

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно–Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ Г.И. Радченко
«__» _____ 2020 г.

Методические указания к выполнению выпускных квалификационных работ

РАЗРАБОТКА АГЕНТА ДЛЯ АВТОМАТИЧЕСКОЙ РАБОТЫ С ТОРГОВЫМИ
ИНСТРУМЕНТАМИ РЫНКА FOREX

Руководитель работы,
к.п.н., доцент каф. ЭВМ
_____ Ю.Г. Плаксина
«__» _____ 2020 г.

Автор работы,
студент группы КЭ–405
_____ Д.С. Брунеткин
«__» _____ 2020 г.

Нормоконтролёр,
ст. преп. каф. ЭВМ
_____ Сяськов С.В.
«__» _____ 2020 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно–Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Г.И. Радченко
«___» _____ 2020 г.

ЗАДАНИЕ

на выпускную квалификационную работу бакалавра
студенту группы КЭ–405
Брунеткину Дмитрию Сергеевичу
обучающемуся по направлению
09.03.01 «Информатика и вычислительная техника»

- 1. Тема работы:** «Разработка агента для автоматической работы с торговыми инструментами рынка Forex» утверждена приказом по университету от 24 апреля 2020г. № 627.
- 2. Срок сдачи студентом законченной работы:** 1 июня 2020 г.
- 3. Исходные данные к работе:**

Задача выпускной квалификационной работы: разработка торгового агента, который в автоматическом режиме будет проводить анализ торговых

инструментов на рынке Forex и при появлении соответствующих торговых ситуаций отсылать на сервер торговые приказы на покупку или продажу валют.

4. Перечень подлежащих разработке вопросов:

- разработка и проверка собственной гипотезы(алгоритма, на основе которого будет работать программное обеспечение);
- анализ существующих аналогов;
- проверка отказоустойчивости разрабатываемого программного обеспечения;
- проектирование автономного режима работы программного обеспечения, т.е. работы без непосредственного участия трейдера.

5. Дата выдачи задания: 1 сентября 2019 г.

Руководитель работы _____ /Ю.Г. Плаксина/

Студент _____ /Д.С. Брунеткин /

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2020	
Разработка модели, проектирование	01.04.2020	
Реализация системы	01.05.2020	
Тестирование, отладка, эксперименты	15.05.2020	
Компоновка текста работы и сдача на нормоконтроль	20.05.2020	
Подготовка презентации и доклада	25.05.2020	

Руководитель работы _____ /Ю.Г. Плаксина/

Студент _____ /Д.С. Брунеткин /

Аннотация

Д.С. Брунеткин. Разработка агента для автоматической работы с торговыми инструментами рынка Forex: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2020, 65 с., 11 ил., библиогр. список – 7 наим.

В рамках выпускной квалификационной работы проведен детальный анализ современных программных возможностей, которые предоставляет торговый терминал Forex. Для разработки программного комплекса использован встроенный язык программирования MQL5. Проводится выборка и анализ результатов работы торговой стратегии, рассматриваются преимущества и недостатки разработанного программного продукта - компьютерного ПО, представляющего из себя торговый алгоритм-систему, упрощающую работу на фондовом рынке. Доказывается способность предлагаемого разработанного решения к обеспечению стабильной работы и решению задач пользователя, которые предусмотрены программным комплексом.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	7
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	9
1.1 ПОСТАНОВКА ЗАДАЧИ	10
1.2 ОБЗОР АНАЛОГОВ	10
1.3 КРИТЕРИИ ОЦЕНКИ ТОРГОВОГО АГЕНТА	11
1.4 СООТВЕТСТВИЕ РАЗРАБОТАННОГО ТОРГОВОГО АГЕНТА КРИТЕРИЯМ ОЦЕНКИ	12
1.5 ВЫВОД	14
2. АНАЛИЗ И ВЫБОР ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ	14
3. ПРОЕКТИРОВАНИЕ	16
3.1 АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ	16
3.1.1 БЛОК ИНИЦИАЛИЗАЦИИ	17
3.1.2 РАСЧЕТНЫЙ БЛОК	19
3.1.3 БЛОК ЗАВЕРШЕНИЯ РАБОТЫ	21
3.2 РАЗРАБОТКА РАСЧЕТНО-АНАЛИТИЧЕСКИХ БЛОКОВ	22
3.3 СТРУКТУРА ТЕХНИЧЕСКОГО ИНДИКАТОРА	23
4. РЕАЛИЗАЦИЯ	26
4.1 РЕАЛИЗАЦИЯ ИНДИКАТОРОВ	26
4.1.1 СГЛАЖЕННОЕ СКОЛЬЗЯЩЕЕ СРЕДНЕЕ	26
4.1.2 ФРАКТАЛЫ	27
4.1.3 ИНДИКАТОР ALLIGATOR	28
4.1.4 ИНДИКАТОР НАПРАВЛЕНИЯ ТЕНДЕНЦИИ	30

4.2 СЧИТЫВАНИЕ ЗНАЧЕНИЙ ИНДИКАТОРОВ	31
4.3 ИНТЕРПРЕТАЦИЯ ТОРГОВЫХ СИГНАЛОВ ИНДИКАТОРОВ	34
4.4 ОРДЕРА И ИХ СВОЙСТВА	35
4.5 ПОЗИЦИИ	41
4.6 СДЕЛКИ	43
5. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ	44
5.1 УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ	45
5.2 МИНИМАЛЬНЫЕ СИСТЕМНЫЕ ТРЕБОВАНИЯ	45
5.3 ЗАПУСК В РАБОТУ	45
5.4 ВЫПОЛНЕНИЕ ПРОГРАММЫ	46
6. ЗАКЛЮЧЕНИЕ	50
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	52
ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД	53

ВВЕДЕНИЕ

В современном мире вычислительная техника оказывает огромное влияние на жизнь общества в целом и каждого отдельного человека. Использование технических новинок в самых разных аспектах жизни людей позволяет упростить и ускорить большинство задач, которые человеку приходится выполнять, позволяет держать непрерывную связь с родными, близкими и коллегами на огромных расстояниях, позволяет автоматизировать большинство рутинных процессов и направить весь свой творческий потенциал на решение новых, амбициозных задач.

Одним из аспектов деятельности человека является биржевое дело. Биржевая торговля получила широкое распространение еще в 17 веке, а на сегодняшний день она находится на пике своего развития и доступ на торговую площадку любой страны можно получить из любой точки мира в любое время суток. Это происходит благодаря использованию самых современных технологий для организации торговых площадок и предоставления доступа к ним посредством сети Интернет.

Ключевой фигурой в биржевой торговле являются трейдеры. Это люди, которые совершают сделки купли/продажи ценных бумаг на торговых площадках. Долгое время, пока в биржевую торговлю не были внедрены современные технологии, трейдеры могли совершать сделки, только находясь непосредственно на бирже, либо по телефону. Сегодня существует возможность выполнять финансовые операции со своего персонального компьютера при помощи специального программного обеспечения – торгового терминала. Развитие электронной торговли в свою очередь привело к тому, что появилась

возможность создавать автоматизированные торговые системы – программных роботов, которые по заложенному в них алгоритму выполняют финансовые операции на торговых площадках через торговый терминал.

Таким образом, сегодня разработан мощнейший инструмент, который способен в автоматическом режиме приносить серьезные финансовые прибыли, но только в том случае, если алгоритм торгового робота подходит для финансового инструмента, на котором робот будет работать.

В настоящее время существует множество торговых терминалов предоставляющих котировки цен в реальном времени. Одним из таких программных пакетов является MetaTrader 5 разработанный компанией MetaQuotes^[6]. В комплект программного пакета MetaTrader 5 входит язык программирования MQL5 на котором будет разработан торговый робот. В данной выпускной квалификационной работе, с использованием языка программирования MQL5, разработан торговый робот, который в автоматическом режиме анализирует состояние торговых инструментов рынка Forex и совершает торговые операции купли/продажи валют.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

На сегодняшний день биржевая торговля получила широкое распространение, и существуют тысячи компаний, которые эмитируют свои акции на рынки ценных бумаг, на эти акции создаются производные инструменты, которые тоже доступны для совершения торговых операций. Также существует возможность вкладывать средства в валюты разных стран, металлы, оборачивать средства на товарно-сырьевых рынках. Обилие торговых инструментов просто огромно, а трейдеры не в состоянии анализировать больше нескольких десятков инструментов в день. В случае если трейдер находится в рынке, он должен неотрывно следить за открытыми позициями, т. к. цены в любой момент могут изменить свое направление и если не закрыть позицию вовремя – будут понесены значительные финансовые потери. Такая напряженная работа ведет к постоянным стрессам, хронической усталости и как следствие ошибкам, а ошибки ведут к финансовым потерям или недополученной прибыли.

Существует еще одна проблема, которая связана с тем, что человеку свойственно ошибаться и трейдеры могут неверно трактовать торговые сигналы, вследствие чего несут финансовые потери, либо могут случайно ошибиться и совершить сделку большего или меньшего объема, чем требуется, это также может привести к плачевным последствиям. Торговая программа лишена всех этих недостатков, достаточно ее запустить и она без усталости, 7 дней в неделю и 24 часа в сутки будет работать, анализировать рыночные котировки и совершать сделки, если это будет предусмотрено торговой стратегией.

Таким образом, торговый робот позволяет расширить область деятельности трейдера на финансовых рынках, расширить предметную область для ведения своего бизнеса и работать с сотнями торговых инструментов на разных торговых площадках по всему миру круглые сутки. Количество торговых инструментов будет ограничиваться лишь размером депозита трейдера.

1.1. ПОСТАНОВКА ЗАДАЧИ

Задача выпускной квалификационной работы – разработка торгового агента, который в автоматическом режиме будет проводить анализ торговых инструментов на рынке Forex и при появлении соответствующих торговых ситуаций отсылать на сервер торговые приказы на покупку или продажу валют.

1.2 ОБЗОР АНАЛОГОВ

При проведении исследования рынка разработки торговых агентов, аналогичных представленному в данной выпускной квалификационной работе, выяснилось, что автоматизированные системы торговли на финансовых рынках являются закрытыми продуктами и используются исключительно внутри компаний, которые их разрабатывают.

Компании-разработчики торговых систем не афишируют свои продукты ввиду того, что разработанные и используемые ими алгоритмы и основанное на этих алгоритмах программное обеспечение приносят этим компаниям

регулярную финансовую прибыль. В связи с этим подобные разработки считаются интеллектуальной собственностью компаний и хранятся в строжайшем секрете.

В связи с тем, что автоматические торговые системы являются закрытыми продуктами, в данном разделе не удастся привести обзор аналогичных, разработанному в дипломном проекте продуктов, тем не менее, сформулируем основные критерии, которым должен удовлетворять разработанный торговый агент.

1.3 КРИТЕРИИ ОЦЕНКИ ТОРГОВОГО АГЕНТА

Круглосуточная бесперебойная работа. Данный критерий подразумевает круглосуточное ведение аналитической деятельности торговым агентом и поиск подходящих ситуаций для подачи торговых поручений на всех доступных торговых площадках.

Возможность быстрой взаимозамены, дополнения и отключения аналитических блоков программы. Программный код разработанного торгового агента должен быть реализован таким образом, чтобы в перспективе существовала возможность разрабатывать новые аналитические блоки для торгового агента и с минимальными затратами времени добавлять их в существующий программный код, либо заменять старые блоки.

Возможность регулировки параметров торговой стратегии, которую

реализует торговый агент. В разработанном торговом агенте должна быть предусмотрена возможность задания параметров реализованной торговой стратегии. Необходимо чтобы можно было задавать параметры при запуске торгового агента и изменять их в ходе работы торгового робота.

Наличие функционала, отвечающего за уведомление трейдера о наступлении тех или иных событий. Необходимо, чтобы в торговом агенте присутствовал функционал, реализующий уведомления для оператора, при наступлении каких либо событий на рынке (формирование торговой ситуации или ситуации при которой необходимо закрыть сделку, совершение торговой операции, отправка торгового поручения на сервер и т.д.).

Установка и сопровождение торговых позиций. Необходимо, чтобы торговый агент не только отправлял торговые поручения, совершая таким образом сделки, но и сопровождал открытые позиции, т.е. в каждый момент времени торговый агент должен предоставлять информацию о количестве открытых позиций, торговых инструментах на которых открыты позиции и другой информации, относящейся к текущим сделкам.

1.4 СООТВЕТСТВИЕ РАЗРАБОТАННОГО ТОРГОВОГО АГЕНТА КРИТЕРИЯМ ОЦЕНКИ

Проведем анализ соответствия торгового агента представленным требованиям.

Таблица 1 – Соответствие торгового агента критериям оценки

Критерий	Реализация
Круглосуточная бесперебойная работа	Да

Продолжение таблицы 1.

Критерий	Реализация
Возможность быстрой взаимозамены, дополнения и отключения аналитических блоков программы	Да
Возможность регулировки параметров торговой стратегии, которую реализует торговый агент	Да
Наличие функционала, отвечающего за уведомление трейдера о наступлении тех или иных событий	Да
Установка и сопровождение торговых позиций	Да

Как видно из таблицы 1 – разработанный торговый агент удовлетворяет всем критериям.

1.5 ВЫВОД

В рамках данного раздела был проведен анализ предметной области, выявлены и сформулированы требования, которым должен соответствовать разрабатываемый программный продукт.

2. АНАЛИЗ И ВЫБОР ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ

Так как задача, поставленная в выпускной квалификационной работе, является довольно специфической – реализовать получение доступа к рыночным котировкам и возможность выполнения торговых операций, то необходимо выбрать наиболее распространенную на рынке брокерских услуг платформу, которая будет удовлетворять всем необходимым требованиям для разработки и запуска на ней торгового агента, а именно:

- доступ к торгам не только на рынок Forex, но и на Фьючерсные и биржевые рынки;
- наличие встроенного языка программирования;
- наличие тестера стратегий, который позволяет тестировать разрабатываемые алгоритмы на исторических данных;
- возможность разрабатывать, производить отладку и тестировать торговые роботы во встроенной среде разработки.

В качестве наиболее подходящей платформы для разработки торгового агента выбран программный пакет MetaTrader 5.[3] Программный пакет MetaTrader является одним из самых популярных программных решений, предназначенных для организации брокерского обслуживания, а также соответствует всем представленным требованиям.[4]

Главным аналогом Metatrader 5 является программный комплекс Quik. Quik 7.1 – одна из популярных в России и Украине платформ для доступа на национальные биржи, однако, она имеет существенные недостатки. Так,

например, для работы терминал доступен только 8 часов в сутки – в рабочее время биржи, в России это с 10:00 до 18:00 по московскому времени. К тому же, Quik имеет встроенный алгоритмический язык программирования под названием QPILE, который по сравнению с языком MQL5, основанном на C++, имеет значительно более скромный функционал.

Таким образом, можно сделать вывод, что MetaTrader – это наиболее подходящая платформа для разработки торгового агента и дальнейшего его использования. Также использование данной платформы открывает большие перспективы ввиду того, что MetaTrader предоставляет доступ на совершенно разные торговые площадки.

3. ПРОЕКТИРОВАНИЕ

Логически структуру торгового робота можно разбить на несколько блоков. Для полноценного функционирования робота достаточно наличия одного или нескольких расчетно–аналитических блоков, которые будут отвечать за анализ поступающих котировок и выполнение математических расчетов. Также должен быть разработан блок формирования, отправки и сопровождения торговых поручений.

3.1. АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ

Структура разрабатываемого в данной выпускной квалификационной работе торгового агента схематично представлена на рисунке 1.

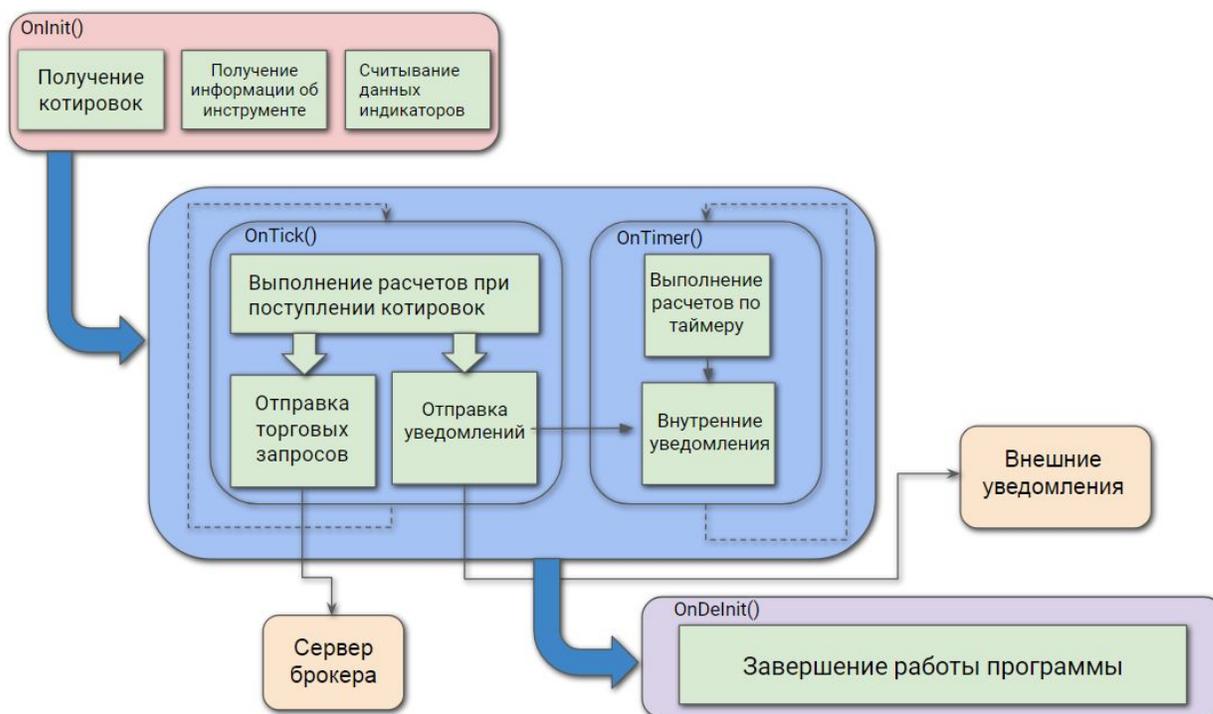


Рисунок 1 – Обобщенная структура торгового робота

В торговом агенте можно выделить три основных блока:

- блок инициализации;
- расчетный блок;
- блок завершения работы.

3.1.1 БЛОК ИНИЦИАЛИЗАЦИИ

Как видно на рисунке 1 – в первом блоке инициализации выполняются все предварительные расчеты, которые необходимы для корректного запуска и дальнейшей работы программы. Здесь происходит создание хендлов всех используемых технических индикаторов.

Хендл – это уникальный идентификатор технического индикатора, созданный для более быстрого обращения к данным индикатора. Создание хендла для каждого индикатора на этапе инициализации торгового агента необходимо для того, чтобы в дальнейшем обращаться к данным индикаторов через заранее созданные хендлы, а не создавать их при каждом поступлении котировок.[1]

В листингах 1 и 2 представлен пример реализации методов класса Indicator. Метод HandleCreate() отвечает за создание хендла индикатора, а метод GetData() заполняет массив, переданный по ссылке расчетными данными индикатора.

Листинг 1 – Создание хендла

```
int Indicator::HandleCreate(MqlParam &parametr[]){ // Создание
отдельного handle
int size = ArraySize(parametr); // привязка массива к переменной
if (size > 0 && SetCountParams(size) == true) // условие на размер
массива и его //изменение
{
handle = IndicatorCreate(symbol, TimeFrame, GetType(),
GetCountParams(),
parametr);
return (handle);
}
else
{
return(-1);
}
}
```

Листинг 2 – Получение данных от индикатора

```
//получение результата запроса на вызов данных из индикатора
bool Indicator::GetData(int numBuffer, int start, int count, double
&array[]){
if (CopyBuffer(GetHandle(), numBuffer, start, count, array) < 0)
{
    return (false);
}
else
{
    return (true);
}
}
```

Помимо создания хендлов технических индикаторов в первом блоке производится инициализация объектов ответственных за открытие и сопровождение торговых сделок. А также объектов реализующих расчетно–аналитические блоки программы.

Программно данный блок реализуется в функции OnInit() торгового робота, обращение к которой происходит при активации торгового робота, а также при смене временных периодов на которых будет производиться работа.

3.1.2 РАСЧЕТНЫЙ БЛОК

Расчетный блок содержит в себе основную часть программы.

Особенность

этого блока заключается в том, что на протяжении всего того времени, что будет

запущен торговый робот – этот блок будет работать циклично, запускаясь заново

каждый раз при поступлении новых котировок.

Именно в этом блоке на протяжении всей работы торгового робота выполняются математические расчеты, сравниваются финансовые показатели и уровни котировок, проводится технический анализ, отдаются торговые приказы и сопровождаются исполненные торговые сделки, т.е. принимается решение, оставлять ли исполненные ордера открытыми или их нужно закрыть.

На схеме из рисунка 1 видно, что расчетный блок состоит из двух функций, это OnTick() и OnTimer(). Вторая является необязательной. Таймер можно использовать только при необходимости, а OnTick() запускается каждый раз при поступлении котировок.[2] Именно в этой функции выполняются все основные расчеты.

В листинге 3 приведен фрагмент текста программы, в котором реализована проверка проверки наличия основного сигнала на открытие торговой позиции и закрытие существующей и при возникновении такого сигнала отправляет торговое поручение на сервер брокера:

Листинг 3 – Проверка наличия основного сигнала на открытие позиции

```
if (d1.GetSignals()){// если найден сигнальный фрактал
// и он направлен вверх и цена установки ордера преодолена, то
if(d1.GetFStype() == "UP" && d1.GetFSprice() <
min.iClose(NULL,0,0))
{
// Закрываем открытые позиции по инструменту
if(pControl.IsSetSymbol(Symbol()) == true &&
pControl.GetType(Symbol())
== "SELL")
{
if(td.PositionClose(Symbol(),5))
{
Print(Symbol(), "CLOSE SELL");
}
else
{
Print(Symbol(), " ERROR CLOSE SELL");
}
}
// открываем позицию
if (qBars < Bars)
{
qBars = Bars;
td.Buy(0.5);
}
}
}
```

3.1.3 БЛОК ЗАВЕРШЕНИЯ РАБОТЫ

Блок завершения работы выполняется при обращении к служебной функции OnDeinit() (см. рисунок 1). Данный блок запускается либо перед завершением работы торгового робота, либо перед тем как переключиться на другой временной интервал и выполняет все операции необходимые для корректного завершения работы торгового агента.

В блоке завершения работы уничтожается таймер и в общий лог, отражающий процесс работы торгового робота, отсылается уведомление о завершении работы.

3.2. РАЗРАБОТКА РАСЧЕТНО–АНАЛИТИЧЕСКИХ БЛОКОВ

Расчетно–аналитические блоки предназначены для проведения анализа поступающих с рынка котировок, их обработки в соответствии с определенными разработчиком формулами и передачи результирующих данных торговому роботу для дальнейшей работы с ними.

Расчетно–аналитический блок представляет собой совокупность технического индикатора как отдельной программы на языке MQL и специально созданной программной оболочки для этого индикатора, которая входит в состав торгового робота и получает данные от технического индикатора. Схематично взаимодействие торгового робота с техническими индикаторами представлено на рисунке 2.

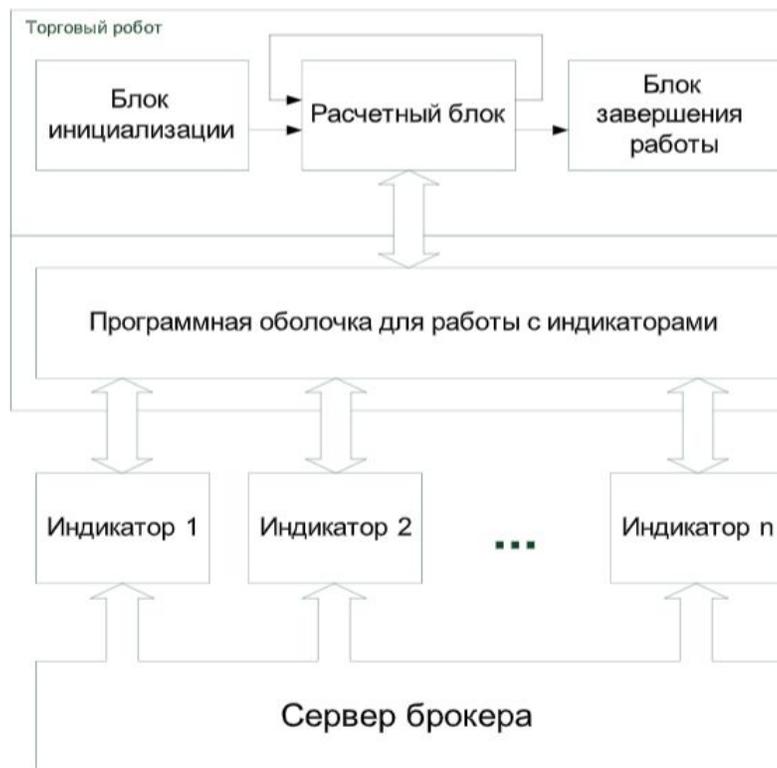


Рисунок 2 – Взаимодействие торгового робота с техническими индикаторами.

3.3. СТРУКТУРА ТЕХНИЧЕСКОГО ИНДИКАТОРА

Технический индикатор представляет собой отдельную, самостоятельную программу, предназначенную для проведения анализа рыночных котировок в соответствии с заложенными в него формулами. Результат анализа индикатора можно отобразить несколькими способами:

- в виде графика одной или нескольких ломаных линий (самый распространенный вариант);
- в виде разноцветной гистограммы, где каждая цветовая зона может обозначать различные типы движения цен;

- в виде графических символов (стрелочные указатели, буквенно-цифровые обозначения и др.).

На рисунке 3 представлены варианты графического отображения индикаторов.



Рисунок 3 – Пример отображения индикаторов на графике в торговом терминале

Кроме графического представления к значениям индикаторов предоставляется доступ извне посредством индикаторных буферов. Буфер индикатора представляет собой одномерный массив вещественного типа (double), каждый i -ый элемент которого соответствует i -ому бару на ценовом графике. Нумерация элементов буферного массива производится с нуля, от конца графика к началу, т.е. последний элемент имеет индекс 0, предпоследний – 1 и т.д.

На рисунке 4 представлена общая структура индикатора.



Рисунок 4 – Использование цены в качестве индикатора

Как видно из схемы на рисунке 4 – структура индикатора схожа со структурой торгового робота но, тем не менее, есть некоторые отличия и они довольно существенны. В индикаторе также как и в торговом роботе присутствуют блоки инициализации и расчета, но отсутствует блок завершения работы. Это обусловлено тем, что никаких предварительных и завершающих операций, кроме объявления буферов и назначения способа их отображения не производится. Объявление буферов происходит в блоке инициализации. В расчетном блоке на каждом тике происходит пересчет формул и заполнение

индикаторного буфера. При выключении индикатора расчеты просто перестают выполняться, буферы автоматически удаляются, и график индикатора пропадает из окна рабочего инструмента.

4. РЕАЛИЗАЦИЯ

4.1. РЕАЛИЗАЦИЯ ИНДИКАТОРОВ

В разрабатываемой выпускной квалификационной работе используется несколько индикаторов, которые, в свою очередь, используют следующие математические алгоритмы:

- фракталы;
- аллигатор (условное название) – представляет собой совокупность сглаженных скользящих средних;
- алгоритм отображения направления тенденции.

4.1.1 СГЛАЖЕННОЕ СКОЛЬЗЯЩЕЕ СРЕДНЕЕ

Сглаженное скользящее среднее (Smoothed Moving Average, SMMA) лежит в основе нескольких используемых в данной торговой системе индикаторов. Первое значение сглаженного скользящего среднего рассчитывается, как простое скользящее среднее.

Скользящая средняя, скользящее среднее — технический индикатор в основе которого лежит анализ поведения котировок ценной бумаги и их скользящего среднего. Скользящая средняя является фильтром низких частот, то есть пропускает низкочастотную активность (долгосрочные циклы и их

линии трендов), отсекая высокочастотные — случайные колебания.

Для использования индикатора одновременно совмещаются графики цены и его скользящей средней. Вид скользящей средней и период построения (количество временных периодов, по которому осуществляется усреднение) иногда называется порядком или временным окном, или длиной, выбирается трейдером на своё усмотрение и зависит от горизонта торговли, волатильности рынка и инструмента. Причём, в разные промежутки времени могут использоваться различные виды скользящих средних и разные периоды построения. Выбор данных параметров считается настолько сложным, что стал отдельной ветвью технического анализа, однако в общем случае признаётся, что чем больше время прогноза, тем больший порядок необходимо выбрать для скользящей средней, и наоборот.

4.1.2. ФРАКТАЛЫ

Фракталы – это технический индикатор, позволяющий обнаруживать впадину или вершину графика цены. Фрактал вверх технически описывается как серия из, как минимум, пяти последовательных баров, в которой непосредственно перед самым высоким максимумом и сразу же после него находятся по два бара с более низкими максимумами. Противоположная конфигурация (серия из пяти баров, в которой непосредственно перед самым низким минимумом и сразу же после него находятся по два бара с более высокими минимумами) соответствует фракталу вниз. На графике фракталы имеют значения High и Low и отмечены стрелками вверх или вниз соответственно. На рисунке 5 в виде стрелочных указателей изображен индикатор “Фракталы”. приведен в листинге А.1 приложения А.



Рисунок 5 – Индикатор “Fractals”

Сигналы технического индикатора Fractals необходимо отфильтровывать с помощью технического индикатора Аллигатор. Другими словами, не следует заключать сделку на покупку, если фрактал находится ниже красной линии Аллигатора, и не следует заключать сделку на продажу, если фрактал находится выше красной линии Аллигатора. После того как сигнал фрактала сформирован и имеет силу, что определяется его позицией вне линий Аллигатора, он остается сигналом до тех пор, пока не будет пробит или пока не возникнет более свежий сигнал фрактала.

4.1.3. ИНДИКАТОР ALLIGATOR

Технический индикатор Alligator в совокупности с индикатором Fractals, образует основную группу индикаторов, которая формирует первичные сигналы для входа в рынок.

Alligator – это комбинация сглаженных скользящих средних. Индикатор представлен на рисунке 6.



Рисунок 6 – Индикатор “Alligator”

Синяя линия – для временного периода, который использовался для построения графика (13–периодное сглаженное скользящее среднее, сдвинутое на 8 баров в будущее). Расчет представлен формулой:

$$\text{ALLIGATORS Blue} = \text{SMMA} (\text{MEDEAN PRICE}, 13, 8) \quad (1)$$

Расчет скользящей средней без сдвига в будущее представлен формулой:

$$\text{MEDIAN PRICE} = (\text{HIGH} + \text{LOW}) / 2 \quad (2)$$

Красная линия – для значимого временного периода на порядок ниже (8–периодное сглаженное скользящее среднее, сдвинутое на 5 баров в будущее). Расчет представлен формулой:

$$\text{ALLIGATORS Red} = \text{SMMA} (\text{MEDEAN PRICE}, 8, 5) \quad (3)$$

Зеленая линия – для значимого временного периода, который ниже еще на один порядок (5–периодное сглаженное скользящее среднее, сдвинутое на 3 бара в будущее). Расчет представлен формулой:

$$\text{ALLIGATORS Green} = \text{SMMA} (\text{MEDEAN PRICE}, 5, 3), \quad (4)$$

где MEDIAN PRICE – медианная цена;

HIGH – максимальная цена бара;

LOW – минимальная цена бара;

SMMA (A, B, C) – сглаженное скользящее среднее;

A – сглаживаемые данные;

B – период сглаживания;

C – сдвиг в будущее.

4.1.4. ИНДИКАТОР НАПРАВЛЕНИЯ ТЕНДЕНЦИИ

Индикатор направления тенденции формирует вторичные сигналы для наращивания объема уже открытой торговой позиции. Сигналы данного индикатора рассматриваются только в том случае, если уже существует сделка, совершенная по сигналу от основной группы индикаторов. Технический индикатор определения направления тенденции – это 34–периодное простое скользящее среднее, построенное по средним точкам баров $(H+L)/2$, которое вычтено из 5–периодного простого скользящего среднего, также построенного по средним точкам $(H+L)/2$:

$$\text{MEDIAN PRICE} = (\text{HIGH} + \text{LOW}) / 2, \quad (5)$$

$$\text{AO} = \text{SMA}(\text{MEDIAN PRICE}, 5) - \text{SMA}(\text{MEDIAN PRICE}, 34), \quad (6)$$

где MEDIAN PRICE – медианная цена;

HIGH – максимальная цена бара;

LOW – минимальная цена бара;

SMA – простая скользящая средняя

4.2 СЧИТЫВАНИЕ ЗНАЧЕНИЙ ИНДИКАТОРОВ

Для считывания значений технических индикаторов разрабатывается специальная оболочка для работы с техническими индикаторами. Она изображена на рисунке 2. Существует несколько способов получения данных индикаторов из торгового агента.

Первый способ, это использование функции `iCustom()` (листинг 4), которая принимает в качестве параметров имя вызываемого индикатора и его параметры, а возвращает хендл.

Листинг 4 – Пример использования функции `iCustom()`

```
MA1 = iCustom(NULL,          // торговый инструмент
              0,             // Временной период
              "Moving Average", // Имя индикатора
              5,             // Период расчета индикатора
              0,             // Смещение
              MODE_SMA,      // Метод расчета
              PRICE_CLOSE    // считаем по ценам закрытия
              );
```

В результате успешного вызова этой функции в `MA1` будет храниться хендл индикатора по которому в дальнейшем можно будет получить расчетные данные индикатора.

Этот способ является достаточно быстрым, но он не обеспечивает необходимой гибкости, и приемлем только в том случае, если в процессе работы не возникнет необходимости обращаться к нескольким разным индикаторам. Наиболее универсальным способом является работа через функцию `IndicatorCreate()`

(листинг 5), которая принимает в качестве параметров имя индикатора и его параметры в виде структуры.

Листинг 5 – Получение хендла индикатора при помощи IndicatorCreate()

```
// установка количества параметров вызываемого индикатора
ArrayResize(params,4);
// Период быстрой МА
params[0].type =TYPE_INT;
params[0].integer_value=5;
// Смещение
params[1].type =TYPE_INT;
params[1].integer_value=0;
// Метод расчета: простое усреднение
params[2].type =TYPE_INT;
params[2].integer_value=MODE_SMA;
// Тип цен для расчета: цены закрытия
params[3].type =TYPE_INT;
params[3].integer_value=PRICE_CLOSE;

MA1 = IndicatorCreate(
Symbol(),           // торговый инструмент
0,                 // Таймфрейм
IND_MA,           // Тип индикатора
4,                // Количество параметров
params            // Массив параметров
);
```

Этот метод является более громоздким, но его универсальность позволяет свободно создавать, уничтожать и создавать новые хендлы индикаторов прямо во время работы торгового робота. В разрабатываемой выпускной

квалификационной работе реализован второй способ работы с индикаторами.

Реализация представлена в листинге 6.

Листинг 6 – Реализация работы с индикаторами

```
class Indicator{
private:
    int                handle;
    int                countParams;
    ENUM_INDICATOR    indicatorType;

    string            symbol;
    ENUM_TIMEFRAMES  TimeFrame;

    // -- Вернет, указанное ранее, количество параметров
    int                GetCountParams();
    // -- Метод установки количества параметров индикатора
    bool SetCountParams(int count);

public:
    // ==
    // ==  Группа методов для установки/сброса параметров
индикатора
    // ==
    // -- Установка параметров для IndicatorCreate
    Void                SetParam(ENUM_INDICATOR type = IND_CUSTOM,
string sm = NULL, ENUM_TIMEFRAMES TF = 0);
    // -- Метод установки одного из параметров индикатора
    int                HandleCreate(MqlParam &parametr[]);
    // -- Сбросить все установки для индикатора
    // void                Reset();
```

```

// -- Сбрасывает установки индикатора и сбрасывает хендл и
удаляет расчетную часть.
bool Release();
// == Группа методов для получения информации об индикаторе
// -- Вернет хендл индикатора
int GetHandle();
// -- Вернет тип индикатора
ENUM_INDICATOR GetType();
// -- Заполняет массив, переданный по ссылке, данными
индикатора
bool GetData(int numBuffer, int start, int
count, double &array[]);
};

```

4.3. ИНТЕРПРЕТАЦИЯ ТОРГОВЫХ СИГНАЛОВ ИНДИКАТОРОВ

Индикаторы как самостоятельные аналитические единицы не представляют собой никакой ценности. Значимые сигналы можно получить лишь при использовании верной комбинации индикаторов. Таким образом, для успешной работы торгового агента необходима верная интерпретация сигналов, которые формируются индикаторами.

Как правило, из всех сигналов один можно выделить как основной. Он формируется комбинацией главных индикаторов индикаторов. В свою очередь, какие индикаторы являются главными, а какие побочными решает разработчик ПО. Если сформировалась подходящая ситуация из комбинации сигналов индикаторов, то это считается сигналом на открытие позиции на покупку (buy). Для открытия позиций на продажу формируется абсолютно противоположный сигнал, т.е. при достижении ценой уровня этого ордера будет открыта позиция на продажу (sell).

4.4. ОРДЕРА И ИХ СВОЙСТВА

Результатом проводимых торговых операций являются две сущности, это торговые позиции и ордера.

Приказы на проведение торговых операций оформляются ордерами. Каждый ордер имеет множество свойств, доступных для чтения. Основные свойства приведены в таблице 2.

Таблица 2 – Свойства ордеров

Идентификатор	Описание	Тип
ORDER_TIME_SETUP	Время постановки ордера	datetime
ORDER_TYPE	Тип ордера	ENUM_ORDER_TYPE
ORDER_STATE	Статус ордера	ENUM_ORDER_STATE
ORDER_TIME_EXPIRATION	Время истечения ордера	datetime
ORDER_TIME_DONE	Время исполнения или снятия ордера	>>
ORDER_TYPE_FILLING	Тип исполнения по остатку	ENUM_ORDER_TYPE_FILLING
ORDER_TYPE_TIME	Время жизни ордера	ENUM_ORDER_TYPE_TIME

Продолжение таблицы 2.

Идентификатор	Описание	Тип
ORDER_MAGIC	Идентификатор эксперта, выставившего ордер (предназначен для того, чтобы каждый эксперт выставлял свой собственный уникальный номер)	long
ORDER_POSITION_ID	Идентификатор позиции, который ставится на ордере при его исполнении. Каждый исполненный ордер порождает сделку, которая открывает новую или изменяет уже существующую позицию.	>>

При отправке торгового запроса, для некоторых операций необходимо указать тип ордера – на покупку или продажу. Тип ордера указывается в поле type специальной структуры MqlTradeRequest, и может принимать различные значения, перечень которых приведен в таблице 3.

Таблица 3 – Перечисление ENUM_ORDER_TYPE. Типы ордеров

Идентификатор	Описание
ORDER_TYPE_BUY	Рыночный ордер на покупку
ORDER_TYPE_SELL	Рыночный ордер на продажу
ORDER_TYPE_BUY_LIMIT	Отложенный ордер Buy Limit
ORDER_TYPE_SELL_LIMIT	Отложенный ордер Sell Limit
ORDER_TYPE_BUY_STOP	Отложенный ордер Buy Stop
ORDER_TYPE_SELL_STOP	Отложенный ордер Sell Stop
ORDER_TYPE_BUY_STOP_LIMIT	По достижении цены ордера выставляется отложенный ордер Buy Limit по цене StopLimit
ORDER_TYPE_SELL_STOP_LIMIT	По достижении цены ордера выставляется отложенный ордер Sell Limit по цене StopLimit

Взаимодействие клиентского терминала и торгового сервера для проведения операций постановки ордеров производится посредством торговых запросов. Полный цикл прохождения торгового запроса от пользовательского терминала до исполнения можно наблюдать на рисунке 7. Красной линией отмечен возврат торговых приказов, которые были отменены трейдером, либо отклонены.



Рисунок 7 – Процесс формирования, обработки и исполнения торгового приказа

Запрос представлен специальной предопределенной структурой MqlTradeRequest, которая содержит все поля, необходимые для заключения торговых сделок. Описание полей структуры MqlTradeRequest представлено в таблице 4. Результат обработки запроса представлен структурой MqlTradeResult.

Таблица 4 – Описание полей структуры MqlTradeRequest

Поле	Описание
action	Тип торговой операции. Значение может быть одним из значений перечисления ENUM_TRADE_REQUEST_ACTIONS
magic	Идентификатор торгового агента. Позволяет организовать аналитическую обработку торговых ордеров. Каждый торговый агент может выставлять свой собственный уникальный идентификатор при отправке торгового запроса
order	Тикет ордера. Требуется для модификации отложенных ордеров
symbol	Имя торгового инструмента, по которому выставляется ордер. Не требуется при операциях модификации ордеров и закрытии позиций
volume	Запрашиваемый объем сделки в лотах. Реальное значение объема при открытии сделки будет зависеть от типа ордера по исполнению.
price	Цена, при достижении которой ордер должен быть исполнен. Для рыночных ордеров по инструментам с типом исполнения “Market Execution” (SYMBOL_TRADE_EXECUTION_MARKET), имеющих тип TRADE_ACTION_DEAL, указание цены не требуется
stoplimit	Цена, по которой будет выставлен отложенный StopLimit ордер, при достижении ценой значения price (это условие является обязательным). До этого момента отложенный ордер в торговую систему не выводится

Продолжение таблицы 4.

Поле	Описание
sl	Цена, по которой сработает Stop Loss ордер при движении цены в неблагоприятном направлении
tp	Цена, по которой сработает Take Profit ордер при движении цены в благоприятном направлении
deviation	Максимально приемлемое отклонение от запрашиваемой цены, задаваемое в пунктах
type	Тип ордера. Значение может быть одним из значений перечисления ENUM_ORDER_TYPE
type_filling	Тип ордера по исполнению. Значение может быть одним из значений ENUM_ORDER_TYPE_FILLING
type_time	Тип ордера по истечению. Значение может быть одним из значений ENUM_ORDER_TYPE_TIME
expiration	Срок истечения отложенного ордера (для ордеров типа ORDER_TIME_SPECIFIED)
comment	Комментарий к ордеру

В ответ на торговый запрос постановки ордера в торговую систему, торговый сервер возвращает данные, содержащие информацию о результате обработки торгового запроса в виде специальной predefined структуры MqlTradeResult. Описание полей структуры результата запроса приведено в таблице 5.

Таблица 5 – Описание полей структуры MqlTradeResult

Поле	Описание
retcode	Код возврата торгового сервера
deal	Тикет сделки, если она совершена. Сообщается при торговой операции TRADE_ACTION_DEAL
order	Тикет ордера, если он выставлен. Сообщается при торговой операции TRADE_ACTION_PENDING
volume	Объем сделки, подтвержденный брокером. Зависит от типа ордера по исполнению
price	Цена в сделке, подтвержденная брокером. Зависит от поля deviation в торговом запросе и/или от типа торговой операции
bid	Текущая рыночная цена предложения (цены реквоты)
ask	Текущая рыночная цена спроса (цены реквоты)
comment	Комментарий брокера к операции (по умолчанию заполняется расшифровкой)

4.5 ПОЗИЦИИ

Результатом исполнения ордеров и совершения торговых операций является открытие позиции, изменение ее объема и/или направления, или ее ликвидация. Торговые операции проводятся на основании ордеров, отправляемых в виде торговых запросов. Для каждого финансового инструмента (символа) возможна только одна открытая позиция. Позиции имеют набор свойств. Основные свойства представлены в таблице 6.

Таблица 6 – набор свойств торговых позиций.

Идентификатор	Описание	Тип
POSITION_VOLUME	Объем позиции	double
POSITION_PRICE_OPEN	Цена позиции	>>
POSITION_SL	Уровень Stop Loss для открытой позиции	>>
POSITION_TP	Уровень Take profit для открытой позиции	>>
POSITION_COMMISON	Уровень комиссии	>>
POSITION_SWAP	Накопленный своп	>>
POSITION_PROFIT	Текущая прибыль	>>
POSITION_SYMBOL	Символ, по которому открыта позиция	string
POSITION_COMMENT	Комментарий к позиции	>>
POSITION_TIME	Время открытия позиции	datetime
POSITION_TYPE	Тип позиции	ENUM_POSITION_TYPE
POSITION_MAGIC	Magic number для позиции	long
POSITION_IDENTIFIER	Идентификатор позиции – это уникальное число, которое присваивается каждой вновь открытой позиции	>>

Направление открытой позиции (покупка или продажа) определяется значением из перечисления ENUM_POSITION_TYPE. В отличие от ордеров, позиции имеют только два типа: buy и sell.

Таблица 7 – Типы позиций

Идентификатор	Описание
POSITION_TYPE_BUY	Покупка
POSITION_TYPE_SELL	Продажа

4.6 СДЕЛКИ

Сделка является отражением факта совершения торговой операции на основании ордера, содержащего торговый приказ. Каждая сделка описывается свойствами, позволяющими получить информацию о ней. Свойства сделок представлены в таблице 8.

Каждая сделка характеризуется типом, возможные значения перечислены в таблице 9.

Таблица 8 – Свойства сделок

Идентификатор	Описание	Тип
DEAL_VOLUME	Объем сделки	double
DEAL_PRICE	Цена сделки	>>
DEAL_COMMISSION	Комиссия по сделке	>>
DEAL_SWAP	Накопленный своп при закрытии	>>
DEAL_PROFIT	финансовый результат сделки	>>

Таблица 9 – Типы сделок

Идентификатор	Описание
DEAL_TYPE_BUY	Покупка
DEAL_TYPE_SELL	Продажа
DEAL_TYPE_BALANCE	Начисление баланса
DEAL_TYPE_CREDIT	Начисление кредита
DEAL_TYPE_CHARGE	Дополнительные сборы
DEAL_TYPE_CORRECTION	Корректирующая запись

Сделки различаются не только по типам, которые описаны в таблице 8, но и по способу изменения позиции. Это может быть простое открытие позиции или наращивание объема ранее открытой позиции (вход в рынок), закрытие позиции сделкой противоположного направления соответствующим объемом (выход из рынка) или переворот позиции в том случае, когда объем сделки в противоположном направлении перекрывает объем ранее открытой позиции.

5. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

Целью данного дипломного проекта является внедрение автоматизации процессов в биржевую торговлю. На протяжении дипломного проектирования проводится детальное изучение процессов происходящих во время ведения торгов на различных финансовых рынках, и подробно рассматриваются все специфические тонкости работы. В результате проделанной за время дипломного проектирования работы должен быть разработан торговый агент на базе программного пакета MetaTrader 5. В основе разрабатываемого торгового

агента должен лежать алгоритм, с помощью которого будет происходить отправка торговых поручений на сервер брокера.

5.1. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Для установки, настройки и работы программной системы пользователь должен обладать профессиональными углубленными навыками работы на персональном компьютере, обладать навыками работы с программным комплексом MetaTrader 5 и навыками установки и настройки приложений в операционной системе Windows. Торговый агент выполнен в виде приложения, написанного на встроенном в программный пакет MetaTrader языке MQL5.

5.2. МИНИМАЛЬНЫЕ СИСТЕМНЫЕ ТРЕБОВАНИЯ

Минимальные требования для успешной установки и работы торгового агента:

- персональный компьютер под управлением операционной системы Windows;
- процессор не ниже Intel Pentium 4 или эквивалентный ему по производительности процессор другой архитектуры;
- 1Гб оперативной памяти или более;
- установленный программный пакет MetaTrader 5;
- наличие доступа к сети Интернет со скоростью не менее 512кб/с.

5.3. ЗАПУСК В РАБОТУ

Для установки и запуска торгового агента необходим установленный на рабочем компьютере программный пакет MetaTrader 5.

Для того чтобы установить программный пакет MetaTrader 5, необходимо:

1. Скачать его инсталлятор с сайта компании–разработчика.
2. Запустить процесс установки. Пользователю будет предложено выбрать директорию для установки.
3. После выбора директории будет произведена загрузка всех необходимых для установки MetaTrader файлов.
4. После завершения установки нужно нажать на кнопку «Готово» и запустить торговый терминал.
5. Для того, чтобы торговый робот стал доступен в терминале, нужно поместить файл с его исходным текстом в директорию Папка_торгового_терминала\MQL5\Experts\, а подключаемые библиотеки в директорию Папка_торгового_терминала\MQL5\Include\.
6. Для запуска торгового робота необходимо в терминале, в окне Навигатор/общие или Навигатор/избранное вызвать контекстное меню для нужного торгового агента и выбрать пункт «Присоединить к графику». Теперь торговый робот запущен. Процесс запуска и работы будет отображаться в окне ведения логов.

5.4 ВЫПОЛНЕНИЕ ПРОГРАММЫ

Изначально разрабатываемый торговый агент планировался как торговая система, уведомляющая трейдера о наступлении подходящей ситуации на рынке для совершения ордера. Эту идею удалось воплотить в рамках данной выпускной квалификационной работы и поэтому данная разработка нуждается в минимальном сопровождении со стороны человека.

Алгоритм работы программы следующий:

В первую очередь производится анализ группы индикаторов signals и DT–ZigZag, 3–Level–Semafor.

Индикатор DT-ZigZag предназначен для отображения локальных экстремумов наиболее крупных колебательных движений цены на рынке на текущем таймфрейме. При возникновении подходящей ситуации индикатор отображает такой экстремум чертой желтого цвета.

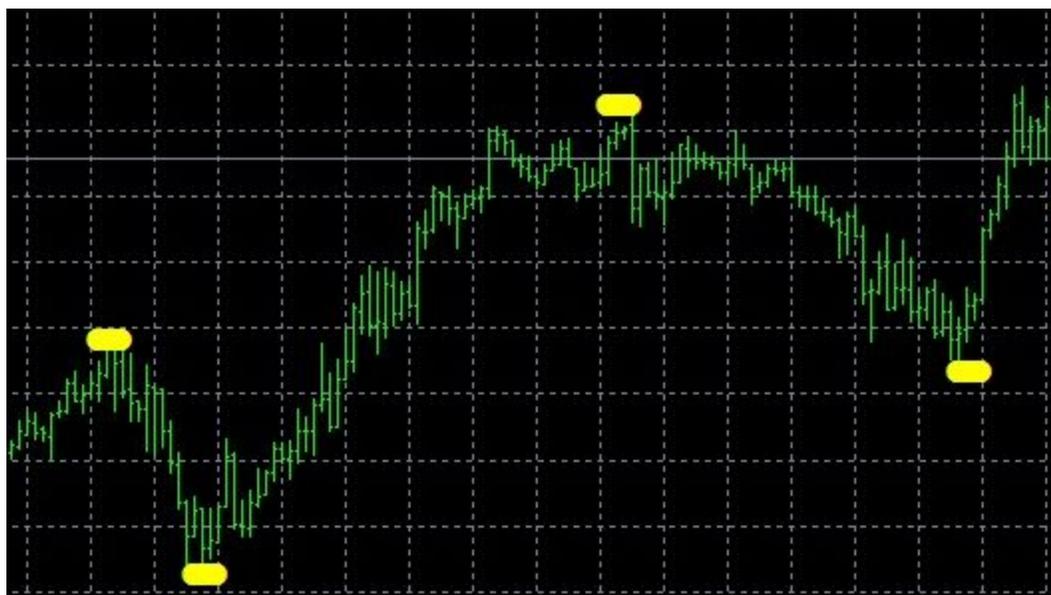


Рисунок 8 – Пример отображения индикатора DT-ZigZag.

Индикатор 3-Level-Semafor также отражает локальные минимумы и максимумы. Как видно из названия, он определяет 3 уровня этих уровней:

- маленькие. Небольшие колебания цены.
- средние. Колебания средней величины.
- большие. Наибольшие локальные экстремумы на заданном временном интервале.

Каждый из уровне обозначается соответствующей цифрой от 1 до 3, в зависимости от величины минимума или максимума.



Рисунок 9 – Пример отображения индикатора 3–Level–Semafor.

В связи с тем, что первое появление локального экстремума не означает разворот цены в данной точке (В связи с перерисовкой индикатора), ориентироваться только на них нельзя. На данном этапе создается дополнительный фильтр для определения подходящих торговых ситуаций. Для этого используется индикатор Shark signals. Этот индикатор обозначает экстремумы более мелких колебательных движений внутри крупных колебательных движений, которые, в свою очередь, рисуются выше указанными индикаторами, но его простое наложение на график не приведет к желаемому результату, т.к. он обладает такими же недостатками, что и остальные индикаторы, т.е. перерисовывается.

Shark signals работает по следующему принципу:

- 1) Ищет локальные минимум или локальный максимум
- 2) Отображает найденный экстремум стрелкой вниз – сигнал sell или стрелкой вверх – сигнал buy.
- 3) Следит за появлением второго экстремума подряд.

4) В случае нахождения, рисует прямоугольную область, которая отображает благоприятную ситуацию для совершения операции.

Для того, чтобы отличить сигналы buy и sell используются разные цвета, красный – сигнал для продажи, зеленый – сигнал для покупки.

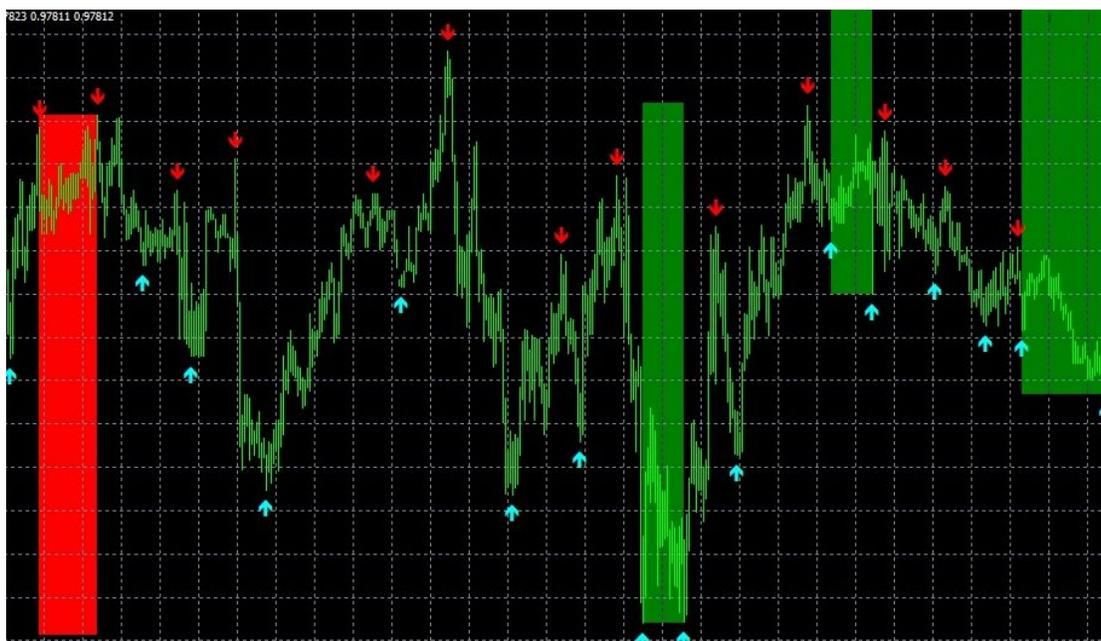


Рисунок 10 – Пример работы индикатора signals

Для улучшения результатов торговли, объединяем все три индикатора в единый фильтр. В итоге, торговые сигналы ставятся в точках совпадения всех трех индикаторов. А именно, сигнал на покупку ставится в точке совпадения зеленого квадрата, минимума индикатора Semafor и минимума DT–зигзага, с сигналами на продажу ситуация обратная. Разработанный советник позволяет визуализировать торговые ситуации в режиме онлайн и в нужный момент принимать правильные решения о направлении устанавливаемого ордера.

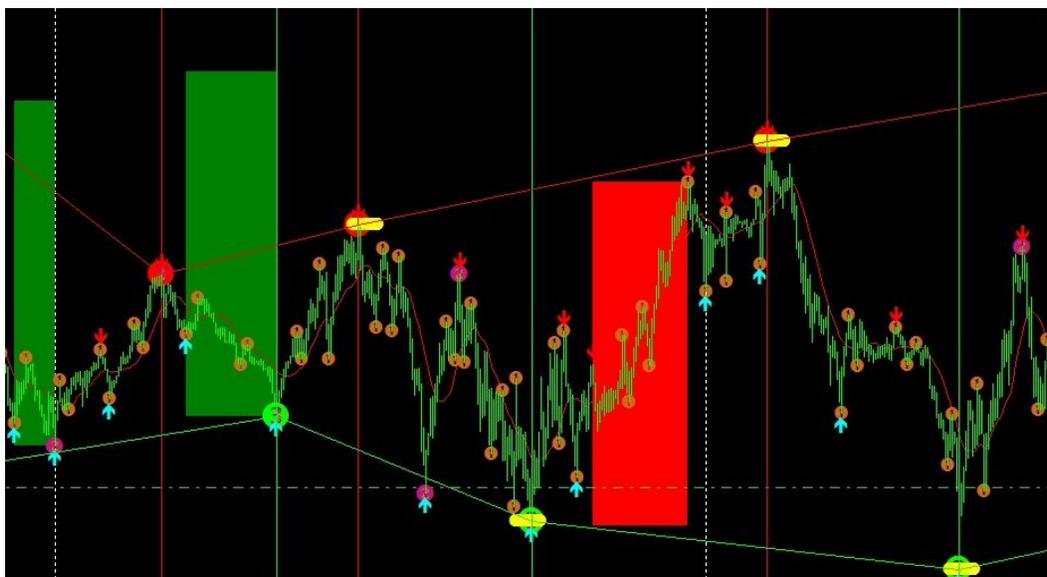


Рисунок 11 – Пример одновременной работы всех инструментов

6 ЗАКЛЮЧЕНИЕ

В представленной выпускной квалификационной работе были представлены подробности создания автоматической торговой системы для совершения операций купли/продажи валют на рынке Forex. Был проведен поиск и анализ исходных данных и поставлены задачи. Опираясь на анализ доступных технологий и сред разработки, была выбрана среда разработки и необходимые программные средства. На данной основе была разработана структура проектируемого программного продукта, а также торговый алгоритм, на базе которого происходит совершение валютных операций.

За время дипломного проектирования была изучена технология разработки автоматических торговых систем MQL5. В процессе изучения данной технологии была подана заявка на участие во Всероссийской Программе по поддержке коммерчески ориентированных научно–технических проектов молодых исследователей “УМНИК”.

Также во всех подробностях были изучены доступные методики проведения анализа на финансовых рынках, и на основе полученного опыта были выбраны наиболее подходящие для написания дипломного проекта методики. Основываясь на опыте, полученном во время изучения новых технологий разработки и методик анализа, был разработан торговый агент системы MetaTrader 5, который полностью удовлетворяет поставленной в данном дипломном проекте задаче. Дальнейшее развитие системы предполагает увеличение базового функционала, добавление новых аналитических методов и улучшение текущих модулей системы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Документация по MQL5 / Страница скачивания // URL: http://mql5.com/files/pdf/mql5_russian.pdf (дата обращения 10.04.2020).
2. Язык разработки торговых стратегий MQL5 / Страница скачивания // URL: <http://www.mql5.com/ru/docs> (дата обращения 05.04.2020).
3. Компания–разработчик программного пакета MetaTrader / Страница скачивания // URL: <http://www.metaquotes.net/ru> (дата обращения 25.02.2020).
4. Круглов, В.В. Нечеткая логика и искусственные нейронные сети: учебное пособие / В.В. Круглов, М.И. Дли, Р.Ю. Голунов. – М.: Физматлит, 2001. – 224 с.
5. Антонов, В.М. Использование обобщённых ситуаций при формировании памяти робота с обучаемой системой управления / В.М. Антонов // Теория и техника автоматического управления: сб. науч. тр. – Томск: УНПК «Кибернетика» Томского политехн. ин–та, 1990. – С. 172 – 181.
6. Вишняков, И.В. Модели и методы оценки коммерческих банков в условиях неопределенности: автореферат дис. ... д–ра экон. наук / И.В. Вишняков. – М.: Изд–во МГУ, 2002. – 34 с.
7. Тархов, Д.А. Нейрокомпьютеры и их применение. В 20 кн. Кн. 18: Нейронные сети. Модели и алгоритмы / Д.А. Тархов. – М.: Радиотехника, 2005. – 253 с.