Automated Trading Bot: Feinkonzept

Juri Stoffers[a,1]

[a] *Gymthun*

Begleitperson: Dr. Ostrin Geoffrey

1. My Goals & Requirements

My primary goal is to develop and test a trading strategy that can be deployed on real markets with minimal risk of capital loss. During the live testing phase, the focus is not on maximizing profit, but on verifying the robustness, stability, and correct execution of the system under real-time conditions.

I concentrate solely on Bitcoin, as it combines high liquidity with around-the-clock market access and excellent data availability. Crypto exchanges provide free access to high-resolution data, including order book snapshots and liquidation feeds, which are crucial for the types of strategies I intend to develop.

Key requirements for my system include:

- Capital preservation during live testing
- Modular architecture for easy debugging and updates
- Reliable execution via API (Hyperliquid)
- Real-time performance monitoring and logging
- Strategy transparency and ease of evaluation

2. My System Design

2.1. Data Sources

I gather both real-time and historical market data to feed my bot, ensuring I have enough coverage for backtesting and live operations.

2.2. Signal Processing

I develop modules to analyze data for trade signals, whether it's technical indicators (e.g., EMA, Orderbook Delta) or more advanced methods (machine learning).

2.3. Order Management

I connect to broker APIs so my bot can execute trades automatically. This includes handling order placement, cancellation, and partial fills.

2.4. Risk Management

Safety features like stop-loss orders and position sizing help me control potential drawdowns. I might use trailing stops or percentage-based position sizing.

2.5. Monitoring

I set up real-time tracking and logging to catch issues before they escalate, and to log trades for auditing performance over time.

3. Backtesting & Simulation

Before going live, I test my strategy on historical data to refine it and uncover potential flaws. I also run a simulated (paper-trading) environment to confirm the bot's logic and that my strategy doesn't suffer from overfit without risking real funds. My plan is to test more than one strategy in the live test with paper money in case my main strategy doesn't work at all.

4. Implementation & Operation

My language of choice will be Python, where the main reason is that it enables me to build the entire pipeline — from data collection and backtesting to execution and visualization — in a single language, without needing to switch between multiple technologies.

4.1. Time Period of Running Bot

The bot will run live from **June 16 to July 14, 2025** (weeks 25–28). During this four-week period, I will test the strategy under real market conditions using real capital.

The goal is not to maximize profit, but to validate the system's stability, execution logic, and risk management. I will:

- Monitor trade execution and system behavior
- Log trades, market data, and performance metrics
- Compare live results with backtest performance

This phase ensures that the strategy functions reliably before further optimization or longer-term deployment.

5. Feinkonzept Notes (Additional Topics)

5.1. What is Bitcoin

Here I'll briefly explain what Bitcoin is, so it's clear what asset my algorithm trades. Bitcoin is a decentralized digital currency with a fixed supply of 21 million coins. It runs on a public blockchain and allows peer-to-peer transactions without banks or intermediaries.

People buy Bitcoin as a hedge against inflation, for speculation, or long-term investment.

5.2. Backtesting Details

- Testing different kinds of strategies (Sample vs Out-of-sample)
- Sharpe ratio threshold (e.g., below 1.5 = more realistic)
- Real-time paper trading test
- Outline which indicators I'll be using

6. Efficient Market Hypothesis

6.1. EMH explained

Is it possible to abstract money from the market? The **EMH** states that markets fully price in all information, making it impossible to beat the market.

6.2. Why do I think I'm able to abstract money from the market?

- Markets have **temporary inefficiencies**; the goal is to exploit these before they disappear. This is especially true in crypto, where frequent liquidations (forced sells/buys) create short-term dislocations.
- Simple moving average crossover strategies are widely known; there may be more **complex strategies** with hidden edges.
- **Behavioral factors** play a major role. The EMH assumes rational players, yet many traders act emotionally. A disciplined system can exploit these behavioral inefficiencies.

6.3. Evaluation

Was I able to abstract money from the market? What was my edge allowing me to abstract money from the markets? If not where would I have to change things?