2016 Fall Competition
Case Packet

traders@mit.edu
traders.mit.edu

November 5, 2016

Contents

1 Introduction

Traders@MIT welcomes you to our 9th Annual Intercollegiate Trading Competition! We are excited to be holding the competition again this year and are confident that it will provide a rich and challenging learning experience.

Our competition consists of both open-outcry and electronic trading. Teams will be ranked overall based on their total weighted rankings from each of the four cases. You will also have several opportunities to speak with our attending sponsors:

Platinum: DRW Trading, Wolverine Trading, Town Square Trading and Optiver.

Gold: D.E. Shaw, Hudson River Trading, Bank of America Merrill Lynch, Jane Street Capital, Group One Trading, Goldman Sachs, IMC Financial Markets and Akuna Capital.

Silver: Flow Traders, Belvedere, SIG, Element, 3Red, Volant Trading, Allston Trading, KCG and TD Securities.

The remainder of this packet contains the competition schedule, descriptions of the cases that will be presented, logistical notes, and other preparatory information. We recommend that you read it thoroughly and build the necessary models before the day of the competition.

1.1 Schedule of Events

Note that all events in this schedule are **mandatory** for competitors to attend if they wish to receive travel reimbursements and be eligible for prize money; attendance will be taken. If this is not possible for you, then please email `traders@mit.edu`, with a detailed reason as to why you have a conflict with any of the events.

Friday, November 4

8:00PM	–	10:00PM	Networking at Champions Sports Bar, sponsored by Wolverine Trading

Saturday, November 5

DRW Trading will be taking the top five teams out to dinner immediately after the awards ceremony. Please plan return trips accordingly.

9:00AM	–	9:45AM	Check-In and Breakfast at MIT Sloan Tang Center (Room 345)
9:45AM	–	10:00AM	Opening Remarks by Optiver
10:00AM	–	11:00AM	Case 1: Quant Open Outcry
11:00AM	–	12:00PM	Case 2: Options
12:00PM	–	1:45PM	Sponsor and Competitor Lunch
2:00PM	–	3:00PM	Case 3: Price Discovery
3:00PM	–	4:00PM	Case 4: Foreign Exchange
4:00PM	–	5:00PM	Networking
5:00PM	–	5:30PM	Closing Remarks by Town Square
5:30PM	–	6:00PM	Awards Ceremony

1.2 Directions

Check-in and all cases will take place at the MIT Sloan Tang Center (Building E51) in room 345. Its address is

MIT Building E51
70 Memorial Drive
Cambridge, MA 02139

The Friday night networking session will be held at the Champions Sports Bar in Cambridge from 7PM-9PM on November 4th. It will be sponsored by Wolverine Trading and is a great chance to meet representatives from Wolverine as well as your fellow competitors. The address is

Champions Sports Bar
2 Cambridge Center
Cambridge, MA 02142

The luncheon will take place in the Sloan Executive Education Suite in Building E61, which is a short walk from the competition rooms. The afternoon networking session and the awards ceremony will take place in the MIT Media Lab, which is also a short walk from the Sloan Tang center. Traders@MIT executive members will guide competitors and sponsors on how to get to lunch and the networking session/awards ceremony from the competition.

1.3 Scoring and Awards

We will determine each team's final rank from weighting each team's rank in the four cases. The case weights are

Case	Weight
Quant Outcry	5%
FX	25%
Options	35%
Price Discovery	35%

We will award cash prizes to the top three overall teams and the top team in each case.

1.4 Requirements

All trading will be done on MangoCore, an in-browser trading platform. We require that all competitors use the browser Google Chrome.

In addition, all competitors must attend at least one remote practice session. We will be holding 4 online practice sessions: 10/27, 10/31, 11/1, and 11/2 at 8pm EST on each day. Each will be hosted at http://m.angocore.com/web. The practice session on 10/27 is optional - we will help you get familiarized with MangoCore. Competitors are required to attend a practice session on one of the latter three dates.

1.5 Piazza

We have created a Piazza forum to answer questions about cases and any other questions you may have. **Many of our updates and announcements will be placed here, so it is in your best interest to be on this forum**:

www.piazza.com/tradersmit/fall2016/tamit2016

The access code is **tamit2016**. Traders@MIT executive board members will do our best to reply to all questions within 24 hours. If you have questions specific to your team, feel free to contact us directly at traders@mit.edu.

1.6 Attire

Both the Friday networking session and the Saturday competition are business casual. Our Gold and Platinum sponsors will be in attendance.

2 Quantitative Outcry

2.1 Overview

During the Quant Open Outcry case, teams of two will trade TAMIT Index futures. Each team consists of a trader, who will trade on the floor of the trading room, and an analyst, who will be located in the gallery seating section of the room.

Teams will construct pricing models based on historical index levels and the indicators explained below. Analysts can view the current indicators on the MangoCore web interface or get this data programmatically, as described in MangoCore API. These values should be used to construct pricing models. Teams should construct their pricing model **before** the day of the competition. Analysts will use their models and convey trading instructions to traders.

The case will last for 45 minutes, divided into three 15-minute rounds. A reading for each of the indicators will be released every 30 seconds to the analysts. Note that this data will not be available to pit traders. Indicator releases can be considered accurate and free of errors. Although pit traders cannot see indicator releases, both traders and analysts will be able to see the underlyings market in real time on the projector screen.

Accurately forecasting the index level and rapidly relaying trading instructions will result in higher profits. The TAMIT Index will begin trading at a value of 1000, and the final value for each heat will depend on the economic indicators. Note that the index returns are noisy – you should not expect to predict them perfectly from the indicator values.

Correlations between the TAMIT Index and the indicators may or may not stay fixed throughout the entire event, so it is suggested that analysts keep track of the accuracy of the model and consider re-running analyses when necessary.

2.2 Constructing a Useful Pricing Model

There are various indicators that affect the TAMIT Index, and most of them are types of economic data.

We provide 50,000 data points of historical data for the level of the TAMIT Index and the economic indicators. You may use this data to estimate a model to predict the TAMIT Index. This data is saved as `outcry_data.csv` in the shared Dropbox folder.c

Each team is advised to develop a model (using, for example, Excel or Python) to determine the relationship between economic indicators and the TAMIT Index. New indicator values should be entered into the Excel model when they are released, in order to predict the next-period index value. The provided historical data should be considered accurate for modeling the future position of the TAMIT Index.

The following indicators will be available:

1. Gross Domestic Product (GDP)
2. Money Supply (MS)
3. Consumer Price Index (CPI)
4. Producer Price Index (PPI)
5. Personal Income (PI)
6. Unemployment Rate (U)
7. Housing Starts (HS)
8. Retail Sales (RS)
9. Manufacturing and Trade Inventories and Sales (MTIS)

Each indicator has a different effect on the TAMIT index; it is your job to model these effects using the provided appendix of historical data. More specifically, it is your goal to build a model that can, given the indicator values up to time t and the index values up to time $t-1$, predict the value of TAMIT for time t. Here are a few suggestions for building a good model:

1. Use the mathematical software of your choice to fit the historical data to a polynomial regression, with the independent and dependent variables chosen in a manner that fits the context of this case. Note that the regression may be between transformations on the provided values, like the derivative $dx \approx x_t - x_{t-1}$ or proportional change dx/x.
2. To model proportional change in a regressor, note that $dy/y = d \log y$. Then, the proportional change between times $t-1$ and t for an indicator can be approximated by $\log(\text{INDICATOR})_t - \log(\text{INDICATOR})_{t-1}$. The proportional change in the TAMIT index can be similarly estimated.
3. At time t, you will have all the indicator values up to time t and all the index values up to time $t-1$. Incorporate dependence on as few or as many previous values as you see necessary.
4. Some indicators may have no effect on the TAMIT index.

Note that you and your analyst will be using this model on the day of the competition, so make sure it is easy for your analyst to use in predicting index values.

2.3 Starting Values

The starting values of the indicators on the day of the competition are given below in Table 3.2. Use these values in your trained model to predict the value of the TAMIT index.

Indicator	Starting Value	Units
GDP	1600	USD billion
MS	997	USD billion
CPI	-0.1	percent change
PPI	-0.4	percent change
PI	275	USD billion
U	6	percent
HS	1.09	million
RS	450	USD billion
MTIS	1279	USD billion

2.4 Rules

Each team must switch off as trader/analyst every period. During the trading period, traders must make markets consisting of a bid and an ask price. For example, you can say: "500 at 1200", meaning that you are willing to buy at 500 and sell at 1200. When making markets, traders can also drop the dollar values and just state the cents as an abbreviation if it is clear which dollar prefix you are referring to. For example, you can say: "97 at 103" as a shorthand for "10.97 at 11.03". Use good judgement when using shorthand.

In order to complete a transaction, the two parties must fill out an order form completely. Both sides of the trade must present this form to a Traders@MIT executive board member. We reserve the right to reject all poorly written trade forms.

2.5 Scoring

Each team's outstanding positions at the end of the trading session will be cash settled. The payoff at the end of the round for a futures contract can be calculated as:

$$\text{Payoff} = \$10 \cdot (S_T - F) \text{ for a long position, and}$$
$$\text{Payoff} = \$10 \cdot (F - S_T) \text{ for a short position,}$$

where S_T is the spot level of the TAMIT index at the end of one year, and F is the futures level specified by the TAMIT futures contract. The futures level will depend on the supply and demand from other traders.

The quality of analysts' models will not factor into your final score; however, accurate estimates of the index will greatly assist you in achieving higher profits. Each round, a team is scored based on total PNL. The overall team placing for the case will be the average placing across rounds.

3 Options

3.1 Overview

In this case, you will use news releases to trade European options on T@MIT, an equity index. Your ability to earn profits will depend on your ability to:

1. predict changes in implied volatility across different strike prices based on news releases
2. identify, enter, and exit appropriate positions
3. hedge positions to reduce risk

This case is relatively complex. We strongly recommend investing time in understanding the relevant concepts and building a useful model before the competition.

For the rest of this case description, we will simply refer to European options as options.

3.2 Options Basics

Options come in two flavors: A *call* option confers the right to buy a given product (the underlying) at a fixed price (the strike price) on – but not before – a given date (the expiration date). Likewise, a *put* option gives the right to sell the underlying at the strike price on the expiration date.

For example, imagine that an investor is holding an Apple call option at a strike of \$100 and expiry of November 15th. (S)he would exercise the right to buy if the price of Apple was greater than \$100 on November 15th.

The price or value of an option depends on the volatility of the underlying. *Volatility* is the standard deviation of the logarithmic return distribution of the underlying. Intuitively, it is a way to quantify how much the price will fluctuate. There are two relevant measures of volatility:

- Realized volatility is the volatility of the underlying over some past time period and is generally estimated as the sample standard deviation of the annualized log returns during the period.
- Implied volatility is the volatility of the underlying that when used in a pricing model returns the current market price of the option. We will explore this concept more later.

Options can be either in-the-money, out-of-the-money, or at-the-money. In-the-money refers to an option that is profitable if it were to be exercised. Out-of-the-money is defined analogously. At-the-money refers to an option that is whose strike price is at the spot price; i.e. the option would break even if it was exercised today.

3.3 Options Pricing Models

Given the current price of the underlying (the *spot* price) S, strike price K, risk-free interest rate r, volatility σ, current time t, and expiration time T, the Black-Scholes formula for the value C of a call option on a non-dividend-paying asset is

$$C = N(d_1)S - N(d_2)Ke^{-r(T-t)}$$

where

$$d_1 = \frac{ln(\frac{S}{K}) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}$$

$$d_2 = d1 - \sigma\sqrt{T-t}$$

and $N(x)$ is the standard normal cumulative distribution function. A derivation of this model can be found here: http://www.math.cuhk.edu.hk/~rchan/teaching/math4210/chap08.pdf. Briefly, the Black-Scholes formula and related extensions assume the spot price follows a geometric Brownian motion, which is equivalent to claiming that incremental logarithmic returns are normally-distributed and have a constant standard deviation. In reality, returns are not typically normally distributed. They tend to have a higher probability of extreme events than would be the case under a normal distribution ("fat tails"), particularly for extreme negative events. The Black-Scholes formula does give a fast and relatively accurate approximation, and is widely used as a basic model. The value P for a corresponding put option can be obtained from put-call parity, which for European options is as follows:

$$C + Ke^{-r(T-t)} = P + S$$

The intuition behind this formula is as follows: Ignoring for the moment the interest rate (which will be $r = 0$ during the competition), the asset package on the left side of this equation (i.e. one call option plus the dollar value of the strike) will pay out a guaranteed K, plus an additional bonus value if the option is in-the-money equal to the difference between the price of the underlying and the strike price. Therefore the left side will pay out the larger of K and the price of the underlying at expiry. Meanwhile, the right side (comprised of a put P and a future S) will pay out the value of the underlying, plus additionally the difference between the strike price and the underlying if the put option is in-the-money. Take a moment to understand why these are equal; the interest rate factor simply accounts for the time value of money.

3.4 Volatility Curves, Smile and Skew

Under the assumption that the options pricing follows Black-Scholes, we can invert the above formula for a given strike price and expiry time to deduce the implied volatility. Perhaps surprisingly, options with different strike prices often have different implied volatilities. Typically options that are more out-of-the-money or in-the-money will have higher implied volatilities, because the actual distribution of the returns of the underlying is more fat-tailed than the normal distribution. This effect is known as the *volatility smile*. Moreover, options with strike prices much lower than the current spot price often exhibit higher implied volatility than options with strike prices equally higher than the current spot price. This effect is known as the *volatility skew*. The best way to visualize this is by plotting the implied volatility against the strike price; this is known as the volatility curve. Figure 1 shows the effect of the October 1987 stock market crash on the volatility curve.

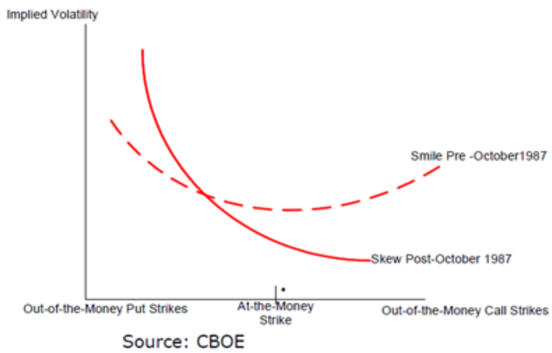
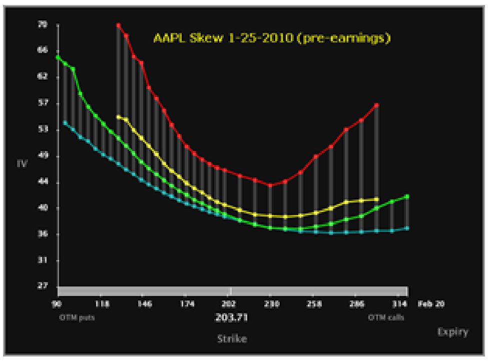
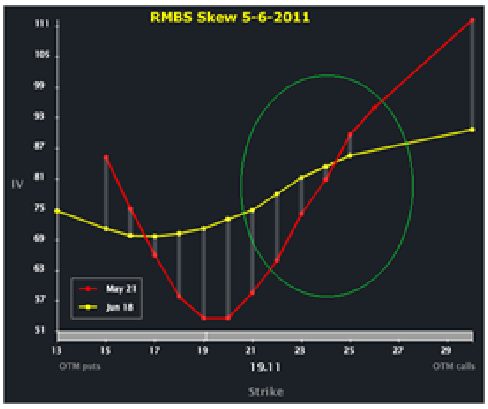


Figure 1: The S&P 500 Implied Volatility Curve Pre- and Post-1987

As seen above, an increase in the perceived likelihood of extreme negative events can increase volatility skew. Similarly, a belief of increased likelihood in extreme events in either direction compared to more typical events unchanged can make the volatility smile more pronounced. Note however, that these effects are *not* equivalent to that of belief that price movements will on average be larger than they have been in the recent past; this instead causes a parallel shift upward in the implied volatilities of at all strike prices. These effects can occur simultaneously.

Below, we show two examples of changing volatility curves. The left is from just prior to an earnings release for a stock. Note the pronounced curvature and asymmetry in the front-month options (red). Also note that the overall level of volatility for the front-month options is higher than for longer-dated options, indicating that most of the variability in price is expected to occur in the near-term:





In contrast, the chart on the right comes from a company that engages in numerous patent lawsuits. A particular one is expected to reach a decision soon, and the decision will be either very good or very bad for the future of the company. The front-month options again exhibit a substantial volatility smile, but it is more symmetrical this time. Also, subsequent volatility remains high in this example given the risks of other ongoing litigation and R&D. However, these later risks are expected to be less binary than the impending decision: market participants predict the later return distribution will have thinner tails, so the implied volatility surface is flatter.

3.5 Greeks and Hedging

Option prices are a function of multiple variables, including the price of the underlying, the volatility of the underlying and time. We can differentiate with respect to these variables to get various quantities, commonly called the “Greeks." The Greeks describe the effects of changes in various parameters on the price of the option. We can think of the Greeks as representing exposure to market risk along different axes. Hedging is the process of minimizing such risk across each Greek.

The simplest and most important Greeks are:

- Delta: the rate of change of an option’s value relative to a change in the underlying. It is always positive for calls and always negative for puts. Traders can think of delta as the impact of the movement of the underlying on the value of their portfolio; therefore, it is also equivalent to how many shares of the underlying to sell in order to hedge their option against the movement of the underlying. Having a portfolio with a small (in absolute value) delta corresponds to not having an opinion as to whether the underlying will move up or down. The value of delta for a call option is always between 0 and 1, and the values of delta for put and call options at the same strike price can be related via put-call parity (since the derivative of the underlying with respect to itself is clearly 1). Intuitively, for call options we should expect that delta should approach 1 for very in-the-money options and 0 for very out-of-the-money options; this is indeed the case.

- Gamma: the rate of change of delta relative to a change in the underlying. Since delta increases from 0 to 1 as the underlying passes the strike price, gamma is always positive for call options; since the delta for put options is simply 1 less than that for call options via put-call parity, gamma is the same for put and call options. Gamma determines how quickly you must adjust a delta hedge when price changes.

- Vega: the rate of change of an option’s value relative to the (implied) volatility of the underlying. Vega is positive for every option. Intuitively, this is because extremely out-of-the-money options result in a fixed loss, whereas extremely in-the-money options can have large payouts proportional to exactly how in-the-money they are; hence, as volatility and hence the likelihood of both extremely good and extremely bad events increases, the options price should only reflect the added value of the extremely good outcomes and will therefore increase. Vega tends to be higher for options with later expiry dates and for options with a strike price near the current spot. A trader holding a portfolio with a small (in absolute value) vega has no opinion about whether the volatility of the underlying will go up or down across all strike prices; however, (s)he may still have an opinion about the shape of the volatility curve (which will be discussed in the next section). The vega for the underlying will be zero.

Expressions for the Greeks from differentiating Black-Scholes are below, assuming an interest rate of 0. Note these can become unstable for options very close to expiry: where $N(x)$ is the

Table 3.1: Formulas for Greeks

Greek	**Calls**	**Puts**
Delta	$N(d_1)$	$N(d_1) - 1$
Gamma	$\frac{N'(d_1)}{S\sigma\sqrt{T-t}}$	$\frac{N'(d_1)}{S\sigma\sqrt{T-t}}\ N(d_1)$
Vega	$SN'(d_1)\sqrt{T-t}$	$SN'(d_1)\sqrt{T-t}$

standard normal cumulative distribution function and $N'(x)$ is the standard normal probability density function. Successful competitors will be able to compute the Greeks for each product using a model receiving live updates from MangoCore, making use of the implied volatilities computed by inverting Black-Scholes.

3.6 Volatility Arbitrage

Volatility arbitrage aims to take advantage of inconsistent future and current volatilities. For a certain underlying P, profits can come from differences between the anticipated future realized volatility of P and current implied volatilities of options for P. Profits can also come from differences between current and future implied volatilities, and from the differences in implied volatilities across different strike prices.

For example, suppose a trader believes that the future realized volatility for a given product will be higher than the current implied volatility of a call option on that product across all strike prices. Then the trader believes the option is cheaper than it should be, so (s)he buys it, hedges delta by shorting some amount of the underlying, and adjusts the hedge as the price of the underlying changes. Note that this trade has a positive vega, because the trader believes that the implied volatility will increase.

One of the most common types of volatility arbitrage is called volatility surface relative value trading. It involves attempting to earn profits by predicting changes in the shape of the volatility smile, independent of beliefs about vega.

Suppose you expect an increase in the curvature of the volatility smile due to news releases. Then you could buy options far out on the curve, perhaps hedge some vega exposure by shorting options with strike prices closer to the current spot price (at-the-money options) if you have no view on the future overall level of volatility, and then hedge your delta exposure by holding a position in the underlying equal and opposite to the remaining delta of your options portfolio. This leaves you exposed only to changes in the curvature of the smile. A completely analogous strategy could be employed if you believe that the skew of the volatility smile will increase. Successful competitors in this case will be able to predict changes in the volatility curve based on news items, and trade accordingly.

3.7 Case Information

The tradable instruments are options and futures on the T@MIT index, the latter solely for hedging. Both are cash-settled. Note the options underlying is the index, not the futures contract.

There are ten options in each round of the case: five puts and five calls. Options expire at the end of each round and their tickers are subsequently re-used. All tickers are as follows:

Table 3.2: Tradable Options

Strike Price	Put Option Ticker	Call Option Ticker
$90	T90P	T90C
$95	T95P	T95C
$100	T100P	T100C
$105	T105P	T105C
$110	T110P	T110C

The ticker for futures on the index is “TMXFUT”. The ticker for the T@MIT index itself is "TMX". The index is viewable but not tradable.

To successfully trade this case, you should make a pricing model to show the current volatility curve, your current portfolio Greeks, and any refinements or derived metrics that help you trade faster and better. This will allow you to quickly pursue arbitrage opportunities and to hedge your positions. You should also pay attention to news releases, which could help predict future market movements. Competitors should read MangoCore API to understand how to get live MangoCore updates programatically and then build models off of them.

3.8 Case Specifications

There will be four 7.5-minute periods. Each period represents one month of trading. One team member will trade while the other monitors a pricing model and advises the trader. Team members will switch roles after each period. Trading activity by both partners during any given round is grounds for disqualification. Unlike the SEC, we have granular audit logs – please don’t make us use them.

All tradable instruments expire at the end of each round. Each round begins with the T@MIT index at 100. The continuously compounded risk-free interest rate is **0%**. No dividends will be paid out during case periods.

The contract multiplier for **options** on the T@MIT index is 1. Options have a trading fee of $0.005 per contract per transaction. There is a gross trading limit of 200,000 options contracts, and a net limit of 100,000 options contracts.

The contract multiplier for **futures** on the T@MIT index is 1. Futures have a trading fee of $0.005 per contract per transaction. There is a gross trading limit of 50,000 futures contracts, and a net limit of 50,000 futures contracts. All future contracts are cash-settled upon expiry.

There will also be delta and vega **portfolio limits** of ±1000 and ±1000 respectively. One share of the underlying will have a delta of 100. Upon exceeding either limit, competitors will be fined proportionally to the square of the difference between their Greek and the limit, at a rate of $0.001 per squared excess per second. For instance, a competitor holding a portfolio with a vega of -1250 and a delta of 1100 will be fined at a rate of $72.50 per second. MangoCore will display an alert when your portfolio nears the threshold.

All outstanding positions at the end of each period will be automatically closed out at their fair prices, and options will be automatically exercised if they are in-the-money. In each round, your rank will be calculated based on your P&L (including any and all fees) relative to other competitors; winners will be determined by the lowest average rank across the four rounds.

4 Price Discovery

4.1 Overview

EXE, CPP, HS, PYPY, RB have just issued their earnings reports for the most recent quarter, and as highly traded stocks, many analysts are continuously predicting the target price of these 5 stocks. However, these analysts are not always accurate, and their predictions should be considered holistically. Additionally, there is an index of these stocks, each with a weight of 1, trading under the ticker IDX.

In this case, traders will analyze predictions to come up with a maximum likelihood estimate of the true price of the 6 products (5 stocks + 1 index).

There are four 10-minute periods. One team member will trade while the other performs analysis. Team members will switch roles after each period.

When a number of traders hold potentially different information or views on a particular security, the markets provide a mechanism for price discovery. In this case, traders can take price estimates they receive from research analysts to deduce the true value of each stock. Competitors should consult MangoCore API for information on how to programatically receive pricing information from MangoCore.

4.2 Case Meta-information

In the first period, all securities are tradeable, potentially allowing for arbitrage opportunities between the basis securities and the index. In the subsequent cases, all but one security is tradeable.

For each period, the true prices of each security are independently and randomly drawn from a uniform distribution. The true prices are known to be within the following bounds:

	minimum	maximum
TRA	20	40
DER	10	20
SA	60	90
TM	30	70
IT	50	80

You should not rely on pricing data from past periods to help you in the upcoming periods.

Traders begin each period with an endowment of $1,000,000 in cash. Each period represents one quarter of the year, and the risk free rate is 0% so there is no interest paid on cash balances.

Fees are \$0.005 per contract, and rebates are \$0.0025. The fine for self-trading is \$0.01. The position limit is ±1000 for each basis security. Orders that violate this limit will be rejected.

4.3 Predictions

During each period, each trader will receive many stock predictions from analysts. Each prediction is a sample from a multivariate normal distribution with mean equal to the true prices of the securities.

The standard deviation for a news release from source s at time t seconds, for any ticker is:

$$\frac{q[s](600-t)}{60}$$

Intuitively, this means that certain news sources are more reliable than others, and that predictions are more accurate as time goes on.

Traders will receive estimates from Buzzfeed, The Associated Press, Seeking Alpha, and @ETFGodfather (on Twitter). Not in any particular order, the quality vector q is [.4, .8, 2, 4]. It is the trader's and analyst's responsibility to discover how reputable each source is. The quality of each source is constant throughout all periods. For example, if Buzzfeed has a quality of .4 in round 1, it will have a quality of .4 in all other rounds. Competitors will receive a news release once every few ticks, and each news release will only contain the information from one of the four predictors.

An example of the body of a news report is:

> TRA estimated to be worth 39.70; DER estimated to be worth 73.68; SA estimated to be worth 49.03; TM estimated to be worth 12.62; IT estimated to be worth 45.83

The correlation coefficients for the multivariate normal are given as follows:

	TRA	DER	SA	TM	IT
TRA	1	0.46	-0.58	-.06	-0.39
DER	0.46	1	-0.22	0	-0.38
SA	-0.58	-0.22	1	-0.11	0.52
TM	-0.06	0	-0.11	1	-0.19
IT	-0.39	-0.38	0.52	-0.19	1

4.4 Other Market Participants

In this case, the primary other market participants are market makers. These market makers are not particularly intelligent and simply make a market around the last transaction price.

The secondary other market participant is a trader that randomly posts market orders for randomly selected products.

Of course, participants will also be trading directly against each other.

4.5 Position Close-Out

Any non-zero position in a security will be closed out at the end of the trading period with the actual price computed from the earnings report of the next quarter. *Note that this price may be significantly different from the last transaction price of the security.*

The risk free interest rate is 0% and there are no dividends paid for any of the underlying stocks.

4.6 Scoring

A team's score for this case is the average relative rank among all teams for each period (determined by PNL). This score is then normalized as the relative rank amongst all teams.

5 Algorithmic Trading: Foreign Exchange

5.1 Overview

The foreign exchange (FX) market is the global platform for trading currencies and currency products. FX is the largest and most liquid market in the world: about $5.1 trillion is traded per day. The real-world FX market is heavily dependent on macroeconomic news and events. However, this case will not contain news and will instead focus on arbitrage opportunities. You will write an algorithm to make money from price discrepancies between currency pairs. You will also be able to click trade on top of your algorithm, but we emphasize that you will greatly increase your PNL by creating a well-written algorithm. **Information about the TradersBot API will be provided MangoCore API, which will allow you to pull in pricing information and also submit trades programmatically**.

5.1.1 Understanding Foreign Exchange Bid/Ask Quote

FX quotes are given in terms of currency pairs. The first currency listed is the base currency, and the second currency listed is known as the quote currency. When you are going "long" or buying a currency pair, you are buying the base currency and selling the quote currency. The quote currency is quoted in terms of its value relative to the other currency in the pair. For example, if the USD/JPY exchange rate is given as

$$\text{USD/JPY} = 106.78,$$

this means that US$1 = 106.78 Japanese yen. In the next example, USD is the quote currency and EUR is the base currency. If the exchange rate is the following:

$$\text{EUR/USD} = 1.2508,$$

then 1 Euro = US$1.2508.

In the FX market, these currency pairs can be bought and sold at the given bid/ask price. The bid is the price the counterparty is willing to buy the pair from you, and the ask is the price they are willing to sell the pair to you. Remember, "buying the pair" means buying the base currency and selling the quote currency, while "selling the pair" means selling the base currency and buying the quote currency.

5.2 Case Information

In this case, you will be trading pairs of the following 5 underlying currencies: USD (US dollar), EUR (Euro), JPY (Japanese yen), CHF (Swiss franc), and CAD (Canadian dollar).

You will be able to trade 8 currency pairs, listed below:

USD/CAD	EUR/USD	USD/CHF	USD/JPY
EUR/CAD	EUR/JPY	EUR/CHF	CHF/JPY

Note that this is not all $\binom{5}{2} = 10$ possible currency pairs.

Case details:

Starting Endowment Per Round	US$100,000
Rounds of Trading	3
Length of Round	15 min/round
Fees	$0.005/contract
Fines	Trading with yourself: $0.008/contract
Constraints	Position limits will be enforced; see below. Orders that would cause you to go outside your position limit will be rejected. You may only place a maximum of 100 orders/sec, otherwise your orders will get rejected.

Position limits for the underlying currencies:

Currency	Position Limits
USD	±1,000,000
EUR	±900,000
JPY	±102,000,000
CHF	±980,000
CAD	±1,300,000

Note: Position limits for underlying currencies do not imply position limits on each currency pair.

5.3 Trading Tips and Strategies

In this section, we describe the concept of triangular arbitrage, which can be used in your algorithm. As mentioned before, you may also click trade; however, remember that solely click trading will not be as effective as a well-written algorithm.

We will provide servers to backtest your code on. If your code is consistently profitable when traded on historical data, there is a better chance it will be profitable during the competition.

We now explain the concept of triangular arbitrage. The motivation behind triangular arbitrage is to take advantage of price discrepancies between three currencies. For example, theoretically, the price of the pair EUR/JPY should equal the price of the pair EUR/USD times the price of the pair USD/JPY. If not, then an arbitrage opportunity exists. As an example, suppose that

$$\text{EUR/JPY} = 134$$
$$\text{EUR/USD} = 1.12$$
$$\text{USD/JPY} = 119$$

Note that $1.12 \cdot 119 = 133.28 < 134$. We can make money by buying the two currency pairs EUR/USD and USD/JPY, and selling EUR/JPY. Explicitly, we can convert 13328 yen to US\$112 by buying 112 contracts of USD/JPY. Next, we can can convert US\$112 to 100 Euro by buying 100 contracts of EUR/USD. Finally, we can convert 100 Euro to 13400 yen by selling 100 contracts of EUR/JPY. Overall, we made a profit of $13400 - 13328 = 72$ yen. Our profit from this trade is US\$72/119 $\approx$ US\$0.61.

5.4 Position Close-Out and Scoring

Any non-zero position in a currency will be closed out at the end of the trading period with the closing price. Your PNL during the case will be displayed in US dollars—our grading script will make the necessary currency conversions in real time.

Within each of the 3 rounds, you will be ranked by your PNL. Your PNL will reset to 0 at the beginning of each of the rounds. To determine final rankings, we will consider your average rank across all three rounds.

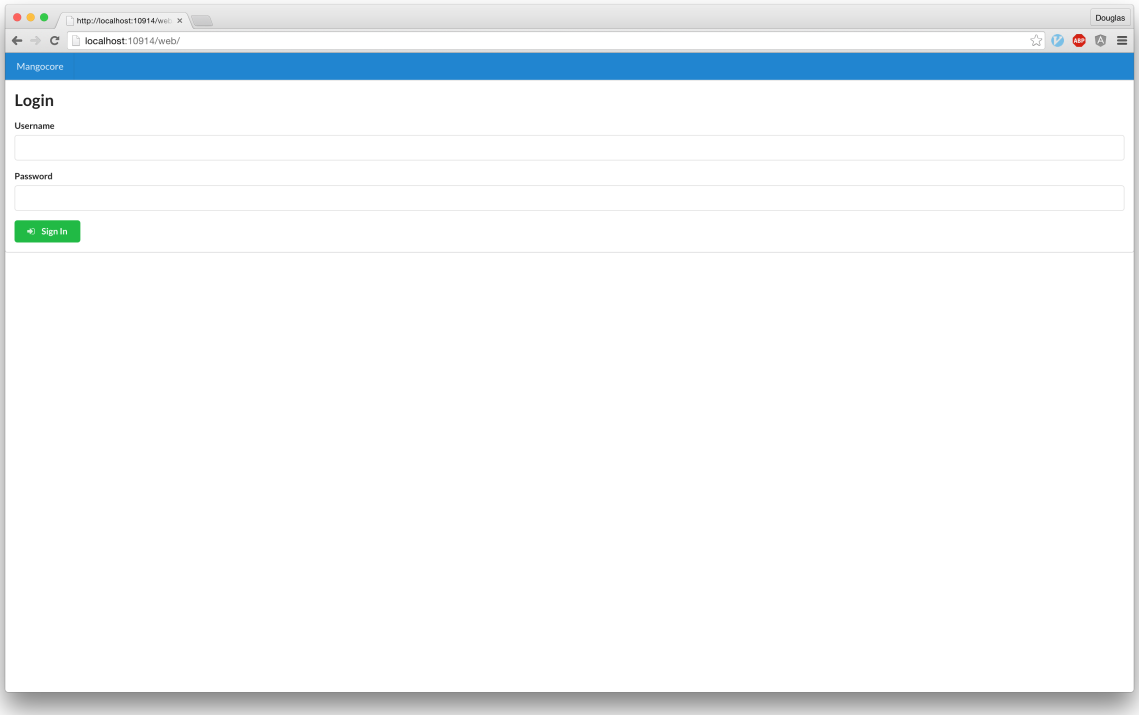
6 MangoCore Web Interface User Guide

6.1 Overview

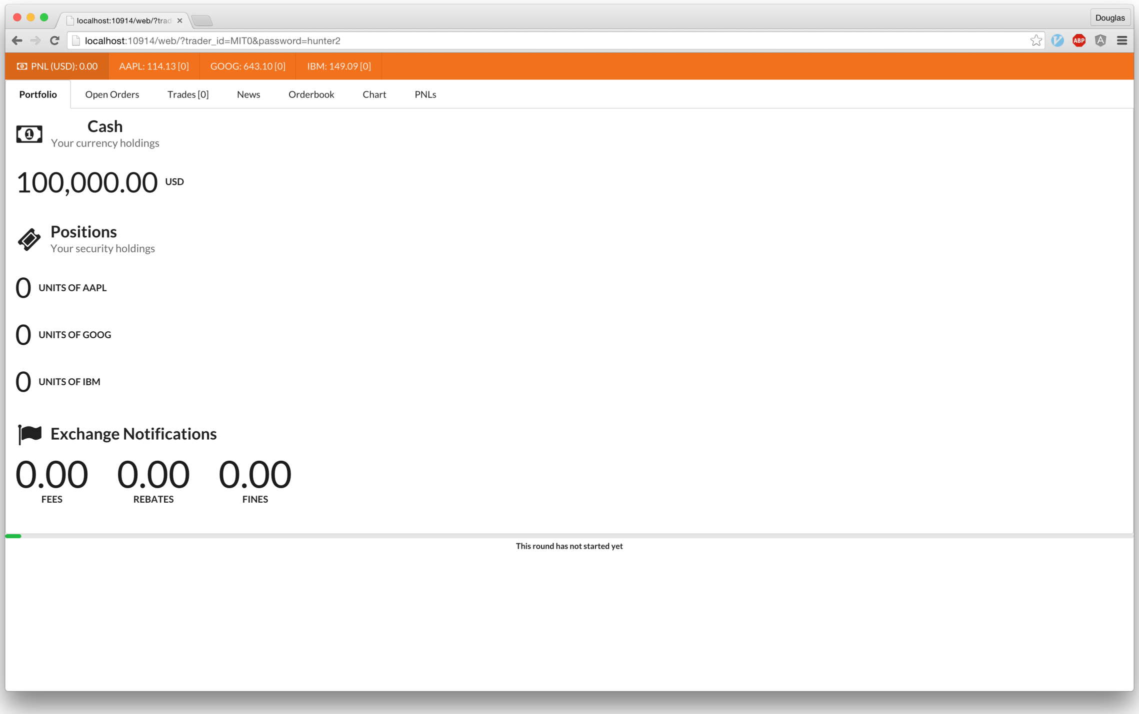
MangoCore is our general-purpose broker and exchange system for educational purposes. It is not connected to live exchanges and does not process real money. For these cases, you can interact with the simulator either through our web interface or through its JSON API. To access our web interface, go to http://m.angocore.com/web with a modern web browser.

6.2 Navigation

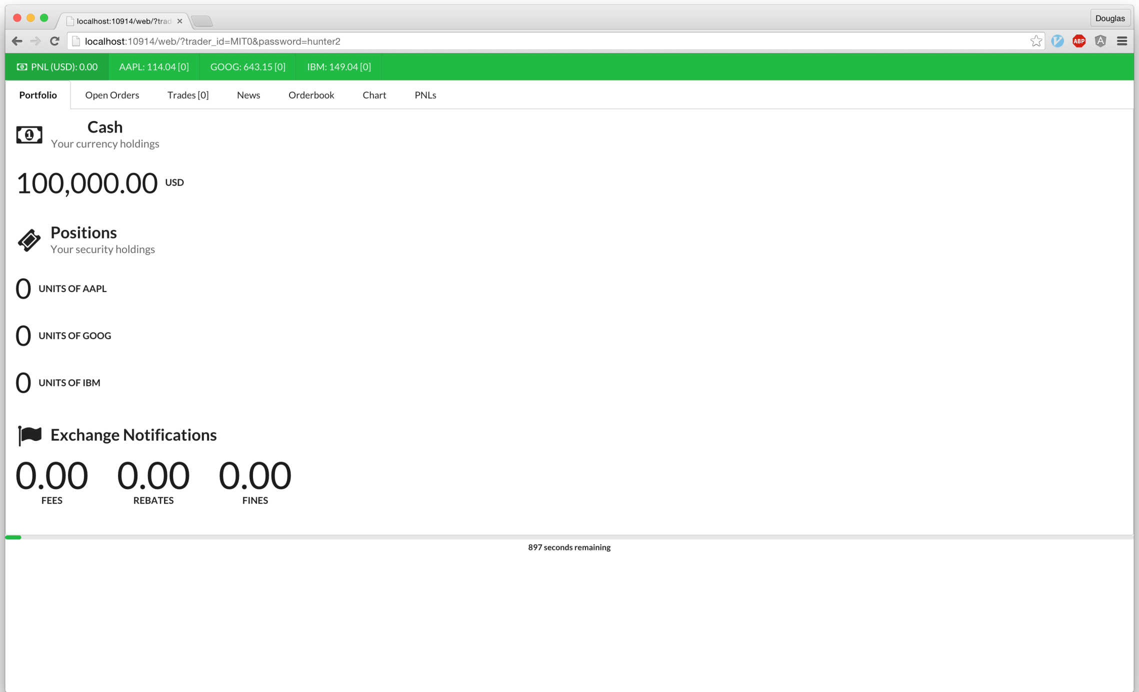
You will initially be presented with a login screen:



After logging in, but before the case starts, the summary bar at the top will be orange.



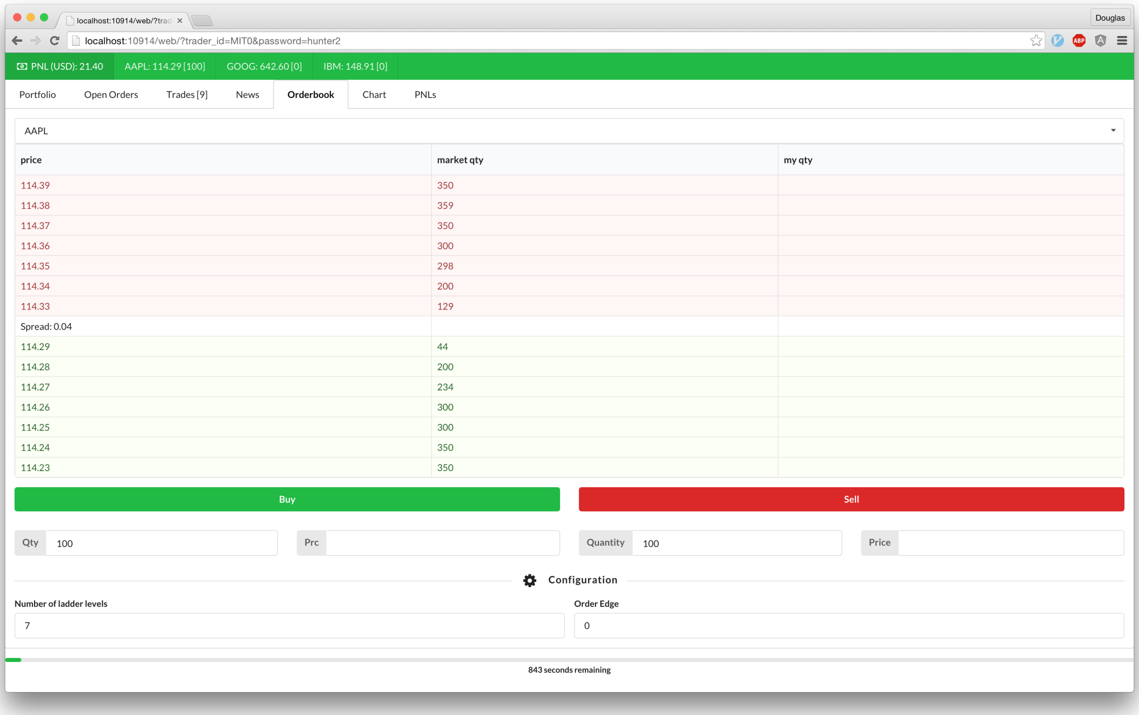
Once the case starts, the summary bar will be green. The summary bar at the top of the screen shows your PNL (profit and loss), as well as the market price and your current position for each security. You can view your portfolio, your open orders, released news, the orderbook, and charts of the product price paths in each tab. To view multiple tabs at once, you can open additional browser windows with Mangocore.



6.2.1 Orderbook

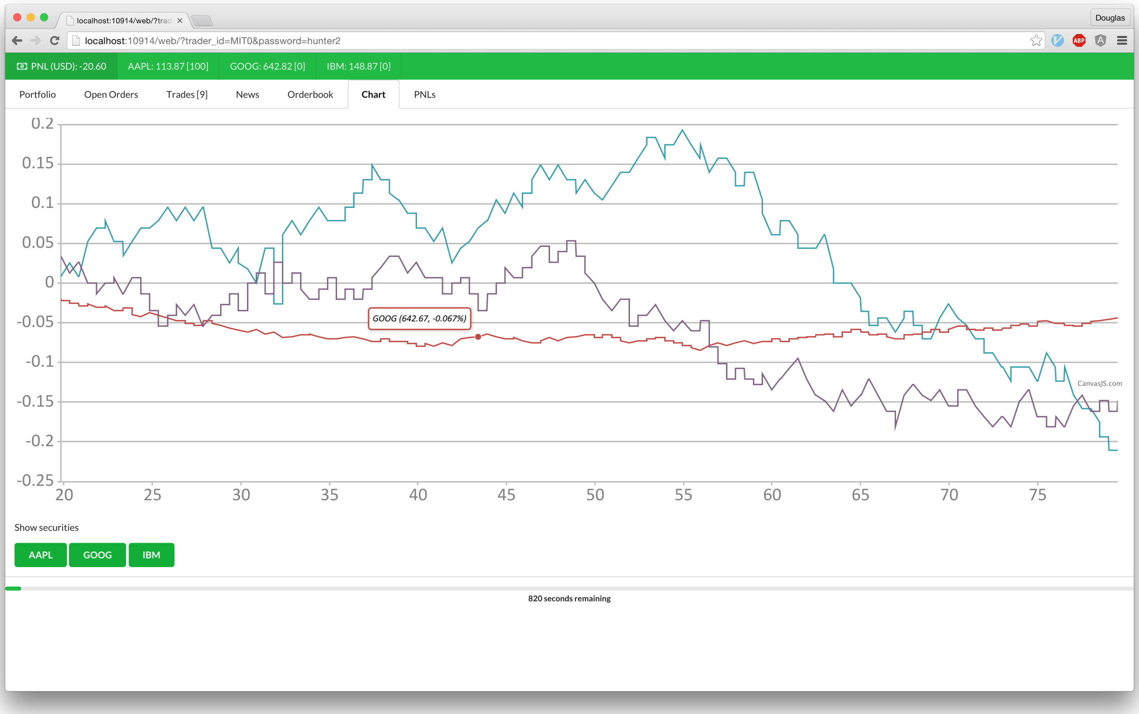
The orderbook shows a ladder with only the price levels corresponding to orders in the market. In the center of the ladder the current spread between the highest bid and lowest ask is shown. Like orderbook 1, there are also "Buy" and "Sell" buttons, which behave the same way. Alternatively, the upper half of the ladder highlighted in red can be clicked to submit limit orders to sell at the corresponding price. The lower half of the ladder highlighted in green can be clicked to submit limit orders to buy.

If there is an error such as exceeding position limits, it will be shown at the bottom of the screen in a yellow bar.



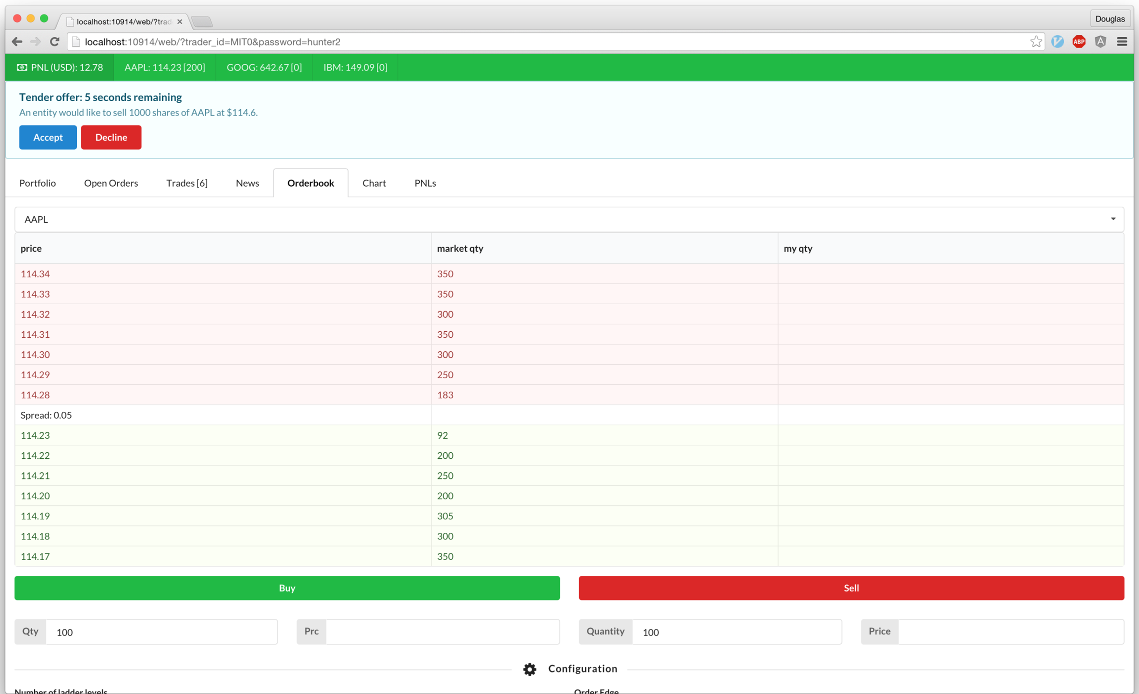
6.2.2 Chart

The chart shows the relative change in the price of securities over time. The horizontal axis represents time. The vertical axis represents the percentage deviation of the security's price from its starting price. The buttons on the bottom control which securities are shown on the chart. Mousing over the lines in the chart shows the security's name and price.



6.2.3 Tender Offers

Some cases have tender offers. These will be displayed at the top of the screen in a blue bar. You will have a limited amount of time to accept or decline these tender offers.



7 MangoCore API

7.1 Overview

MangoCore interfaces with external clients with a JSON API over a websocket connection. Users can use this API to programmatically query our simulator servers which may be helpful for cases that are suited for higher frequency trading. This guide details how to set up a local server for testing and provides information about the TradersBot Python package (available on pip), which allows easy interfacing with Mangocore's JSON messages.

For all cases, competitors should expect to pull price paths in addition to other information relevant to each case from MangoCore and analyze this data with real-time models. Ideally information from these models should inform competitors' trades. The only case for which traders will be allowed to *submit* orders programmatically through MangoCore is the algorithmic trading case (FX), where competitors can both pull price paths and submit orders for various securities in real-time.

7.2 TradersBot Python Package

As MangoCore interfaces with external clients with a JSON API over a websocket connection, theoretically it is possible to interact with the client through any programming language of choice, provided the user knows how to set up a websocket connection. However, we *highly* recommend that competitors use the TradersBot Python package, which provides a Python wrapper to the MangoCore API that allows for easy websocket setup to interface with the Mangocore server. TradersBot's installation, usage and general documentation are extensively detailed at `http://mangocore-client.readthedocs.io`, which provides instructive examples along with a thorough description and purpose of each feature. This URL is subsequently referred to as the *Documentation*.

In this section, we will broadly describe the functions provided by TradersBot as well as the types of messages that the MangoCore server sends to TradersBot. We also provide details on how to run a local instance of MangoCore to allow for local testing.

7.3 Server setup

MangoCore is run on a server; during competition day, we will be hosting the MangoCore server and provide the necessary IP/URL for competitors to connect to. However, competitors might want to run MangoCore on a local server in order to test their strategies. To run the server locally, grab the executable for your operating system from the competitor Dropbox and follow the instructions in the instructional PDF in the Dropbox folder. The server will open up a port on localhost:10914 and you can connect to it by opening up a websocket connection to `ws://localhost:10914/#trader0`, although if you are using TradersBot, the websocket connection will be setup for you. *Note: you cannot use your given competition traderid locally because the provided binaries default to a small set of recognized traderids that will not include your assigned traderid.*

To run your algorithm on our server for competition day and practice sessions, switch the host from `localhost:10914` to `m.angocore.com` and `trader0` to your given trader ID.

7.4 TradersBot Setup

There are several parameters when connecting with a MangoCore server: the MangoCore server IP/URL (which will be provided to you on competition day for the competition server and can be run locally with information located in the section Server Setup), a team username, a password, and an optional token.

TradersBot will automatically establish a connection with the server when you instantiate it with necessary parameters. A thorough description of how to use TradersBot, including instantiation, is in the documentation.

7.5 Messages

MangoCore has several types of messages that it will deliver to TradersBot. We describe the types of messages below; the exact information they contain and the message format is described in more technical detail in the documentation. In general, you can expect that each message is delivered as some sort of Python dictionary with keys corresponding to relevant information for that type of message. For example, a trade message might contain a dictionary with a key “ticker" that denotes which stock ticker the trade is referring to.

For each type of message, you can implement a function in your Python script that takes in as input the `message`, and parses the information in the message, possibly analyzing it and/or updating any internal variables you have with the parsed information. TradersBot has an internal function on receiving each type of message that we can set to a user defined function. An example TradersBot use case is in the documentation, which implements functions for each type of message.

We describe the content and purpose of each message below. Again, for specific details about the message format and the dictionary that comes with each message, refer to the documentation. This is meant to outline and guide competitors on the purpose of each type of message.

7.6 Passive Information

7.6.1 onMarketUpdate

Sometimes, you will receive market updates for securities. This generally happens every 500ms as opposed to following every API call that may modify the state of the order books. However, you will receive every trade notification so you may be able to maintain an orderbook estimate that can be more accurate.

Additionally, you can ping the market with micro-orders to discover the top of the book. You will want to do this sparingly because of the trading limits (see the last section).

7.6.2 onTraderUpdate

As with market updates, you will generally receive trader updates every 500ms. However, you will be able to calculate your precise trader state at any moment in time based on other information you receive.

7.6.3 onTrade

You will receive a notification following every trade (not just the ones concerning your orders) immediately (after network/processing latencies of course). You can use this information to keep an accurate internal order book and also have more precise information about the state of the market.

7.6.4 onNews

For quant outcry, you will receive news periodically. Note that for other cases you will also receive news, which will also be sent through the MangoCore API, but unlike the news for quant outcry, this news will not be machine parseable, and likely be of no use to deal with programatically.

7.7 Submitting and Canceling Orders

The only case for which submitting or cancelling orders is allowed through the Mangocore API is the FX algorithmic trading case. In this case, users can submit and cancel orders with the Mangocore API. The rest of this section is in the context of the FX case.

For each message type, the user should have some specified function that takes in as parameters a `message` and also an object `order`, which initially starts off as an empty order. If the user wishes to buy a stock, they should use the method `addBuy` with several parameters, including the ticker of the stock, quantity, price and an optional unique token that identifies the order. If the user wishes to sell a stock, use the method `addSell` with the same parameters. If the user wishes to cancel an already existing order, then use the method `addCancel` with the unique token identifier of the order that they wish to cancel.

More detail about these functions is given in the documentation, along with a clarifying example.

7.8 Message Limits

There are limits for how frequently you may send messages: you may send a maximum of 25 pieces of information per second, where a piece of information is defined as a message, order, or cancel. For example, a message with 6 orders and 2 cancels counts as $1 + 6 + 2 = 9$ pieces of information. Any piece of information past this limit of 25 gets discarded. Also to note, messages are either entirely processed or entirely discarded. In other words, if you have sent 20 pieces of information in the last second and you attempt to send 10 more pieces in your next message, nothing in your next message will be processed (but we will record that you have tried to send 30 pieces of information in this past second).

If you send over 250 pieces of information in any second, your connection will be immediately killed (this isn't because we don't like you; we just don't want you to overload our network and starve other connections).

We will be monitoring server load and if it seems that our limits are too strict, we will relax them.

7.9 More Information

You will run your algorithms on your computers - we will not run them for you. Our server will be running somewhere (possibly in a public cloud, possibly in a box somewhere on MIT, etc). You may wish to optimize latency by trying to find where our server is.

If your client falls more than 5 seconds behind in receiving information, we will kill the connection.

7.10 Backtesting

A brief note about back-testing: profits that you see in your backtesting may not be the same as profits you may see during the actual competition. This may be due to competition (other market makers decreasing your edge), increased latencies (your algorithm will not be running on the same machine as the exchange), and other factors. You may want to simulate these pessimistic conditions in your backtesting to make it more rigorous.