k požadavku na aktuální snapshot trhu (order book). Použité FIX tagy:

- 35=V (MsgType)
- 55=product
- 263=0 (snapshot only)
- 264=depth

Název zprávy: **MarketDataSnapshot**

Strana komunikace: Server → Klient

Popis: Odpověď obsahující aktuální stav trhu. Order book je předáván jako JSON v tagu 58. Použité FIX tagy:

- 35=W (MsgType)
- 55=product
- 58=orderBook (serializovaný JSON)

C. Poznámky k implementaci

- Generování a parsování FIX zpráv je zajištěno knihovnou `simplefix`, která automaticky validuje strukturu zpráv a kontroluje povinné tagy podle specifikace protokolu.
- Ve specifických případech byly některé standardní tagy z původní specifikace FIX protokolu vynechány nebo upraveny, aby lépe odpovídaly požadavkům simulátoru.
- Komplexní datové struktury, jako jsou limitní kniha objednávek, historické záznamy a zůstatky, jsou serializovány do formátu JSON a odesílány prostřednictvím tagu 58 (text) pro zjednodušení.

Bibliografie

1. JAIN, Konark; FIROOZYE, Nick; KOCHEMS, Jonathan; TRELEAVEN, Philip. *Limit Order Book Simulations: A Review*. 2024. Dostupné z arXiv: 2402.17359.
2. ZARE, Mohammad; ARJMAND, Omid Naghshineh; SALAVATI, Erfan; MOHAMMADPOUR, Adel. An Agent-Based model for Limit Order Book: Estimation and simulation. *International Journal of Finance & Economics*. 2021, roč. 26, č. 1, s. 1112–1121. Dostupné z doi: 10.1002/ijfe.1839.
3. WANG, Xintong; WELLMAN, Michael P. Spoofing the Limit Order Book: An Agent-Based Model. In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2017, 651–659. AAMAS '17.
4. BOER-SORBAN, Katalin. *Agent-Based Simulation of Financial Markets: A Modular, Continuous-time Approach*. 2008. Dostupné z URL: <http://hdl.handle.net/1765/10870>. Dis. pr. Erasmus Research Institute of Management.
5. ALVES, Thiago W.; FLORESCU, Ionut; CALHOUN, George; BOZDOG, Dragos. *SHIFT: A Highly Realistic Financial Market Simulation Platform*. 2020. Dostupné z arXiv: 2002.11158.
6. CLIFF, Dave. *BSE: A Minimal Simulation of a Limit-Order-Book Stock Exchange*. 2018. Dostupné z arXiv: 1809.06027 [cs.CE].
7. BERTI, Leonardo; PRENKAJ, Bardh; VELARDI, Paola. *TRADES: Generating Realistic Market Simulations with Diffusion Models*. 2025. Dostupné z arXiv: 2502.07071 [q-fin.TR].
8. ABERGEL, F.; ANANE, M.; CHAKRABORTI, A.; JEDIDI, A.; I.M., Toke. *Limit Order Books*. Cambridge University Press, 2016. PHYSICS OF SOCIETY: ECONOPHYSI. ISBN 9781107163980. Dostupné z doi: 10.1017/CB09781107163980.
9. GOULD, Martin D. et al. Limit order books. *Quantitative Finance*. 2013, roč. 13, č. 11, s. 1709–1742. Dostupné z doi: 10.1080/14697688.2013.803148.

10. LI, Mengyu; QI, Tiange; HUANG, Yongjian; FENG, Panpan; ZHAO, Yingying. Can call auction reduce closing price manipulation in the stock market? *Asia-Pacific Journal of Accounting & Economics*. 2024, roč. 0, č. 0, s. 1–18. Dostupné z DOI: 10.1080/16081625.2024.2336116.
11. INVESTOPEDIA. *Adjusted Closing Price Definition*. 2023. Dostupné z URL: https://www.investopedia.com/terms/a/adjusted_closing_price.asp. [Citováno: 7. 4. 2025].
12. KYLE, Albert S. Continuous Auctions and Insider Trading. *Econometrica* [online]. 1985, roč. 53, č. 6, s. 1315–1335 [cit. 2025-01-20]. ISSN 00129682, ISSN 14680262. Dostupné z URL: <http://www.jstor.org/stable/1913210>.
13. RUBISOV, Anton D. *Statistical Arbitrage Using Limit Order Book Imbalance*. 2015. Dostupné z URL: <https://api.semanticscholar.org/CorpusID:55130442>.
14. PREIS, Tobias. Price-Time Priority and Pro Rata Matching in an Order Book Model of Financial Markets. In: *Econophysics of Order-driven Markets: Proceedings of Econophys-Kolkata V*. Ed. ABERGEL, Frédéric; CHAKRABARTI, Bikas K.; CHAKRABORTI, Anirban; MITRA, Manipushpak. Milano: Springer Milan, 2011, s. 65–72. ISBN 978-88-470-1766-5. Dostupné z DOI: 10.1007/978-88-470-1766-5_5.
15. BLACKBURN, Douglas W; GOETZMANN, William N; UKHOV, Andrey D. Trading behavior of style and multi-style investors. *Unpublished working paper. Yale School of Management*. 2011.
16. LIEN, K. *Day Trading and Swing Trading the Currency Market: Technical and Fundamental Strategies to Profit from Market Moves*. Wiley, 2008. Day Trading & Swing Trading the Currency Market: Technical &. ISBN 9780470377369. Dostupné z DOI: 10.1002/9781119212997.
17. NORRIS, Emily. *Scalping: How small, quick profits can add up*. Investopedia, 2024. Dostupné z URL: <https://www.investopedia.com/articles/trading/05/scalping.asp>. [Citováno: 24. 1. 2025].
18. THOMPSON, Cedric. *What is arbitrage? definition, example, and costs*. Investopedia, 2024. Dostupné z URL: <https://www.investopedia.com/ask/answers/what-is-arbitrage/>. [Citováno: 24. 1. 2025].
19. HAYES, Adam. *Gap trading: How to play the gap*. Investopedia, 2024. Dostupné z URL: <https://www.investopedia.com/articles/trading/05/playinggaps.asp>. [Citováno: 24. 1. 2025].

20. HAYES, Adam. *Understanding trend analysis and trend trading strategies*. Investopedia, 2024. Dostupné z URL: <https://www.investopedia.com/terms/t/trendanalysis.asp>. [Citováno: 24. 1. 2025].
21. GRATTON, Peter. *Introduction to momentum trading*. Investopedia, 2024. Dostupné z URL: <https://www.investopedia.com/trading/introduction-to-momentum-trading/>. [Citováno: 24. 1. 2025].
22. AXI. *Trading Strategies*. 2025. Dostupné z URL: <https://www.axi.com/eu/blog/education/trading-strategies>. [Citováno: 3. 2. 2025].
23. SHARPE, William F. The Sharpe Ratio. *The Journal of Portfolio Management*. 1994, roč. 21, č. 1, s. 49–58. Dostupné z DOI: 10.3905/jpm.1994.409501.
24. SUTHAR, Jyotsna. *Quote stuffing*. WallStreetMojo, 2024. Dostupné z URL: <https://www.wallstreetmojo.com/quote-stuffing/>. [Citováno: 7. 2. 2025].
25. PARLOUR, Christine A.; SEPPI, Duane J. CHAPTER 3 - Limit Order Markets: A Survey. In: THAKOR, Anjan V.; BOOT, Arnoud W.A. (ed.). *Handbook of Financial Intermediation and Banking*. San Diego: Elsevier, 2008, s. 63–96. Handbooks in Finance. ISSN 15684997. Dostupné z DOI: 10.1016/B978-044451558-2.50007-6.
26. GODE, Dhananjay K.; SUNDER, Shyam. Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality. *Journal of Political Economy* [online]. 1993, roč. 101, č. 1, s. 119–137 [cit. 2025-01-21]. ISSN 00223808, ISSN 1537534X. Dostupné z URL: <http://www.jstor.org/stable/2138676>.
27. TVERSKY, Amos; KAHNEMAN, Daniel et al. *Choices, values, and frames*. Cambridge University Press New York, 2000.
28. CONT, Rama; STOIKOV, Sasha; TALREJA, Rishi. A Stochastic Model for Order Book Dynamics. *Operations Research* [online]. 2010, roč. 58, č. 3, s. 549–563 [cit. 2025-01-21]. ISSN 0030364X, ISSN 15265463. Dostupné z URL: <http://www.jstor.org/stable/40792679>.
29. SMITH, Eric; FARMER, J Doyne; GILLEMOT, László; KRISHNAMURTHY, Supriya. Statistical theory of the continuous double auction. *Quantitative Finance*. 2003, roč. 3, č. 6, 481–514. ISSN 1469-7696. Dostupné z DOI: 10.1088/1469-7688/3/6/307.
30. COMMUNITY, FIX Trading. *FIX Trading Community*. 2017-2025. Dostupné z URL: <https://www.fixtrading.org>. [Citováno: 15. 1. 2025].
31. SWIFT. *SWIFT: The global provider of secure financial messaging services*. 2024. Dostupné z URL: <https://www.swift.com/>. [Citováno: 15. 1. 2025].

32. DEMARCO, Darren. *Exploiting Financial Information Exchange (FIX) Protocol*. 2012. Dostupné z URL: <https://www.giac.org/paper/gcih/20359/exploiting-financial-information-exchange-fix-protocol/126181>. GIAC (GCIH) Gold Certification.
33. WOOLDRIDGE, Michael. *An Introduction to MultiAgent Systems*. 2nd. Wiley Publishing, 2009. ISBN 0470519460.
34. RUSSELL, Stuart; NORVIG, Peter. *Artificial Intelligence: A Modern Approach*. 3. vyd. Prentice Hall, 2010.
35. PARASUMANNA GOKULAN, Balaji; SRINIVASAN, D. An Introduction to Multi-Agent Systems. In: 2010, sv. 310, s. 1–27. ISBN 978-3-642-14434-9. Dostupné z DOI: 10.1007/978-3-642-14435-6_1.
36. PRING, M.J. *Technical Analysis Explained*. McGraw-Hill, 2002. ISBN 9780071396127. Dostupné z URL: <https://books.google.cz/books?id=Qfr-OwAACAAJ>.
37. CLIFTON, Jesse; LABER, Eric. Q-Learning: Theory and Applications. *Annual Review of Statistics and Its Application*. 2020, roč. 7, č. Volume 7, 2020, s. 279–301. ISSN 2326-831X. Dostupné z DOI: 10.1146/annurev-statistics-031219-041220.

Seznam obrázků

2.1	Ukázková zpráva ve FIX protokolu	21
2.2	Diagram znázorňující systém FIX (převzato z [32])	22
3.1	Schéma architektury simulátoru	30
3.2	UML diagram klientské aplikace	35
3.3	Schematické znázornění obchodní strategie Range Trader v rámci cenového pásma.	40
3.4	Schematické znázornění obchodní logiky využívající Bollingerova pásma	44
3.5	Navržené webové uživatelské rozhraní pro manuální obchodování	49
4.1	Vývoj střeny ceny a výtělků zúčastněných obchodníků během prvního dne velikonoční hry	57
4.2	Vývoj střední ceny a absolutních výtělků jednotlivých agentů v průběhu simulace bez cenové stagnace	61
B.1	Overview of the Honicoín Crypto simulator interface, showing the various panels containing trading information.	72
B.2	Active Orders	73
B.3	Price Chart	74
B.4	Order Book	75
B.5	Trading Details	75
B.6	Order History	76
B.7	Trading Panel	77

Seznam tabulek

4.1	Shrnutí výsledků zátěžových testů serveru při standardní zátěži (100 uživatelů) a vysoké zátěži (2000 uživatelů)	55
4.2	Souhrn vybraných statistik obchodníků za první den experimentu . .	57
4.3	Souhrn vybraných statistik obchodníků v experimentu bez spoofingu .	63
4.4	Souhrn vybraných statistik obchodníků v experimentu se spoofingem	64
C.1	Seznam podporovaných typů zpráv	81

1101001 1100001
10101100001110010 1100001
101011010101 10



11010011101101001
0110000110101
111000101011101