Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Институт естественных и точных наук
Факультет математики, механики и компьютерных технологий
Кафедра прикладной математики и программирования
Направление подготовки Программная инженерия

РАБОТА ПРОВЕРЕНА
Рецензент,

_____
«____»_____2017г.

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой, д.ф.-м.н.,
доцент
_____А.А.Замышляева
«____»_____2017 г.

Разработка модуля управления капиталом для платформы MetaTrader

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ–09.03.04.2017.65.ПЗ ВКР

Руководитель работы, доцент
_____/С.М. Елсаков
«____»_____2017 г.
Автор работы
Студент группы ЕТ-484
_____/ М.П. Майоров
«____»_____2017 г.
Нормоконтролер, доцент
_____/Т.Ю.Оленчикова
«____»_____2017 г.

Челябинск 2017

АННОТАЦИЯ

Майоров М. П. Разработка модуля управления капиталом для платформы MetaTrader.– Челябинск: ЮУрГУ, ЕТ-484, 60 с., 41 ил.,14 табл., библиогр. список – 22 наим., 1 прил.

В работе исследованы существующие системы управления капиталом. Рассмотрена работа различных торговых стратегий. Исследованы модели комбинации нескольких торговых стратегий различными способами. Проведено тестирование работы на историческом периоде. Проанализированы результаты тестирования, сделаны соответствующие выводы.

Построенную комбинацию можно улучшить, путем увеличения числа моделей, входящих в рассмотрение. Полученную разработку можно использовать в качестве модуля управления капиталом для платформы MetaTrader в целях получения прибыли.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Актуальность темы дипломной работы связана со значительным распространением числа инвесторов в России. Количество уникальных пользователей на отечественной площадке ММВБ составляет порядка 780 000 человек и эта цифра увеличивается из года в год. Существует разработанная государственная стратегия, предполагающая определенный порядок развития финансовых рынков. В соответствии с этим планом к 2020 году количество торговцев валютой и ценными бумагами в нашем государстве достигнет отметки 20 млн. человек [1].

Интернет как средство связи с финансовыми рынками расширил возможности торговли, а именно увеличил скорость доступа к актуальной информации о состоянии котировок, позволил открывать сделки вне зависимости от местоположения.

Процесс принятия решения о покупке или продаже валюты связан с длительными процессами осознание потребности, поиска информации, оценки различных вариантов, реакции.

Эффективная система управления капиталом и надежная торговая платформа необходимы для постоянного роста благосостояния. Разработка таких систем автоматизации торговли через интернет, освобождает человека от рутинной работы, повышается качество проделанной работы: сокращение времени принятия решения, увеличения объемов продаж, увеличение прибыли, сокращение издержек.

1 АНАЛИЗ СУЩЕСТВУЮЩИХ МЕТОДОВ РАСПРЕДЕЛЕНИЯ КАПИТАЛА

1.1 Анализ предметной области

Для распределения капитала разработано и реализовано большое количество методов. Рассмотрим несколько наиболее распространенных методов прогноза курсов валютных пар.

Паритет покупательной способности (ППС) один из самых популярных методов. Этот подход чаще остальных упоминается в работах по экономике [2]. В основе теории ППС лежит принцип «закона одной цены», который утверждает, что стоимость идентичных товаров в разных странах должна быть одинаковой. Например, цена на автомобиль в Финляндии должна быть аналогичной цене на такой же автомобиль в Польше, принимая в учет обменный курс валют и без учета транспортных и обменных затрат. Иными словами, не должно быть повода для спекуляции, чтобы купить дешево в одной стране и продать дороже в другой.

Принцип относительной экономической стабильности позволяет спрогнозировать направление движения обменного курса. В качестве основы данного метода рассматриваются темпы роста экономики разных стран, которые дают возможность спрогнозировать динамику обменного курса. Логично предположить, что здоровый бизнес климат и стабильный экономический рост будут привлекать больше инвестиций. Для инвестирования, соответственно, необходима покупка национальной валюты, что приводит к росту спроса на национальную валюту и ее последующее укрепление. Этот метод подходит не только при сравнении состояния экономики двух стран. С его помощью можно определить наличие и интенсивность инвестиционных потоков. Например, инвесторов привлекают более высокие процентные ставки, позволяющие получить от своих инвестиций максимальную доходность. Соответственно, опять растет спрос на национальную валюту и происходит ее укрепление. В противовес низкие процентные ставки могут сократить поток иностранных инвестиций и стимулировать внутреннее кредитование. Такое положение имеет место в Японии, где процентные ставки снижены до рекордных минимумов.

Построение эконометрической модели [2]. Большой популярностью для прогнозирования курсов валют пользуется метод создания модели, описывающий связь курса обмена валюты с факторами, которые, по мнению инвестора или трейдера, влияют на ее движение. При составлении эконометрической модели, как правило, применяют величины из экономической теории, однако при расчетах могут использоваться любые другие переменные, оказывающие на обменный курс существенное влияние.

Метод анализа временных рядов является исключительно техническим и не принимает в расчет экономическую теорию. Самыми популярными моделями при анализе временных рядов является семейство моделей авторегресионного скользящего среднего (ARMA). В основе методов лежит принцип прогнозирования ценовых моделей валютной пары на основании прошлой

динамики. Расчет проводится автоматически на основе введенных параметров временного ряда, результатом которого является создание индивидуальной ценовой модели конкретной валютной пары [3].

Несомненно, прогнозирование валютных курсов, управление капиталом задачи крайне сложные. Большинство инвесторов попросту предпочитают страховать валютные риски. Другие инвесторы осознают всю важность прогнозирования валютных курсов и стремятся к изучению факторов, влияющих на них. Приведенные выше методы могут стать хорошей информационной базой именно для таких участников рынка.

1.2 Обзор платформы MetaTrader

MetaTrader это институциональная мультирыночная платформа для торговли, технического анализа, использования автоматических торговых систем (торговых роботов) и копирования торговых сделок других трейдеров. С MetaTrader вы можете торговать на валютном рынке (Форекс, Forex), фондовых биржах, а также фьючерсами (Futures) и контрактами на разницу (CFD) одновременно [4].

Торговая платформа MetaTrader является одной из самых удобных и популярных терминалов и выглядит современной.

Платформа имеет распределенную архитектуру, совершенную систему безопасности, автоматический и мобильный трейдинг. Причем это лишь некоторые отличительные характеристики MetaTrader.

Несмотря на простой интерфейс компонентов, MetaTrader имеет большой набор функций, что делает ее достаточно гибкой платформой. С ее помощью можно контролировать настройки групп, баз данных, инструментов, источников котировок.

Преимущества платформы: мультивалютность, поддержка на нескольких языках, экономичность и высокая производительность, безопасность и надежность.

Данная торговая платформа поставляются с открытыми интерфейсами (MetaTrader API). Это позволяет расширить функциональность платформы и интегрировать ее с остальными торговыми системами. Помимо этого, к услугам трейдеров предоставляются готовые плагины, которые облегчают всевозможные аспекты работы терминала MetaTrader[5].

Платформа MetaTrader обладает всем необходимым для технического анализа: тридцать встроенных и две тысячи пользовательских индикаторов; двадцать три аналитических объекта (фигуры, линии, значки), девять временных периодов, три вида представления графиков (линейные, в виде баров и японских свечей). Открыть можно сразу несколько графиков. Все они интерактивны, прокручиваются и масштабируются в режиме реального времени[4].

Для успешной аналитики в терминал встроены индикаторы и линейные инструменты, которые накладываются на график. Эти объекты помогают трейдеру проанализировать поведение валютной пары в прошлом и предсказать

возможные движения и тренды в будущем. Индикаторы накладываются на график автоматически, а линейные инструменты вручную.

Линейные инструменты – это различные геометрические фигуры для обозначения трендовых движений и различных уровней, таких как уровни поддержки и сопротивления. Они включают в себя: горизонтальные и вертикальные линии, трендовые линии, инструменты Ганна и Фибоначчи, цикличные линии, вилы Эндрюса. Индикаторы Форекс это алгоритмы, которые помогают увидеть возможные движения цены в ближайшем будущем. Индикатор дает трейдеру подсказки в виде узоров, которые указывают ему, когда лучше входить или выходить с рынка.

В MetaTrader по умолчанию добавлены стандартные индикаторы. Различают четыре основных вида индикаторов Форекс: осцилляторы, индикаторы тенденций, индикаторы объема, индикаторы Билла Вильямса. Осцилляторы определяют зоны перекупленности и перепроданности на рынке [5] и указывают момент, когда цена развернется в другую сторону.

Индикаторы тенденций показывают, в каком тренде движется цена в данный момент. Индикаторы объема указывает изменения цены в тиках. Индикаторы Билла Вильямса авторские сигналы, которые предсказывают движение цены в пяти измерениях.

В платформе встроена функция копи-трейдинга. Это вид автоматической торговли, во время которой сделки на вашем аккаунте копируются с аккаунта другого трейдера. Нет необходимости самостоятельно вести торговлю, другой трейдер все сделает за вас. За это он получает комиссию или нарабатывает рейтинг. Провайдер сигналов зарабатывает сам и помогает в этом вам [4].

Торговая платформа MetaTrader универсальна: терминал доступен как на компьютере, так и на мобильном устройстве. Существует три версии платформы: десктопная для ПК с ОС Windows, мобильная для смартфонов и планшетов с ОС Android или iOS, онлайн версия открывается из браузера с любых устройств.

Разные версии позволяют постоянно оставаться в терминале, а значит торговать 24 часа в сутки. Трейдер не привязан к месту или времени. Скачать любую из версий можно бесплатно и без регистрации.

Десктопная версия МТ самая мощная и функциональная среди всех остальных. Основные особенности платформы для ПК: торговля финансовыми инструментами рынка Форекс; 9 временных периодов на графиках; котировки в режиме реального времени; интерактивные графики; торговля в один клик; торговля с графика; возможность подключения торговых роботов (советников); более 50 индикаторов и графических паттернов для технического анализа; работа с рыночными и отложенными ордерами; новостная лента; внутренняя почта [4].

Особенности MetaTrader для устройств с iOS: торговля, не привязанная к месту; торговля финансовыми инструментами рынка Форекс; 9 временных периодов на графиках; котировки в режиме реального времени; открытие, закрытие и корректировка рыночных и отложенных ордеров; торговля с графика; 30 технических индикаторов; панель с отображением открытых и закрытых сделок, торговой истории, оповещений, журнала и новостей; интерактивные

графики, которые можно прокручивать, увеличивать, уменьшать, настраивать цвета баров, объемы, сетку и т.д.; оффлайн-режим, в котором доступны котировки, графики, торговая история [4].

Если у вас смартфон или планшет с Android, то вы получите торговую программу с множеством функций и простым дизайном. Плюсы платформы для Android: понятный интерфейс; торговля на рынке Форекс в режиме 24/5; графики с девятью таймфреймами; котировки в режиме реального времени; полноценная работа с рыночными и отложенными ордерами; возможность торговли с графика; 30 индикаторов для технического анализа; отображение открытых и закрытых сделок, торговой истории, журнала, новостей; возможность увеличивать, уменьшать, прокручивать и настраивать графики; оффлайн-режим с котировками, графиками, торговой историей.

Если вам не удается скачать торговый терминал MT или вы не хотите этого делать, воспользуйтесь вебтерминалом. Вебтерминал – это онлайн версия платформы MT, которая работает через браузер. Функционал у браузерной версии упрощен. Чтобы работать в MT WebTrader нужно устройство с доступом в Интернет. Особенности браузерного терминала: не нужно скачивать и устанавливать терминал; доступен на любом устройстве с любой ОС с браузером и выходом в Интернет; торговля финансовыми инструментами рынка Форекс; 9 временных периодов на графиках; котировки в режиме реального времени; открытие, закрытие и корректировка рыночных и отложенных ордеров; торговля с графика; торговля в один клик; 30 технических индикаторов; отображение полной торговой истории; интерактивные графики, которые можно прокручивать, увеличивать, уменьшать, настраивать цвета баров, объемы, сетку и т.д.; оффлайн-режим, в котором доступны котировки, графики, торговая история; торговля в полноэкранном режиме [6].

Торговый терминал MetaTrader позволяет не только писать советники, но и тестировать их перед использованием. Эта полезная функция позволяет проверить работоспособность и эффективность торгового робота на исторических данных. Успешная торговля на рынке Форекс не возможна без использования торговой стратегии. Кто-то применяет механическую торговую систему, полностью полагаясь на ее сигналы, а кто-то применяет более свободный стиль торговли, опираясь еще и на свою интуицию, опыт, собственное видение рынка (дискретный трейдинг). Обычно используется некий базовый набор правил, который составляет основу торговли трейдера, и на которые он опирается главным образом при принятии решений [5]. Этот набор правил, или торговая стратегия, требует на начальном этапе обязательной проверки, тестирования на достаточно продолжительном промежутке времени, чтобы убедиться в эффективности ее использования.

Одним из самых известных способов проверки стратегии является торговля на демо-счете. Но, к сожалению, этот способ не пользуется популярностью, поскольку, во-первых, необходимо много времени для тестирования, а во-вторых, считается, что торговля на демо-счете не может дать той полноты ощущений, которая присутствует при торговле на реальные средства. В общем, как

показывает практика, демо-счет используется неохотно. Тестирование дает возможность приступить к автотрейдингу, зная об особенностях поведения советника в различных рыночных ситуациях. Для этих целей в торговый терминал встроено специальное средство «Тестер стратегий». К параметрам тестирования относятся объем и валюта начального депозита, которые указываются в соответствующих полях. Именно этим депозитом будет оперировать советник при тестировании. В этой вкладке также выбираются типы открываемых позиций при тестировании: Only Long открывать только длинные позиции; Only Short только короткие; Long and Short открывать позиции в обе стороны. Каков бы ни был алгоритм торгового эксперта, он будет открывать позиции только в заданных направлениях. Также можно включить генетический алгоритм тестирования [4].

1.3 Обзор методов управления капиталом

На тему максимизации дохода написано много статей [6]. Они помогли поднять науку управления капиталом на новый уровень. Однако многие книги и статьи по этой теме поверхностно обсуждают риск [7] и психологические особенности максимизации дохода по счету. Во многих случаях максимизация дохода по счету подвергает трейдеров большему риску разорения или, как минимум, большим потерям. Многие люди не готовы воспринимать 50% уменьшение своего счета, даже если это только уменьшение ранее полученной прибыли. Для финансовых менеджеров 30% падение может быть началом крушения карьеры по той простой причине, что институциональные и индивидуальные инвесторы, доверившие ему управление своимим средствами, готовы с пониманием отнестись к уменьшению счета только в пределах 15% [6].

Управление капиталом может применяться всеми трейдерами. Но для эффективного использования методов управления капиталом трейдер должен иметь четкие правила открытия и закрытия позиций, а также результаты тестирования данных правил на истории цен – прибыли и убытки. Хотя некоторые трейдеры не используют определенной системы в торговле, на сегодняшний день у большинства трейдеров есть определенная система или подход, применяемые в торговле. Обладание правильно протестированной системой дает уверенность в том, что трейдер будет получать, работая по ней, стабильные, в пределах статистистических границ, результаты. Именно внутри этих статистических границ и возможно эффективное и квалифицированное управления счетом. Основным способом, с помощью которого трейдеры учатся определять эти статистические границы, является большой набор протестированных сделок [7].

Методы управления капиталом устанавливают долю капитала в конкретной торговой операции и обуславливают рост производительности торговой системы. Относительно эффективности методов управления капиталом участники Форекс дают разную оценку. Существуют сторонники механических торговых систем,

которые полагаются на технический анализ, а также люди, отдающие предпочтение фундаментальному анализу.

Отсутствие управления капиталом [7]. При открытии позиции многие участники рынка Форекс не занимаются расчетами объемов средств, используемых при совершении сделок, размера ожидаемой прибыли и убытков. Данная стратегия существует и применяется довольно часто. Но при данном варианте высока вероятность потери капитала при небольшом его объеме в результате нескольких неудачных операций [8].

Торговля всем капиталом. Данный метод или стратегия является наиболее простым и пользующимся популярностью среди трейдеров. Смысл состоит в том, что участник рынка Форекс при поступлении сигнала на вход от системы открывает максимальную позицию, то есть на все имеющиеся средства. Положительным моментом данной тактики является возможность получать максимальную прибыль при наличном депозите. Но при всем при этом данный метод сопровождается высокими рисками. Данная тактика может оправдать себя лишь на торговых операциях с высоким математическим ожиданием [6].

Торговля фиксированной суммой [7]. Данный метод еще называют «предел суммы» или «лимитирование суммы». С учетом состояния своего счета участник валютного рынка при создании очередной позиции самостоятельно определяет уровень средств, которыми он может рисковать. Все торговые операции осуществляются с учетом границ ставки или запланированного лимита.

Торговля фиксированным количеством контрактов. Данный метод разрешает открытие позиций количеством лотов, которое запланировано заранее. Основной целью данного метода является снижение рисков. Независимо от эффективности системы, участник валютного рынка Форекс вправе держать в обороте постоянный размер денежных средств.

Торговля фиксированным процентом капитала. При использовании данного метода трейдер определяет лимит уже на процент от объема средств. Все остальное схоже с торговлей фиксированной суммой. Величина открываемых позиций изменяется пропорционально размеру счета. В случае небольшого размера начального депозита торговля фиксированным процентом капитала невозможна.

Торговля фиксированной пропорцией (метод Джонса) [7]. Данный метод в свое время был предложен известным экономистом Райаном Джонсом. По его мнению, применяя данную тактику, можно избавиться от недостатков предыдущих методов. Участник рынка определяет некую переменную величину дельту, при достижении которой повышается количество контрактов, а при уменьшении суммы на счете в размере величины дельта, число контрактов уменьшается. Эта тактика дает возможность одновременно осуществлять контроль, как за рисками, так и за рентабельностью [8].

«Пирамида». Трейдер в зависимости от выполнения тех или иных условий вправе осуществлять последовательное наращивание объемов сделки, открытой первоначально, посредством прибавления заранее установленной доли капитала. Данный метод основан на откровенной спекуляции. Вообще данный вид

стратегии является очень рискованным, поскольку строится не с основания, а с вершины. Не следует начинать строить пирамиду с убыточных позиций [9].

Мартингальный и антимартингальный методы. Мартингальный метод предполагает повышение размера позиции при уменьшающемся капитале. Антимартингальный, соответственно, наоборот, предполагает увеличение размера позиции при увеличении капитала. Предварительно установленные количество лотов или процент от объема средств, берутся за единицу увеличения или уменьшения позиции. Эти методы являются разновидностью строительства пирамиды, только условиями изменения размера позиций являются финансовые результаты предыдущих сделок, а не поведение цены или показатели технического анализа. Метод скользящей средней.

Для большинства данный метод известен как сигнал на вхождение в рынок или выход из него. Основой данного метода является анализ кривой доходности торговой системы. Скользящие средние линии (длинная и короткая) используются для определения результатов совершенных торговых операций. В случае нахождения короткой линии выше длинной рекомендуется открывать позиции, если же трейдер получает сигнал из торговой системы о нахождении короткой средней линии ниже длинной, то стоит подождать наиболее подходящего момента [10].

Метод безопасной и оптимальной доли. Оптимальная доля представляет собой специально высчитанную переменную процента капитала, который используется при каждой совершаемой сделке, при которой данная тактика выбранная тактика давала наибольший доход. Безопасная доля выражается в индикаторе ограничения предельно допустимых убытков. Выбор обоих тактик может быть оправдано лишь при постоянстве показателей торговой системы [11].

Серийный тест. Когда в случае с колодой карт мы проводим отбор без замещения, можно путем проверки определить, существует ли зависимость. Для определенных событий (таких, как поток прибыли и убытков по сделкам), где зависимость не может быть определена путем проверки, мы будем использовать серийный тест. Серийный тест подскажет нам, имеет ли наша система больше (или меньше) периодов последовательных выигрышей и проигрышей, чем случайное распределение. Цель серийного теста найти счет Z для периодов выигрышей и проигрышей в системной торговле. Счет Z означает, на сколько стандартных отклонений вы удалены от среднего значения распределения. Таким образом, счет Z = 2,00 означает, что вы на 2,00 стандартных отклонения удалились от среднего значения (ожидание случайного распределения периодов выигрышей и проигрышей). Счет Z это просто число стандартных отклонений, на которое данные отстоят от среднего значения нормального распределения вероятности. Например, счет Z в 1,00 означает, что данные, которые вы тестируете, отклонены на 1 стандартное отклонение от среднего значения. Счет Z затем переводится в доверительную границу, которая иногда также называется степенью достоверности. Связь счета Z и доверительной границы следующая: счет Z является числом стандартных отклонений от среднего значения, а

доверительная граница является долей площади под кривой, заполненной при таком числе стандартных отклонений [16].

1.4 Обзор языка программирования MQL

MetaQuotes Language (MQL) язык программирования технических индикаторов, торговых роботов и вспомогательных приложений для автоматизации торговли на финансовых рынках. MQL является современным языком высокого уровня и разработан MetaQuotes Software Corp. для собственной торгово-информационной платформы. Синтаксис языка максимально близок к C++ и позволяет писать программы в стиле объектно-ориентированного программирования (ООП) [17].

В состав MQL включено большое количество функций, необходимых для анализа текущих и пришедших ранее котировок, встроены основные индикаторы и функции по управлению торговыми ордерами и контролю над ними. Для написания программ на MQL в составе торговой платформы предоставляется среда разработки MetaEditor со всеми современными инструментами для написания кода, включающими в себя шаблоны, сниппеты, отладку, профилировку, автозавершение и встроенное версионное хранилище MQL Storage. Для написания кода программы используется текстовый редактор экспертов MetaEditor, выделяющий цветом различные конструкции языка MQL, что позволяет пользователю лучше ориентироваться в тексте экспертной системы.

Язык MQL является объектно-ориентированным языком программирования высокого уровня и предназначен для написания автоматических торговых стратегий. Он позволяет не только писать разнообразные экспертные системы, предназначенные для работы в режиме реального времени, но и создавать собственные графические инструменты, помогающие принимать торговые решения. В языке MQL присутствуют перечисления, структуры, классы и обработка событий. Благодаря расширению числа встроенных основных типов языка C++, взаимодействие исполняемых программ на MQL с другими приложениями посредством dll максимально облегчено. Синтаксис языка MQL подобен синтаксису C++, и это позволяет легко переносить на него программы из современных языков программирования. Для облегчения написания программ, а также для удобства восприятия исходных текстов программ, в языке MQL предусмотрены предопределенные стандартные константы и перечисления. Кроме того, для хранения информации используются служебные структуры. Стандартные константы являются аналогом макроподстановок и имеют целочисленный тип integer. Синтаксически язык программирования торговых стратегий MQL очень похож на язык программирования C++, за исключением некоторых особенностей: отсутствует адресная арифметика; отсутствует оператор goto; нельзя объявить анонимное перечисление; нет множественного наследования [17].

Поддержка и развитие языка осуществляется на сайте MQL5.community, где находится обширная библиотека пользовательских кодов и множество статей. Эти

статьи охватывают все темы современного трейдинга: нейронные сети [13], статистика и анализ, высокочастотная торговля, арбитраж, тестирование и оптимизация торговых стратегий, использование роботов для автоматизации торговли и многое другое.

Каждый скрипт и каждый эксперт работает в собственном отдельном потоке. Все индикаторы, рассчитываемые на одном символе, даже если они запущены на разных графиках, работают в одном потоке. Таким образом, все индикаторы на одном символе делят между собой ресурсы одного потока [18].

В одном потоке с индикаторами также последовательно выполняются остальные действия по данному символу - обработка тиков и синхронизация истории. Это означает, что если в индикаторе выполняется бесконечное действие, все остальные события по его символу никогда не выполнятся.

При запуске эксперта ему необходимо обеспечить актуальное торговое окружение, доступность истории по данному символу и периоду, а также произвести синхронизацию между терминалом и сервером. На эти процедуры терминал предоставляет эксперту отсрочку запуска не более чем в 5 секунд, по истечении которых эксперт будет запущен с теми данными, что удалось подготовить. Поэтому при отсутствии связи с сервером – это может привести к задержке запуска эксперта.

Сразу после присоединения программы к графику производится ее загрузка в память клиентского терминала и инициализация глобальных переменных. Если какая-либо глобальная переменная типа класса имеет конструктор, то этот конструктор будет вызван в процессе инициализации глобальных переменных.

После этого программа находится в состоянии ожидания события от клиентского терминала. У каждой mql-программы должна быть хотя бы одна функция-обработчик события, в противном случае загруженная программа выполняться не будет. Функции-обработчики событий имеют предопределенные имена, предопределенные наборы параметров и предопределенные типы возврата.

Клиентский терминал отсылает возникающие события в соответствующие открытые графики. Также события могут генерироваться графиками (события графика) либо mql-программами (пользовательские события). Генерацию событий создания и удаления графических объектов на графике можно включать и отключать заданием свойств графика. Каждая mql-программа и каждый график имеют свою собственную очередь событий, куда складываются все вновь поступающие события. Программа получает события только от графика, на котором она запущена. Все события обрабатываются одно за другим в порядке поступления. Если в очереди уже есть событие NewTick либо это событие находится в состоянии обработки, то новое событие NewTick в очередь mql5-программы не ставится.

В языке MQL существует специальная группа торговых функций, с помощью которых можно создавать автоматизированные торговые системы. Программы для автоматической торговли без участия человека называются экспертами (Expert Advisor) или торговыми роботами. Для создания эксперта в редакторе MetaEditor имеется возможность использовать мастер создания советников MQL

Wizard. Expert Advisor(template) позволяет создать шаблон с готовыми функциями обработки событий, который необходимо дополнить всем необходимым функционалом с помощью самостоятельного программирования. Expert Advisor(generate) дает возможность создать полностью готового для работы торгового робота, просто выбирая необходимые вам модули: модуль формирования торговых сигналов, модуль управления капиталом и модуль подтягивания защитного стопа для открытых позиций (Trailing Stop).

Сразу же после того, как клиентский терминал загрузит программу (эксперт или пользовательский индикатор) и запустит процесс инициализации глобальных переменных, будет послано событие Init, которое обрабатывается функцией OnInit(), если она есть. Это событие также генерируется после смены финансового инструмента и/или периода графика, после перекомпиляции программы в редакторе MetaEditor, после смены входных параметров из окна настройки эксперта или пользовательского индикатора. Советник также инициализируется после смены счета. Для скриптов событие Init не генерируется.

Перед деинициализацией глобальных переменных и выгрузкой программы (эксперт или пользовательский индикатор) клиентский терминал посылает программе событие Deinit. Событие Deinit также генерируется при завершении работы клиентского терминала, при закрытии графика, непосредственно перед сменой финансового инструмента и/или периода графика, при удачной перекомпиляции программы, при смене входных параметров, а также при смене счета.

Событие Start это специальное событие для активизации скрипта после его загрузки. Это событие обрабатывается функцией OnStart. Событие Start экспертам и пользовательским индикаторам не посылается.

Событие NewTick генерируется при поступлении новых котировок и обрабатывается функцией OnTick() у присоединенных советников. Если при поступлении новой котировки выполнялась функция OnTick, запущенная на предыдущей котировке, то пришедшая котировка будет проигнорирована советником, так как соответствующее событие не будет поставлено в очередь событий эксперта. Все пришедшие во время выполнения программы новые котировки программой игнорируются до тех пор, пока не завершится очередное выполнение функции OnTick(). После этого функция будет запущена только после прихода очередной новой котировки. Событие NewTick генерируется независимо от того, запрещена или разрешена автоматическая торговля. Запрет автоматической торговли означает только запрет на отправку торговых запросов из эксперта, работа эксперта не прекращается. Запрет автоматической торговли путем нажатия на указанную кнопку не прерывает текущее выполнение функции OnTick().

Событие Trade генерируется при завершении торговой операции на торговом сервере. Обработка события Trade производится функцией OnTrade() для следующих торговых операций [17].

Для импорта функций во время выполнения mql-программы используется раннее связывание. Это значит, что если в программе есть вызов импортируемой

функции, то соответствующий модуль (ex или dll) загружается в процессе загрузки программы. Библиотеки MQL и DLL выполняются в потоке вызывающего модуля. Системные библиотеки (DLL) загружаются по правилам операционной системы. Если библиотека уже загружена (например, другим экспертом и даже из другого клиентского терминала, запущенного параллельно), то обращение идет к уже загруженной библиотеке. Перед загрузкой эксперта (скрипта, индикатора) формируется общий список всех библиотечных модулей, которые предполагается использовать как из загружаемого эксперта (скрипта, индикатора), так и из библиотек из этого списка. Таким образом обеспечивается однократная загрузка многократно используемых библиотечных модулей. Функции, импортируемые из DLL в mql-программу, должны обеспечивать соглашение о связях, принятое для функций Windows API. Для обеспечения такого соглашения в исходном тексте программ, написанных на языках C или C++, используется ключевое слово __stdcall, которое является специфическим для компиляторов от фирмы Microsoft. В случае если соответствующая библиотека не смогла загрузиться, либо установлен запрет на использование DLL, либо импортируемая функция не была найдена – эксперт останавливает свою работу с соответствующим сообщением "expert stopped" в журнале. При этом эксперт не будет запускаться, пока не будет заново проинициализирован. Эксперт может быть переинициализирован в результате перекомпиляции либо после открытия таблицы свойств эксперта и нажатия кнопки OK.

1.5 Постановка задачи исследований

Целью работы является разработка программного модуля распределения капитала в MetaTrader.

Задачи для того, чтобы построить модуль, необходимо:

1) изучить существующие методы управления капиталом;
2) изучить язык программирования торговых стратегий MQL;
3) разработать алгоритм, управления капиталом;
4) представить полученные данные в графическом виде.

Полученные результаты в дальнейшем можно применить для получения дополнительного источника дохода, увеличения скорости принятия решений, повышения качества работы.

1.6 Выводы по разделу

С распространением интернета и увеличения вычислительной мощности компьютеров стало возможным автоматизировать многие процессы, в том числе и управление капиталом. Существенным преимуществом автоматизированных торговых систем является скорость принятия решений, бесперебойная работа круглые сутки без перерыва, экономичность обслуживания. В данной работе рассматривается модуль управления капиталом разработанный для платформы MetaTrader.

2 РАЗРАБОТКА АЛГОРИТМА УПРАВЛЕНИЯ КАПИТАЛОМ

2.1 Алгоритм «Критерий К»

Переключение на данную модель осуществляется тогда, когда К ее последних прогнозов являются наилучшими в сравнении с прогнозами по другим моделям, входящим в базовый набор адаптивной комбинированной модели (АКМ).

Один из способов определения наилучшего сигнала входящих в АКМ Критерий К (рисунок 2.1) [19].

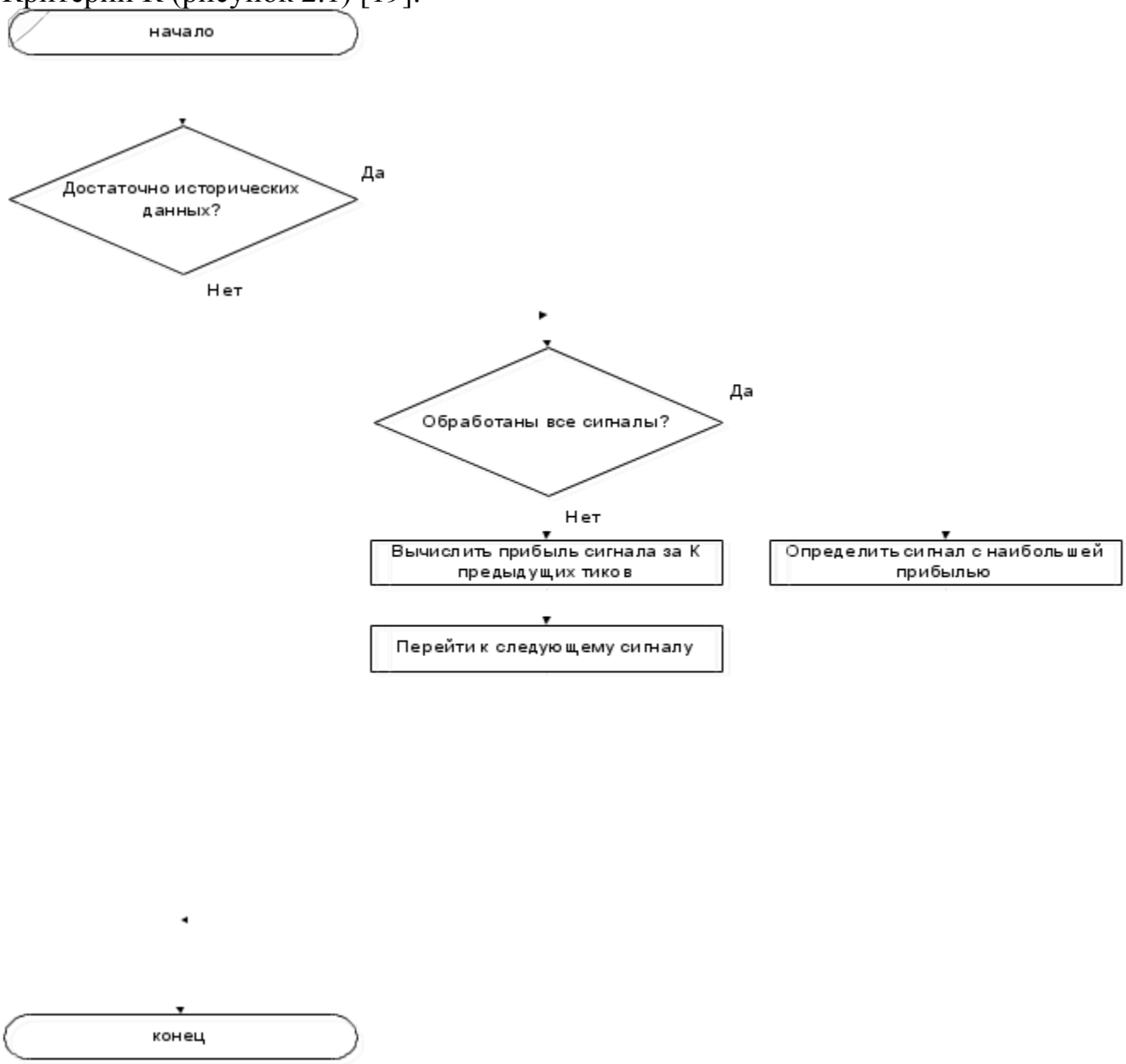

Рисунок 2.1 – Схема вспомогательного алгоритма определения наилучшего сигнала Критерий К

Определение сигнала с наибольшей прибылью осуществляется с помощью вспомогательной функции FindBestSignal, которая возвращает индекс сигнала. Согласна Критерию К предпочтительным в данный момент времени (на текущем

тике) является работа по информации, полученной от этого сигнала. Далее после определения наилучшего сигнала возвращаем его индекс в вызывающую подпрограмму, чтобы затем совершить торговую операцию (рисунок 2.2).

```
int FindBestSignal()
{
   for(int i=0; i<SIGNALS_N;i++)
   {
      profits[i] = 0;
      for(int j = 1; j<= K;j++)
        {
         int oldTick=(N+currentTick-j)%N;
         if(predicts[i][oldTick]==BUY)
           {//Советник просигнализировал о покупке.
            //Купили по цене asks[i]. Продаем по цене last_tick.bid.
            profits[i]+=last_tick.bid-asks[oldTick];
           }
         if(predicts[i][oldTick]==SELL)
           {//Советник просигнализировал о продаже.
            //Продали по цене bids[i]. Покупаем по цене last_tick.ask.
            profits[i]+=last_tick.ask-bids[oldTick];
           }
        }
   }
   int bestSignal=0;
   for(int i=0; i<SIGNALS_N;i++)
      if(profits[i]>profits[bestSignal])
         bestSignal=i;
   return bestSignal;
}
```

Рисунок 2.2 – Листинг функции поиск наилучшего сигнала

После того, как будет выяснено какой сигнал является предпочтительным на текущем тике, происходит передача управления в вызывающую подпрограмму, фрагмент кода представлен ниже (рисунок 2.3).

```
int RulesK()
  {
   if(currentTick<=K)//Не достаточно истории.
      return -1;

   int bestSignal = FindBestSignal();

   return bestSignal;
  }
```

Рисунок 2.3 – Листинг функции Критерий К

2.2 Алгоритм «Критерий В»

Переключение на данную модель осуществляется тогда, когда ее экспоненциально сглаженный квадрат ошибки прогнозирования В минимален по сравнению с аналогичным показателем для остальных моделей в базовом наборе АКМ.

Критерий В формируется следующим образом:

$$B_t=\left(1-\alpha_B\right)B_{t-1}+\alpha_B e_\tau^2(t-\tau) \quad , \qquad (2.1)$$

где α_B – параметр сглаживания; $e_\tau(t-\tau)$ – ошибка прогноза, сделанного в момент $t-\tau$ на τ шагов вперед.

Параметр B характеризует инерционность переключения, промедление с переключением может привести к плохой работе АКМ, нарушить соответствие структуры модели динамике процесса.

Один из способов определения наилучшего сигнала входящих в АКМ Критерий В [2] (рисунок 2.4).

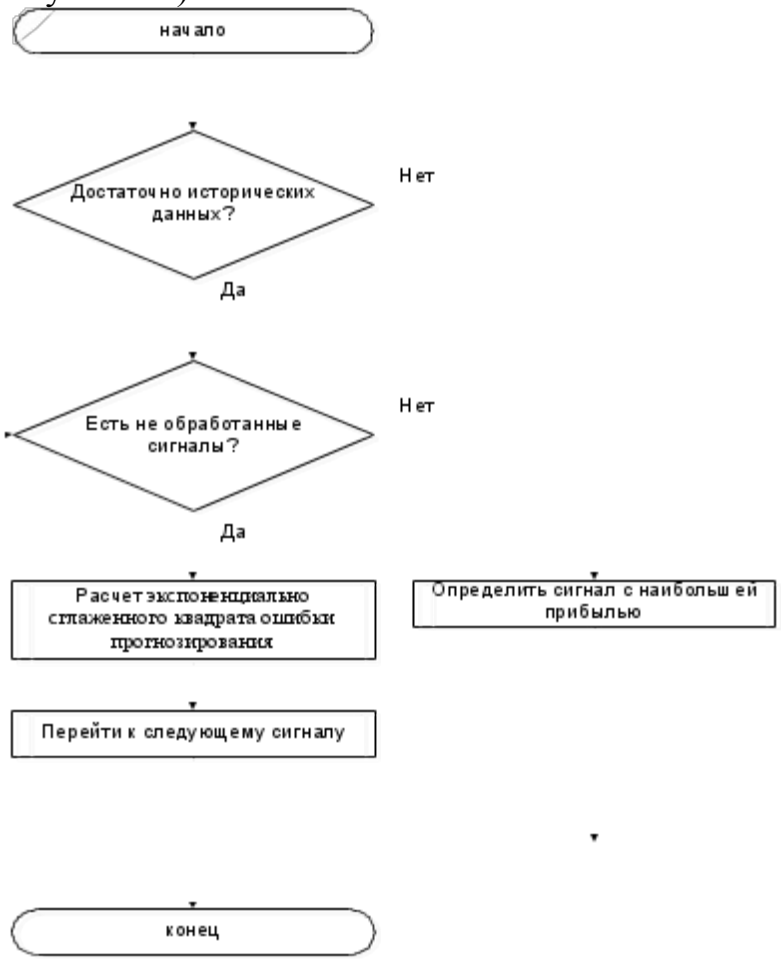

Рисунок 2.4 – Схема вспомогательного алгоритма определения наилучшего сигнала Критерий В

Пример реализации работы алгоритма определения наилучшего сигнала по Критерию В представлен ниже (рисунок 2.5).

```
int RulesB()
  {
   if(currentTick <= tau)//Не достаточно истории.
      return -1;
   for(int i = 0;i < SIGNALS_N; i++)
   {
      double e = 0;
      if(predicts[i][currentTick - tau]==BUY)
           {//Советник просигнализировал о покупке.
            //Купили по цене asks[i]. Продаем по цене last_tick.bid. Разницу в карман.
            e = last_tick.bid-asks[currentTick - tau];
           }
      if(predicts[i][currentTick - tau]==SELL)
           {//Советник просигнализировал о продаже.
            //Продали по цене bids[i]. Покупаем по цене last_tick.ask.
            e = last_tick.ask-bids[currentTick - tau];
           }
      if(e > 0)
         e = 0;

      B[i] = (1 - alfa) * B[i] + alfa * e * e;
   }
   int bestSignal=0;
   for(int i=0; i<SIGNALS_N;i++)
      if(B[i] < B[bestSignal])
         bestSignal=i;
   return bestSignal;
  }
```

Рисунок 2.5 – Листинг функции Критерий B

2.3 Алгоритм «Голосование Большинства»

Определение наилучшего сигнала по правилу голосования большинства осуществляется следующим образом, собирается информация о всех моделях, входящих в АКМ, далее идет подсчет голосов за покупку, продажу либо воздержание совершения операций. После того, как проведен подсчет и определен один из сигналов, голосовавших за операцию победителя, происходит возврат в подпрограмму (рисунок 2.6).

Рисунок 2.6 – Схема вспомогательного алгоритма определения наилучшего сигнала Голосование Большинства

Пример реализации алгоритма Голосования большинства представлен ниже (рисунок 2.7).

```
92 int MajorityVote()
93 {
94    int cntSell = 0, cntBuy = 0, cntWait = 0;
95    int signalSell = -1, signalBuy = -1, signalWait = -1;
96    for(int i=0; i<SIGNALS_N;i++)
97    {
98       if(predicts[i][(N+currentTick-1)%N] == BUY)
99       {
100         cntBuy++;
101         signalBuy = i;
102      }else if(predicts[i][(N+currentTick-1)%N] == SELL)
103      {
104         cntSell++;
105         signalSell = i;
106      }
107      else
108      {
109         cntWait++;
110         signalWait = i;
111      }
112   }
113   if(cntSell > cntBuy)
114      return signalSell;
115   if(cntBuy > cntSell )
116      return signalBuy;
117
118   return signalWait;
119 }
```

Рисунок 2.7 – Листинг Голосование Большинства

2.4 Основной алгоритм работы модуля

Особенностью программ, предназначенных для работы в клиентском терминале MetaTrader, является их работа с постоянно обновляющейся информацией в режиме реального времени. В языке MQL существуют специальные функции: NewTick(), Init(), DeInit() и др. (рисунок 2.8).

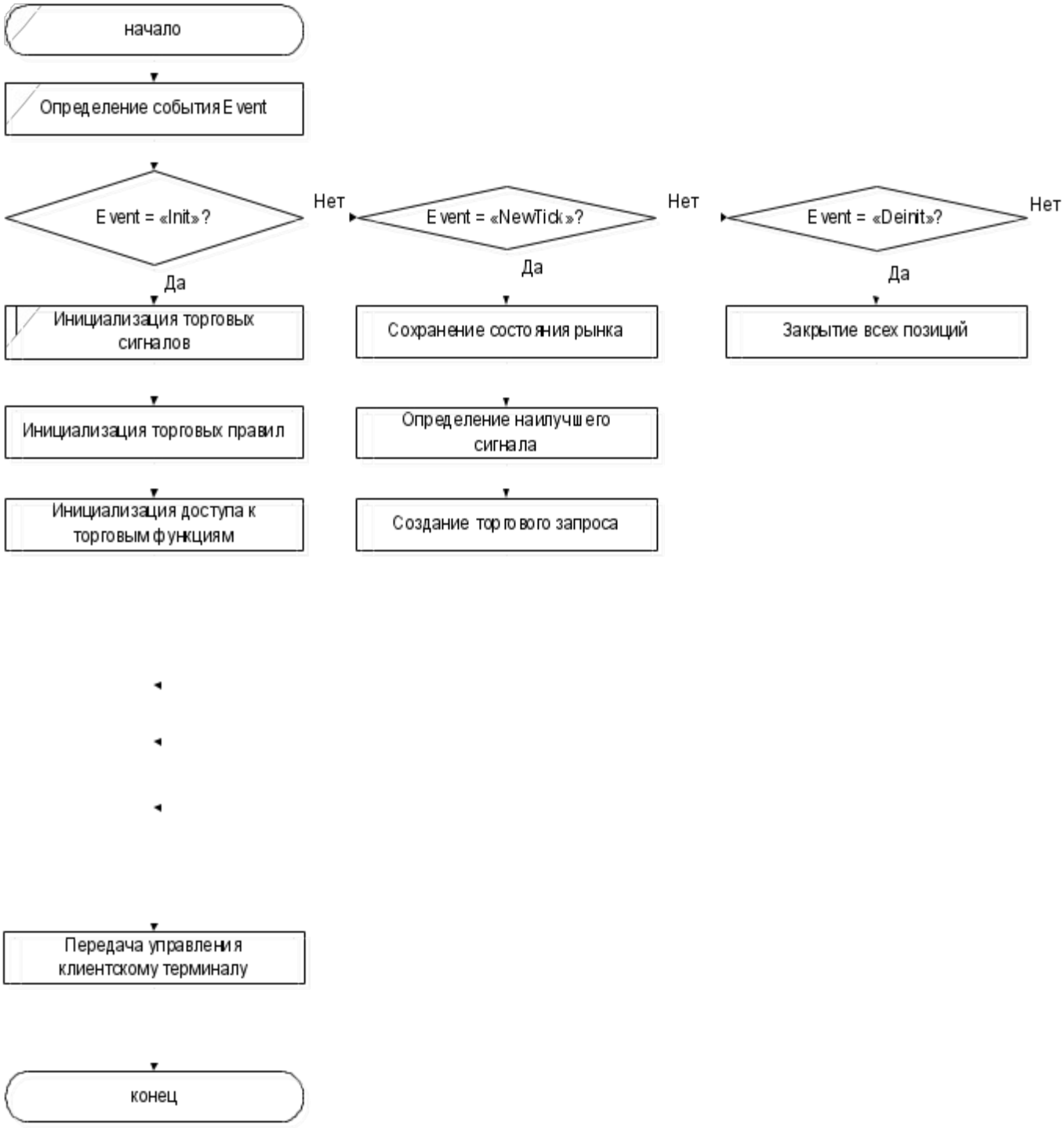

Рисунок 2.8 – Схема основного алгоритма программы

Функция Init() вызывается автоматически клиентским терминалом MetaTrader при запуске программы. Программист должен определить действия, которые необходимы для дальнейшей работы торговой стратегии. Например, инициализировать торговые сигналы, далее торговые правила и функции для работы с торговыми запросами. (рисунок 2.9)

```
48 int OnInit()
49   {
50    InitSignals();
51
52    InitRulesB();
53
54    InitTrade();
55    Print("Init success");
56    return(INIT_SUCCEEDED);
57   }
58
```

Рисунок 2.9 – Листинг инициализации

Функции InitSignals, InitRules, InitTrade являются заранее определенными программистом.

Для увеличения числа сигналов, входящих в базовый набор адаптивной комбинированной модели (АКМ) достаточно увеличить константу SIGNALS определенную в начале программы следующим образом:

#define SIGNALS_N 11

Далее в блоке инициализации торговых сигналов добавить самостоятельно разработанный сигнал, следует обратить внимание, что нумерация начинается с нуля. Пример добавления собственного сигнала представлен ниже.

```
44 #define SIGNALS_N 12
45
46 void InitSignals()
47  {
48    signals[0] = (new MovingAverageIntersect());
49    signals[1] = (new MacdIntersect());
50    signals[2] = (new Stochastic());
51    signals[3] = (new AMA());
52    signals[4] = (new RSI());
53    signals[5] = (new CCI());
54    signals[6] = (new WPR());
55    signals[7] = (new BANDS());
56    signals[8] = (new StdDevChannel());
57    signals[9] = (new Envelopes());
58    signals[10] = (new Alligator());
59
60    signals[11] = (new MyCustomSignal());
61  }
```

Рисунок 2.10 – Листинг инициализация сигналов

Для управления созданием торгового запроса имеется возможность изменить функцию InitTrade, в которой определены величина проскальзывания, а также режим исполнения торговых операций.

```
void InitTrade()
  {
//--- установим допустимое проскальзывание
   int deviation=10;
   trade.SetDeviationInPoints(deviation);
//--- какую функцию использовать для торговли: true - OrderSendAsync()
   trade.SetAsyncMode(true);
//---
  }
```

Рисунок 2.11 – Листинг инициализация торговых функций

Событие NewTick создается для экспертов при возникновении нового тика по символу, с графиком которого работает модуль. Тело функции OnTick имеет следующее содержание:

```
299 void OnTick()
300   {
301    RememberState();
302    int signal=RulesB();
303    MakeOrder(signal);
304   }
```

Рисунок 2.12 – Листинг обработка события NewTick

Сперва вызывается подпрограмма, сохраняющая информацию о котировках на текущем тике. Сохраняется информация о цене покупки и цене продажи выбранного символа. Далее вычисляется и сохраняется информация по каждому сигналу, возможны три возвращаемых значения от сигнала – это BUY, SELL и WAIT (покупать, продавать и ждать).

```
81 void RememberState()
82   {
83    SymbolInfoTick(Symbol(),last_tick);//Получение информации о текущем тике.
84    asks[currentTick % N] = last_tick.ask;
85    bids[currentTick % N] = last_tick.bid;
86    for(int i=0; i<SIGNALS_N; i++)
87      {
88       OrderType predict=signals[i].calc();
89
90       predicts[i][currentTick]=predict;
91      }
92    currentTick = (currentTick + 1)%N;
93   }
```

Рисунок 2.13 – Листинг сохранение состояния

После того, как было сохранено состояние котировок и получена информация от сигналов, определяется модель, входящая в базовый набор АКМ, которая по заданному алгоритму определяет наилучший сигнал, по которому в дальнейшем будет сделан торговый запрос. В данной работе рассмотрены три алгоритма определения наилучшего сигнала Критерий К, Критерий В и Голосование

Большинства, которые более детально рассмотрены в разделах 2.1, 2.2 и 2.3 соответственно.

Получив информацию о том, какой сигнал на данном тике является наиболее выгодным совершается торговая сделка с брокером.

```
306 if(!trade.Buy(0.01, NULL, 0.0, last_tick.ask - 0.01, last_tick.ask + 0.05))
307   {
308   //--- сообщим о неудаче
309     Print("Метод Buy() потерпел неудачу. Код возврата=",trade.ResultRetcode());
310   }
311   else
312   {
313     openPos++;
314     lastOrder=BUY;
315   }
```

Рисунок 2.14 – Листинг торговой сделки

Рассмотрим детальнее метод торгового терминала Buy.

```
bool  Buy(
   double         volume,          // объем позиции
   const string   symbol=NULL,     // символ
   double         price=0.0,       // цена исполнения
   double         sl=0.0,          // цена Stop Loss
   double         tp=0.0,          // цена Take Profit
   const string   comment=""       // комментарий
   )
```

Рисунок 2.15 – Листинг сигнатуры Buy

Успешное окончание работы метода Buy(...) возвращает true, в случае возникновения ошибки – false и устанавливается ее код. Для получения кода ошибки можно воспользоваться функцией получения результата последнего запроса trade.ResultRetcode().

Функция OnDeinit () вызывается при деинициализации программы. Ее главная задача завершить работу модуля и избежать сохранения открытых позиций. Закрытие позиций необходимо, чтобы избежать потерю контроля за капиталом. После того, как клиентский терминал сгенерирует событие Deinit, происходит освобождение всех ресурсов, использованных в модуле.

```
288 void OnDeinit(const int reason)
289   {
290    CloseAllPosition();
291   }
292
293 void CloseAllPosition()
294    {
295       while(trade.PositionClose(_Symbol));
296    }
```

Рисунок 2.16 – Листинг деинициализация

Так как количество открытых позиций ограничивается только капиталом, то их число может быть очень велико. Перед окончательным завершением работы

модуля следует закрыть все открытые позиции. Метод PositionClose(…) возвращает истину при успешном закрытии последней активной позиции, в противном случае означает, что открытых позиций больше нет, либо произошла ошибка, код которой доступен через запрос trade.ResultRetcode().

Во время работы модуля управления капиталом возможна работа в ручном режиме, иными словами существует возможность самостоятельно открывать длинные и короткие позиции, устанавливать величину stop loss и take profit, а также многое другое. Все сделки, которые совершены вне модуля, не будут влиять на выбор сигнала, но если наилучший сигнал входящий в набор АКМ просигнализирует о покупке, а пользователь совершил продажу, то сделка пользователя будет закрыта, так как она противоречит стратегии.

2.5 Индикаторы, входящие в состав адаптивной модели

В базовый состав АКМ входит 11 различных индикаторов. Каждый из них работает независимо от другого и при необходимости количество может быть изменено. В разные периоды времени индикаторы работают лучше, либо хуже. Алгоритм модуля управления капиталом переключает модель на тот индикатор, который по правилу выбора сигнала является наилучшим.

2.5.1 Индикатор «Боллинджера»

Индикатор получил широкое распространение, благодаря американскому техническому аналитику из Калифорнии и автору этого индикатора John Bollinger.

Графически Bollinger Bands представляет собой две линии, ограничивающие динамику цены сверху и снизу соответственно. Это своеобразные линии поддержки и сопротивления, которые большую часть времени находятся на удаленных от цены уровнях (рисунок 2.17).

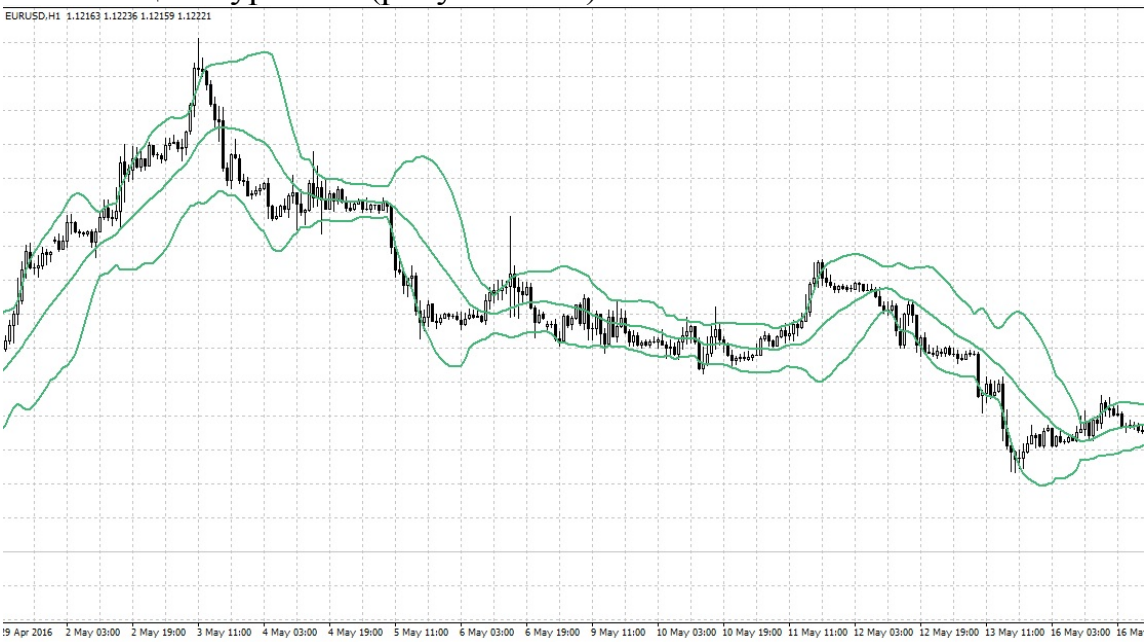

Рисунок 2.17 – Полосы Боллинджера

Основным правилом при построении линий Bollinger является следующее утверждение, около 5% цен должно находиться за пределами этих линий, а 95% внутри [10].

Полосы Боллинджера формируются из трех линий. Средняя линия – это обычное скользящее среднее. Верхняя линия – это та же средняя линия, смещенная вверх на определенное число стандартных отклонений (например, на два). Нижняя линия – это средняя линия, смещенная вниз на то же число стандартных отклонений.

Уникальность диапазонов Боллинджера состоит в том, что их ширина изменяется в ответ на изменение неустойчивости рынка. Полоса Боллинджера строится, как полоса вокруг средней, но ширина полосы пропорциональна среднеквадратичному отклонению от скользящей средней за анализируемый период. Когда на рынке присутствует большая волатильность, например, во время выпуска новостей, полоса расширяется, когда на рынке затишье сужается.

Боллинджеры определяют естественные экстремумы в развивающейся тенденции. Если Боллинджер стремится вверх, цена совершает отскок до тех пор, пока некая достаточно мощная сила не остановит ход цены. Зона застоя образуется ниже верхнего или выше нижнего Боллинджера. Состояние застоя может продолжаться до тех пор, пока Боллинджер не развернется и не начнет, раскрываясь, отходить от ценового бара, что будет свидетельствовать о том, что сопротивление преодолено. Цена может выстреливать в сторону текущей тенденции и придерживаться кромки Боллинджера. Однако не стоит упускать из вида то, что окончательное ценовое движение зависит от всех уровней поддержки/сопротивления, а не только от тех, с которыми оно ассоциируется.

Полосы Боллинджера формируются из трех линий. Средняя линия (Avr , рассчитывается, согласно формуле 2.1.

$$Avr = \frac{\sum_{j=1}^{n} CP_j}{n} , \tag{2.2}$$

где n – число единичных отрезков времени, составляющих период расчета; CP – цена закрытия.

Верхняя линия (UL), рассчитывается, согласно формуле 2.2.

$$UL = Avr + D * \sqrt{\frac{\sum_{j=1}^{n} \left(CP_j - Avr\right)^2}{n}} , \tag{2.3}$$

где n – число единичных отрезков времени, составляющих период расчета; D – число стандартных отклонений; CP – цена закрытия.

Нижняя линия (BL), рассчитывается, согласно формуле 2.3.

$$BL = Avr - D * \sqrt{\frac{\sum_{j=1}^{n} \left(CP_j - Avr\right)^2}{n}} , \tag{2.4}$$

где n – число единичных отрезков времени, составляющих период

расчета; D – число стандартных отклонений; CP – цена закрытия.

Полосы Боллинджера одинаково хорошо работают на любых таймфреймах, но, как правило, их применяют для внутридневной торговли.

Когда цена пробивает или касается верхней границы Боллинджера и затем возвращается обратно – это будет служить сигналом к продаже, если цена пробивает или касается нижней границы Боллинджера, то это послужит сигналом для покупки.

2.5.2 Индикатор «Скользящая Средняя»

Индикатор скользящая средняя (Moving Average, MA) – технический индикатор в основе которого лежит анализ поведения котировок и их скользящего среднего. Индикатор MA выглядит, как фильтр для низких частот. Это значит, что благодаря инструменту не пропускается низкочастотная активность. MA индикатор работает с долгосрочными циклами и их трендовыми линиями, отсекая (не учитывая) случайные высокочастотные 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. Стратегия, при которой инструмент покупается при условии, что график цены пересекает свою скользящую среднюю снизу вверх, и продаётся, когда график цены пересекает график скользящей средней сверху вниз. И то и другое явление называют пробоем. Кроме того, полагают, что если линия графика цены находится выше скользящей средней, то рынок считается «бычьим», на котором можно покупать, а если наоборот – «медвежьим», предпочтительным для продажи (рисунок 2.18) [10].

Рассчитывается MA по формуле 2.4.

$$MA = \frac{\sum_{i=1}^{n} P_i}{n}, \qquad (2.5)$$

где n – число единичных отрезков времени, составляющих период расчета; P – цена закрытия текущего периода.

Рисунок 2.18 – Скользящая средняя

Формирование сигнала для покупки. Для этого возьмем значение быстрой МА на 2-ом баре и сравним его со значением медленной МА на 1-ом баре, а также сравним значение быстрой МА на 1-ом баре со значением медленной МА на 1-ом баре. Если быстрая МА на 2-ом баре меньше значения медленной МА на 1-ом баре и если значение быстрой МА на 1-ом баре больше значения медленной МА на 1-ом баре, это значит, что произошло пересечение быстрой МА медленную МА снизу вверх, и это и будет сигналом для покупки. Аналогично для продажи.

2.5.3 Индикатор «Схождение/Расхождение Скользящих Средних»

Технический Индикатор Схождение/Расхождение Скользящих Средних (Moving Average Convergence/Divergence, MACD) – это следующий за тенденцией динамический индикатор. Он показывает соотношение между двумя скользящими средними цены. Технический Индикатор MACD строится как разность между двумя экспоненциальными скользящими средними с периодами в 12 и 26. Чтобы четко обозначить благоприятные моменты для покупки или продажи, на график MACD наносится так называемая сигнальная линия – 9-периодное скользящее среднее индикатора, расчеты согласна формулы 2.4.

MACD наиболее эффективен в условиях, когда рынок колеблется с большой амплитудой в торговом коридоре. Чаще всего используемые сигналы MACD – пересечения, состояния перекупленности/перепроданности и расхождения.

Основное правило торговли с помощью MACD построено на пересечениях индикатора со своей сигнальной линией: когда Moving Average Convergence/Divergence опускается ниже сигнальной линии – следует продавать, а когда поднимается выше сигнальной линии – покупать. В качестве сигналов к покупке/продаже также используются пересечения MACD нулевой линии вверх/вниз.

Когда между MACD и ценой образуется расхождение, это означает возможность скорого окончания текущей тенденции. Бычье расхождение

возникает тогда, когда MACD достигает новых максимумов, а цене не удается их достичь. Медвежье расхождение образуется, когда индикатор достигает новых минимумов, а цена – нет. Оба вида расхождений наиболее значимы, если они формируются в областях перекупленности/перепроданности.

Технический индикатор Moving Average Convergence/Divergence определяется путем вычитания 26-периодного экспоненциального скользящего среднего из 12-периодного. Затем на график MACD пунктиром наносится его 9-периодное простое скользящее среднее, которое выполняет роль сигнальной линии.

Расчет МА приведен в формуле 2.4

Технический индикатор Moving Average Convergence/Divergence определяется путем вычитания 26-периодного экспоненциального скользящего среднего из 12-периодного. Затем на график MACD пунктиром наносится его 9-периодное простое скользящее среднее, которое исполняет роль сигнальной линии (рисунок 2.19).

Рисунок 2.19 – Индикатор Схождение/Расхождение Скользящих Средних

Когда сигнальная линия пересекает основную линию сверху вниз – это будет сигналом для покупки, если сигнальная линия пересекает основную снизу вверх – это будет сигналом для продажи. В остальных случаях будет фиксироваться как отсутствие сигнала.

2.5.4 Индикатор «Стохастический Осциллятор»

Технический индикатор Стохастический Осциллятор (Stochastic Oscillator) сопоставляет текущую цену закрытия с диапазоном цен за выбранный период времени. Индикатор представлен двумя линиями. Главная линия называется K. Вторая линия D – это скользящее среднее линии K. Обычно K изображается сплошной линией, а D – пунктирной. Существует три наиболее распространенных способа интерпретации Стохастического Осциллятора:

Покупайте, когда осциллятор (K или D) сначала опустится ниже определенного уровня (обычно 20), а затем поднимется выше него, и продавайте, когда осциллятор сначала поднимется выше определенного уровня (обычно 80), а потом опустится ниже него;

Покупайте, если линия K поднимается выше линии D, и продавайте, если линия K опускается ниже линии D.

Следите за расхождениями, например, когда цены образуют ряд новых максимумов, а Стохастическому Осциллятору не удается подняться выше своих предыдущих максимумов (рисунок 2.20).

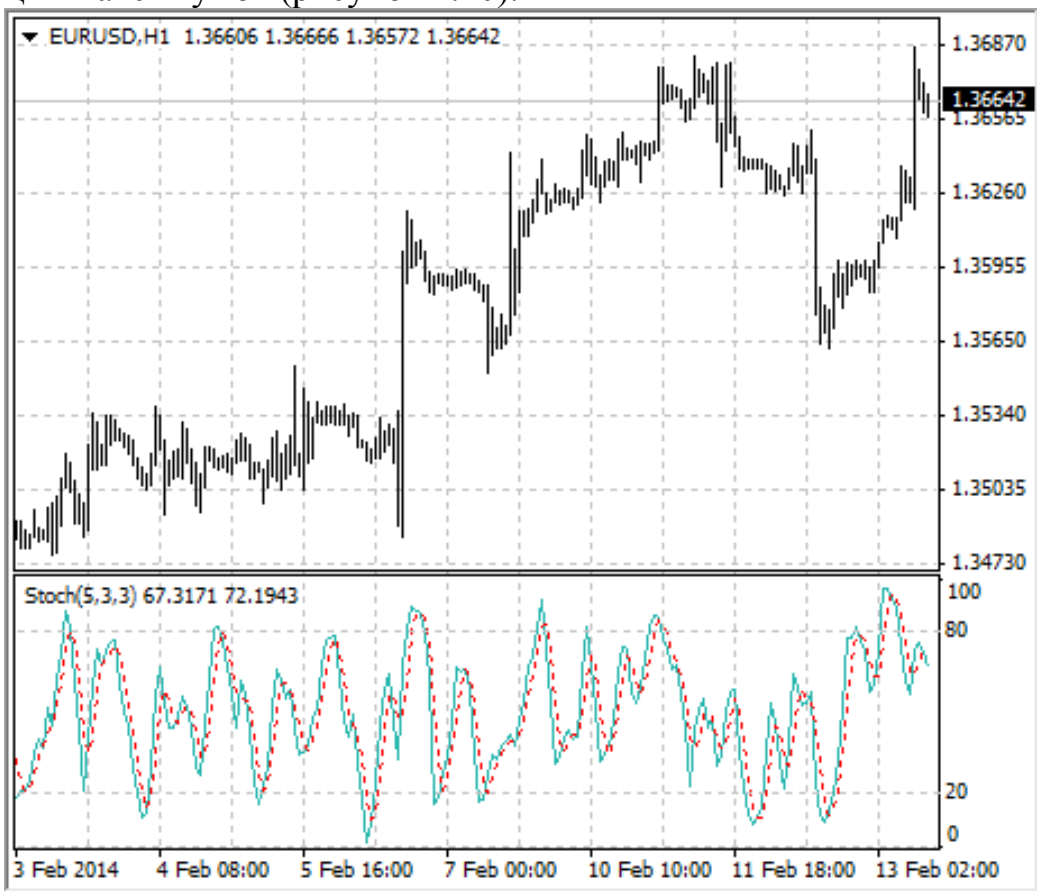

Рисунок 2.20 – Индикатор Стохастический Осциллятор

Покупаем, когда осциллятор (K или D) сначала опустится ниже определенного уровня, а затем поднимется выше него. Продаем, когда осциллятор сначала поднимется выше определенного уровня, а потом опустится ниже него.

2.5.5 Индикатор «Индекс Товарного Канала»

Индекс товарного канала (Commodity Channel Index – CCI) – это разработка известного в определенных кругах трейдера Дональда Ламберта. Изначально этот индикатор создавался для определения разворотных моментов на рынке товаров. Но теперь его часто используют для фондовых рынков и рынка Forex. Индекс товарного канала основан на принципе, согласно которому цены движутся циклично, при этом максимальные и минимальные значения появляются через определенный интервал времени. В качестве делителя в этом показателе используется величина среднего отклонения.

Технический индикатор Индекс Товарного Канала измеряет отклонение цены инструмента от его среднестатистической цены. Высокие значения индекса указывают на то, что цена необычно высока по сравнению со средней, а низкие – что она слишком занижена. Несмотря на название, Commodity Channel Index применим к любому финансовому инструменту, а не только к товарам. Суть индикатора заключается в том, что, после определенных вычислений с данными о цене, мы получаем значение индикатора, которое колеблется вокруг нуля. Когда цена анализируемого финансового инструмента достигает некой экстримальной отметки, то значение индикатора выходит за пределы [-100; 100]. Таким образом, те области на графике индикатора, где значение индекса товарного канала превышает 100 или опускается ниже -100 сигнализируют аналитикам о том, что пора действовать (рисунок 2.21) [11].

Рассчитывается характерная цена TP.

$$TP = \frac{(H + L + C)}{3}, \tag{2.6}$$

где H, L, C – максимум, минимум и цена закрытия соответственно.

Рассчитывается срединное отклонение MD по формуле 2.5.

$$MD = \frac{\sum_{1}^{n} MA - TP_i}{n}, \tag{2.7}$$

где n – число единичных отрезков времени, составляющих период расчета.

Формула самого индикатора CCI будет выглядеть следующим образом:

$$CCI = \frac{TP - MA}{0.015 * MD}, \tag{2.8}$$

Рисунок 2.21 – Индикатор Индекс Товарного Канала

Когда CCI сначала опустится ниже определенного уровня, а затем поднимется выше него – это означает сигнал к покупке. Сигнал к продаже, когда CCI сначала поднимется выше определенного уровня, а потом опустится ниже него.

2.5.6 Индикатор «Процентный Диапазон Вильямса»

Технический Индикатор Процентный Диапазон Вильямса (Williams Percent Range) – это динамический индикатор, определяющий состояние перекупленности/перепроданности. Williams Percent Range очень похож на технический индикатор Stochastic Oscillator. Различие между ними состоит лишь в том, что первый имеет перевернутую шкалу, а второй строится с использованием внутреннего сглаживания.

Значения индикатора в диапазоне от -80% до -100% указывают на состояние перепроданности. Значения в диапазоне от -0% до -20% свидетельствуют о том, что рынок перекуплен. Для построения индикатора Williams Percent Range в перевернутой шкале его значениям обычно присваивается отрицательный знак (например, -30%). При анализе отрицательный знак можно не учитывать [9].

По общему для всех индикаторов перекупленности/перепроданности правилу, действовать по их сигналам лучше всего, дождавшись поворота цен в соответствующем направлении. Так, если индикатор перекупленности/перепроданности указывает на состояние перекупленности, то прежде чем продавать бумагу, разумно дождаться поворота цен вниз.

У индикатора Williams Percent Range есть способность предвосхищать ценовые развороты. Он почти всегда образует пик и поворачивает вниз за определенный промежуток времени до того, как цена достигает пика и поворачивает вниз. Точно так же Williams Percent Range обычно образует впадину и заблаговременно поворачивает вверх (рисунок 2.22).

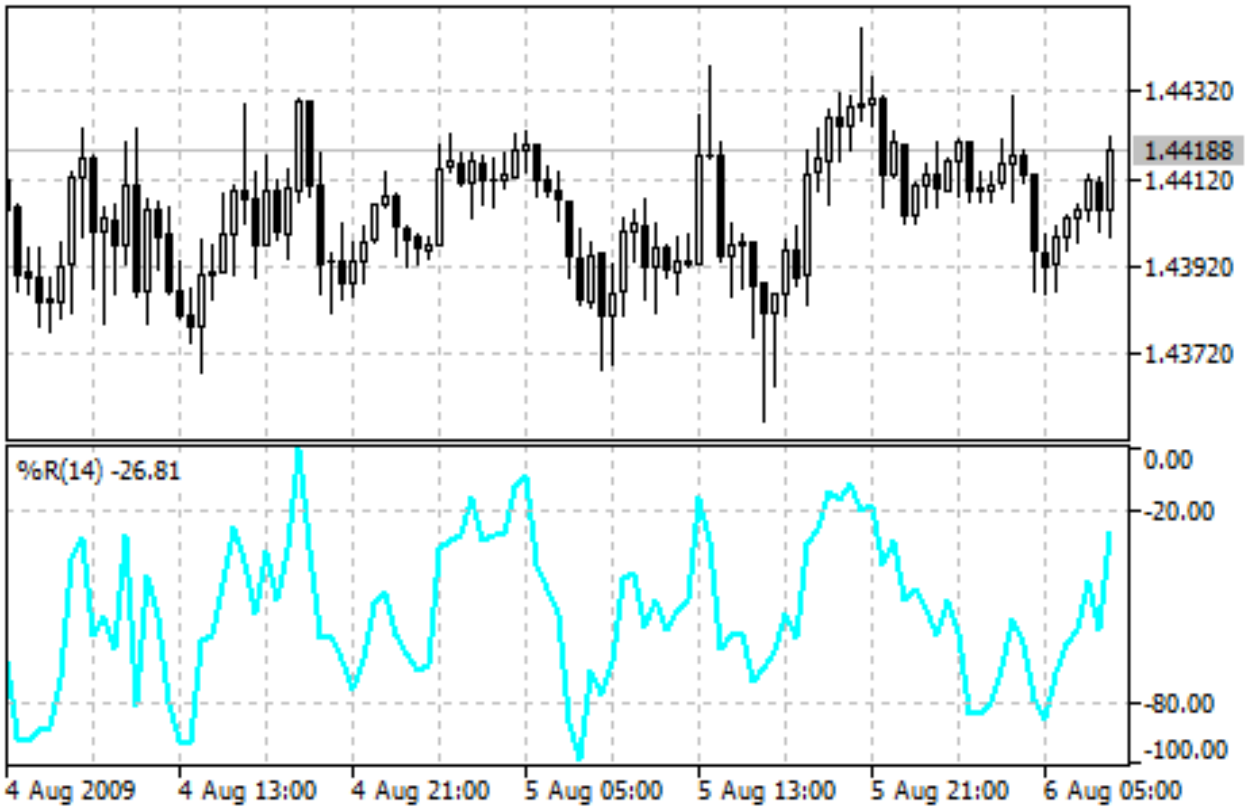

Рисунок 2.22 – Технический Индикатор Процентный Диапазон Вильямса

Сигнал к покупке, когда индикатор сначала опустится ниже уровня -80, а затем поднимется выше него. Продаем, когда индикатор сначала поднимется выше уровня -20, а потом опустится ниже него.

2.5.7 Индикатор «Alligator»

Индикатор Аллигатор был создан автором книг «Торговый Хаос», «Новые измерения в биржевой торговле» Биллом Вильямсом. Он разработал торговую систему, в основе которой было прогнозирование движения цен на основе нескольких индикаторов, в числе которых был и индикатор Аллигатор.

Индикатор Аллигатор – это три простые скользящие средние с разными периодами и разным смещением вперед. Основная функция индикатора – дать сигнал тренда в самом его начале. Особенностью является то, что при расчетах используется не цена закрытия (как в случае с традиционном простым скользящим средним), а средняя цена, т.е. среднее арифметическое максимальной и минимальной цены бара (свечи).

Первое скользящее среднее называется «Челюсть Аллигатора» (англ. Alligators Jaw), с временным периодом, равным 13 и сдвигом на 8 баров вперед, традиционно изображается на графике синим цветом. Второе скользящее среднее – «Зубы Аллигатора» (англ. Alligators teeth) – имеет период 8 и сдвиг, равный 5. Обычно красного цвета. Третье скользящее среднее называется «Губы Аллигатора» (англ. Alligators Lips), оно имеет период, равный 5 и сдвиг, равный 3. Это линия зеленого цвета (Рисунок 2.11).

Все линии Аллигатора показывают уровни цен, которые должен установиться на рынке, если на него не будут влиять новые факторы, причем для разных временных периодов. Сам Билл Вильямс называл линии индикатора «линиями баланса» и интерпретировал их положение следующим образом:

- если все линии переплетены, Аллигатор спит, рынок находится во флете, и сигналов на покупку или продажу нет;
- чем дольше спит Аллигатор, тем он голоднее;
- когда пасть Аллигатора открывается, и цены находятся выше челюсти, это говорит о восходящем тренде;
- когда пасть Аллигатора открывается, и цены находятся ниже губ, это говорит о нисходящем тренде;
- переплетение линий баланса после «охоты» считается моментом фиксации прибыли, т.е. закрытия позиций.

Пример работы продемонстрирован ниже (рисунок 2.23).

Рисунок 2.23 – Индикатор Alligator

2.5.8 Индикатор «Адаптивное Скользящее Среднее»

Технический индикатор Адаптивное Скользящее Среднее (Adaptive Moving Average, AMA) используется для построения скользящей средней с малой чувствительностью к шумам в ценовых сериях и характеризуется минимальным запаздыванием для определения тренда. Его разработал и описал Перри Кауфман в книге "Smarter Trading".

Любые движения цены можно разделить, на две составляющие это, общее движение цены за определенный период и шумовые движения цены внутри этого

периода. Суть, адаптивной скользящей средней, заключается, в достаточно быстром получении сигналов для открытия сделки на динамичном трендовом рынке и своевременном закрытие позиции, когда рынок входит в состояние флета.

Один из недостатков различных алгоритмов сглаживания ценовых рядов заключается в том, что случайные всплески цены могут приводить к появлению ложных сигналов о появлении тренда. С другой стороны, сглаживание приводит к неизбежному запаздыванию сигнала об остановке или развороте тренда. Данный индикатор был разработан с тем, чтобы обойти два этих недостатка (рисунок 2.24).

Рисунок 2.24 – Адаптивное Скользящее Среднее

В отличие от широко известных и применяемых в торговле типов скользящих, обладает неоспоримым преимуществом – расчет средней ведется с динамической оценкой волатильности рынка, поэтому скользящая более четко указывает периоды тренда и периоды флэта.

Адаптивная скользящая средняя использует постоянно меняющийся параметр сглаживания, возрастающий в моменты, когда наклон ценового тренда приближается к вертикали, и падающий, когда наклон ценового тренда подходит к нулю. Иными словами, при быстрорастущем ценовом тренде длина периода становится более короткой, то есть более чувствительной к новым данным, поступающим в вычисления. На ровном ценовом тренде период удлиняется, то есть делается менее чувствительным к вхождению новых данных.

Когда индикатор AMA направлен вверх, это будет служить сигналом к покупке, если AMA направлен вниз, то сигналом к продаже.

2.5.9 Индикатор «Огибающие Линии»

Технический Индикатор Огибающие Линии (Конверты, Envelopes) образуется двумя скользящими средними, одна из которых смещена вверх, а другая – вниз. Выбор оптимальной относительной величины смещения границ полосы определяется волатильностью рынка: чем она выше – тем больше смещение.

Envelopes определяют верхние и нижние границы нормального диапазона колебаний цен бумаги. Сигнал к продаже возникает тогда, когда цена достигает верхней границы полосы, а сигнал к покупке – при достижении ею нижней границы. Применение технического индикатора Envelopes (рисунок 2.25) основано на естественной логике поведения рынка: когда под давлением покупателей или продавцов цены достигают экстремальных значений (т.е. верхней или нижней границы полосы), они часто стабилизируются, возвращаясь к более реалистичным уровням. Такой же принцип используется при интерпретации Полос Боллинджера.

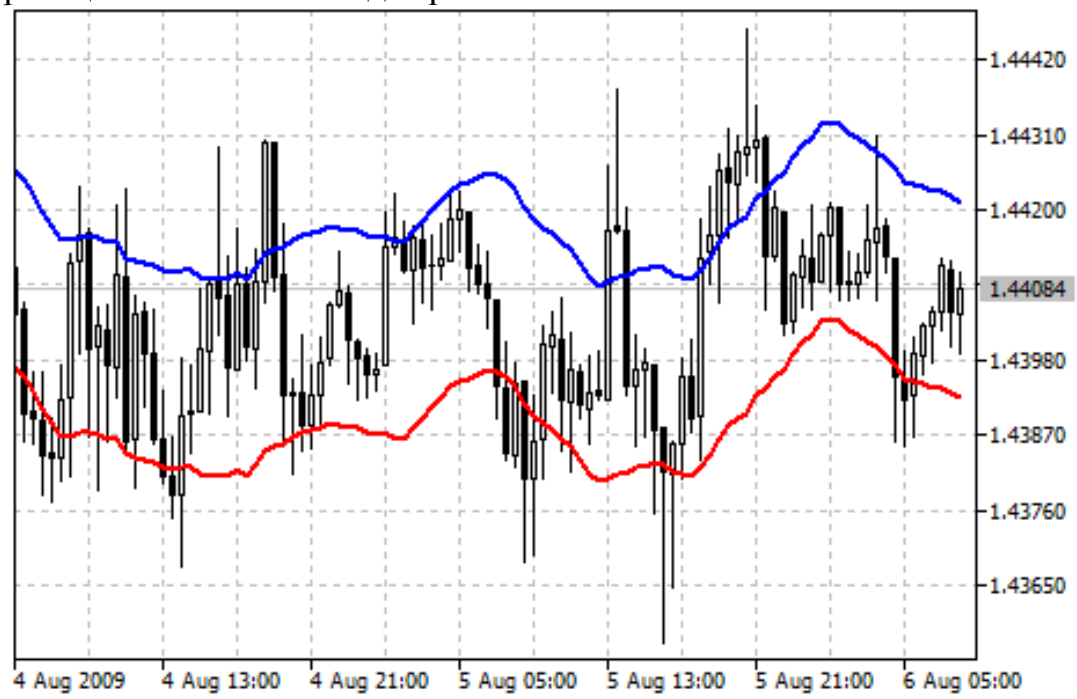

Рисунок 2.25 – Огибающие Линии

Существуют способы использования средних скольжения, чтобы помочь проводить мониторинг уровней поддержки и сопротивления и определения крайних точек рынка. Эта техника размещает линии, называемые конвертами (огибающими), на предопределенный процент выше и ниже средней скользящей линии. Проценты могут различаться в зависимости от того, каков тренд и какой рынок изучается.

Мы можем говорить о том, что выход цены за переделы линий указывает на слабость такого движения, но нельзя однозначно определить насколько затянется такое состояние на рынке. Это позволяет говорить об огибающих линиях, как о весьма ограниченном инструменте, которые может давать скорее рекомендательные сигналы, чем реальные точки входа в рынок. Когда цена пробивает или касается верхней границы Envelopes и затем возвращается обратно

– это будет служить сигналом к продаже, если цена пробивает или касается нижней границы Envelopes, то это послужит сигналом для покупки.

2.5.10 Индикатор «Индекс Относительной Силы»

Технический Индикатор Индекс Относительной Силы (Relative Strength Index, RSI) это следующий за ценой осциллятор, который колеблется в диапазоне от 0 до 100. Вводя Relative Strength Index, У. Уайлдер рекомендовал использовать его 14-периодный вариант. В дальнейшем распространение получили также 9 и 25-периодные индикаторы. Один из распространенных методов анализа индикатора Relative Strength Index состоит в поиске расхождений, при которых цена образует новый максимум, а RSI не удается преодолеть уровень своего предыдущего максимума. Подобное расхождение свидетельствует о вероятности разворота цен. Если затем индикатор поворачивает вниз и опускается ниже своей впадины, то он завершает так называемый «неудавшийся размах» (failure swing). Этот неудавшийся размах считается подтверждением скорого разворота цен. Вершины и основания.

При анализе графиков различают следующие сигналы Relative Strength Index:

Вершины индикатора. Relative Strength Index обычно формируются выше 70, а основания – ниже 30, причем они обычно опережают образования вершин и оснований на ценовом графике.

Графические модели. Relative Strength Index часто образует графические модели – такие как «голова и плечи» или треугольники, которые на ценовом графике могут и не обозначиться.

Неудавшийся размах (прорыв уровня поддержки и сопротивления). Имеет место, когда Relative Strength Index поднимается выше предыдущего максимума (пика) или опускается ниже предыдущего минимума (впадина).

Уровни поддержки и сопротивления. На графике индикатора Relative Strength Index уровни поддержки и сопротивления проступают даже отчетливее, чем на ценовом графике.

Расхождения. Как уже сказано выше, расхождения образуются, когда цена достигает нового максимума (минимума), но он не подтверждается новым максимумом (минимумом) на графике RSI. При этом обычно происходит коррекция цен в направлении движения индикатора Relative Strength Index.

Покупаем, когда RSI сначала опустится ниже уровня 30, а затем поднимется выше него. Продаем, когда RSI сначала поднимется выше уровня 70, а потом опустится ниже него.

Пример работы технического индикатора индекса относительной силы представлен ниже (рисунок 2.26).

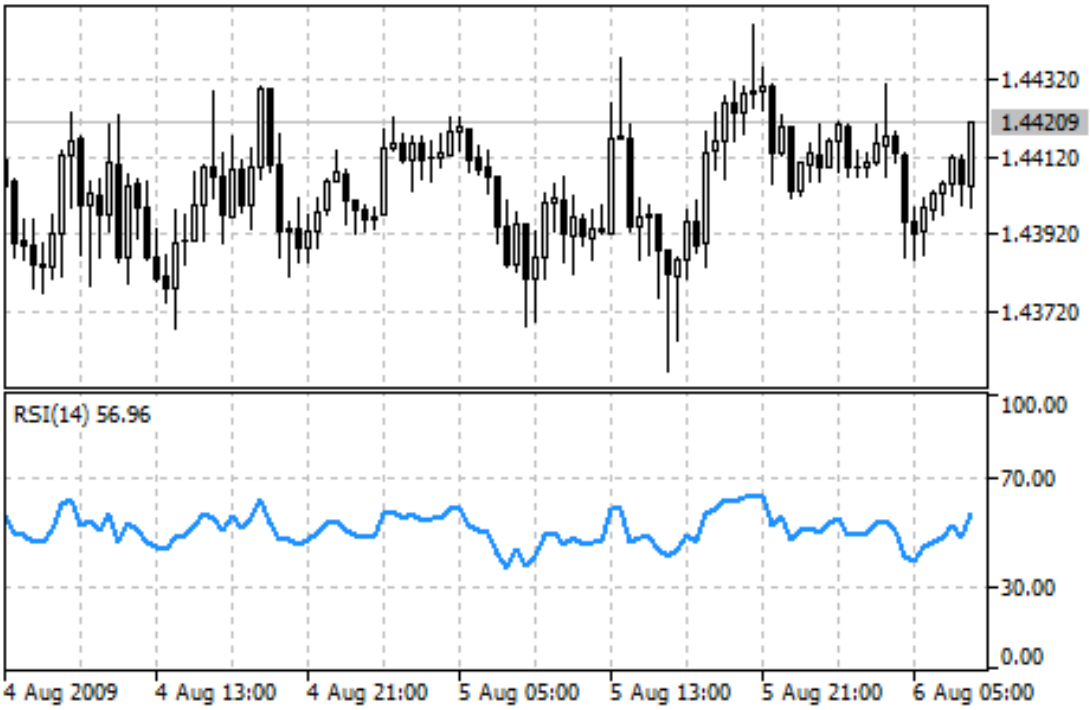

Рисунок 2.26 – Индекс Относительной Силы

2.5.11 Индикатор «Стандартное отклонение»

Технический Индикатор Стандартное отклонение (Standard Deviation, StdDev) измеряет волатильность рынка. Этот индикатор характеризует размер колебаний цены относительно скользящего среднего. Так, если значение индикатора велико, то рынок является волатильным и цены баров достаточно разбросаны относительно скользящего среднего. Если значение индикатора невелико, рынок характеризуется низкой волатильностью и цены баров достаточно близки к скользящему среднему.

Обычно этот индикатор используется как составная часть других индикаторов. Так, при расчете Bollinger Bands значение стандартного отклонения инструмента прибавляется к его скользящему среднему.

Динамика рынка состоит в последовательном чередовании периодов покоя и всплесков активности, поэтому подход к данному индикатору прост:

если значение индикатора слишком мало, то есть рынок в полном покое, то имеет смысл ожидать скорого всплеска активности;

напротив, если индикатор экстремально велик, значит, скорее всего, эта активность скоро пойдет на убыль (рисунок 2.27).

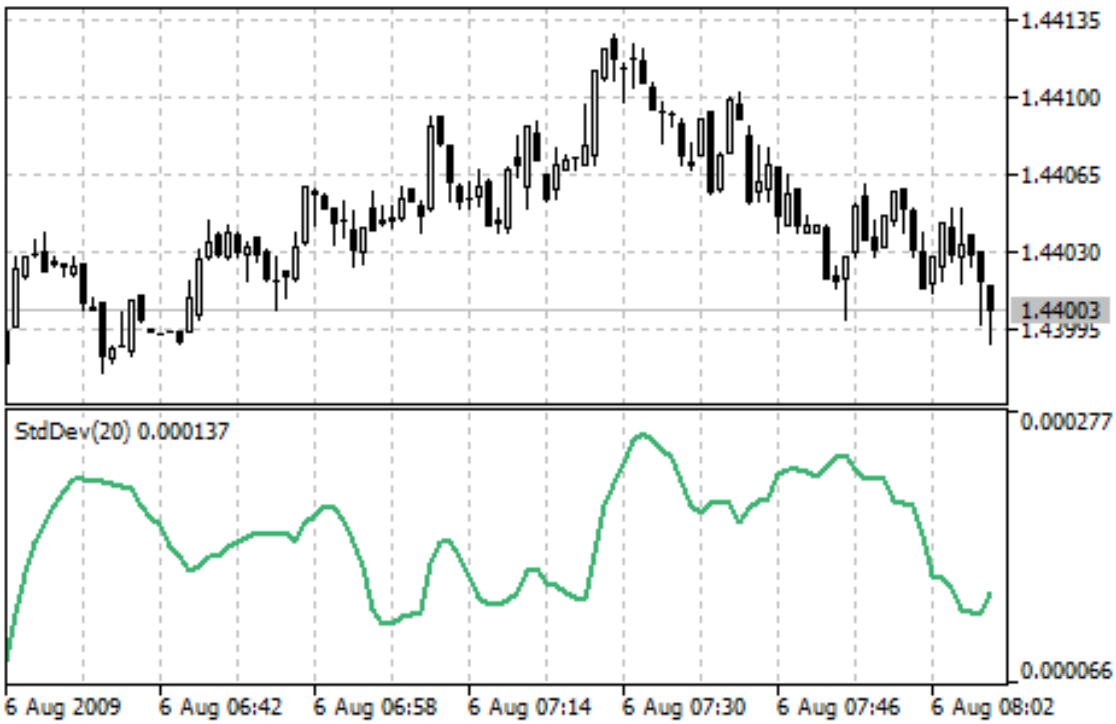

Рисунок 2.27 – Стандартное отклонение

2.6 Выводы по разделу

В данной главе приведены алгоритмы работы модуля управления капиталом, составлены пояснительные материалы к исходному коду. Приведены три торговых стратегии выбора сигнала: Правило К, Правило В и Голосование Большинства. Рассмотрены 11 сигналов, входящих в АКМ, их принцип работы и торговая стратегия.

3 РЕЗУЛЬТАТЫ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА

3.1 Описание тестового стенда

Для проверки качеств торгового робота в MetaTrader встроен Тестер торговых стратегий. Еще до запуска советника в торговлю он позволяет определить его эффективность и подобрать наилучшие входные параметры.

Вся работа Тестера торговых стратегий строится на истории котировок валют и акций. Во время тестирования робот анализирует накопленные котировки и совершает виртуальные сделки в соответствии с заложенным в него алгоритмом. Это позволяет оценить, как бы данная стратегия торговала в прошлом, сохраняя при этом капитал.

Тестирование работы модуля управления капиталом осуществляется с помощью тестера торговых стратегий MetaTrader.

Тестер стратегий в MetaTrader является мультивалютным. Тестируемые в нем роботы имеют доступ ко всем финансовым инструментам и могут торговать на них. Инструмент позволяет испытывать даже сложных советников, которые способны анализировать сразу несколько валют и корреляцию между ними.

Главным преимуществом тестирования является оценка торгового робота без его реальной работы на рынке. Кроме того, в тестере это занимает намного меньше времени – всего несколько минут против дней, недель и месяцев при тестировании эксперта на реальном рынке.

Поддержка распределенного тестирования (MQL Cloud Network) и оптимизации позволяют подключать к этим процессам дополнительные вычислительные мощности. Например, можно использовать вычислительные мощности компьютеров локальной сети и в несколько раз ускорить процесс оптимизации.

Параметры тестирования:

- период тестирования равен одному году начиная с 01.01.2016 по 31.12.2016;
- валюта для испытания выбрана EURUSD;
- тестирование проводится с моделированием задержки 75 мс, выбор данной задержки обусловлен средней задержкой до брокера;
- каждый тик полученный модулем от терминала MetaTrader вычисляется на основе реальных тиков, это означает, что возможны пропуски и задержки информации;
- начальный депозит составляет 100000 USD;
- брокер предоставляющий историю MetaQuotes Software Corp;
- кредитное плечо, предоставляемое брокером 1:100;
- все параметры оптимизации отключены.

Оценки результатов тестирования:

- чистая прибыль – прибыль за весь период тестирования, в единицах базовой валюты;

- общая прибыль – сумма прибыли всех прибыльных сделок;
- общий убыток – это сумма убытков всех убыточных сделок за прошедший период тестирования;
- всего сделок – это общее количество прибыльных и убыточных сделок за период тестирования;
- средняя прибыльная сделка – это средний размер прибыли прибыльных сделок;
- прибыльные сделки – общее количество прибыльных сделок и процент от общего количества сделок;
- самая большая прибыльная сделка – самая большая прибыль среди прибыльных сделок;
- максимальное количество непрерывных выигрышей – это максимальное количество прибыльных подряд идущих сделок за выбранный период тестирования;
- средний непрерывный выигрыш – среднее количество прибыльных сделок подряд;
- абсолютная просадка по средствам – отображает максимальное уменьшение объема денежных средств на счете клиента относительно его первоначальной величины;
- коэффициент Шарпа – это параметр, который показывает насколько доход от стратегии соотносится к потенциальному риску. Расчет приведен в формуле 3.9

$$S(X)=\frac{(r_x-R_f)}{StdDev(X)}\ , \tag{3.9}$$

где X – торговый актив; . r_x – доходность актива X; R_f – безрисковый доход; $StdDev(X)$ – стандартное отклонение r_x;

- максимальное количество непрерывных проигрышей – это число максимального количества подряд идущих убыточных сделок за весь период тестирования;
- средний непрерывный проигрыш – среднее количество убыточных сделок подряд за весь период тестирования;
- убыточные сделки – общее количество прибыльных сделок и процент от общего количества сделок за весь период тестирования.

Результаты тестирования торговых стратегий приведены ниже в таблицах 1-14 и на рисунках 3.16-3.29.

3.2 Результаты индивидуальной работы индикаторов

Таблица 1

Индикатор Аллигатор

Чистая прибыль, $	543.38
Общая прибыль, $	7 667.21
Общий убыток, $	7 123.83
Всего сделок	3024
Средняя прибыльная сделка, $	3.87
Прибыльные сделки, (% от всех)	1980 (65.48%)
Самая большая прибыльная сделка, $	27.81
Максимальное количество непрерывных выигрышей	144
Средний непрерывный выигрыш	54
Абсолютная просадка по средствам, $	1 703.08
Коэффициент Шарпа	0.03
Матожидание выигрыша	0.18
Максимальное количество непрерывных проигрышей	126
Средний непрерывный проигрыш	28
Убыточные сделки (% от всех)	1044 (34.52%)

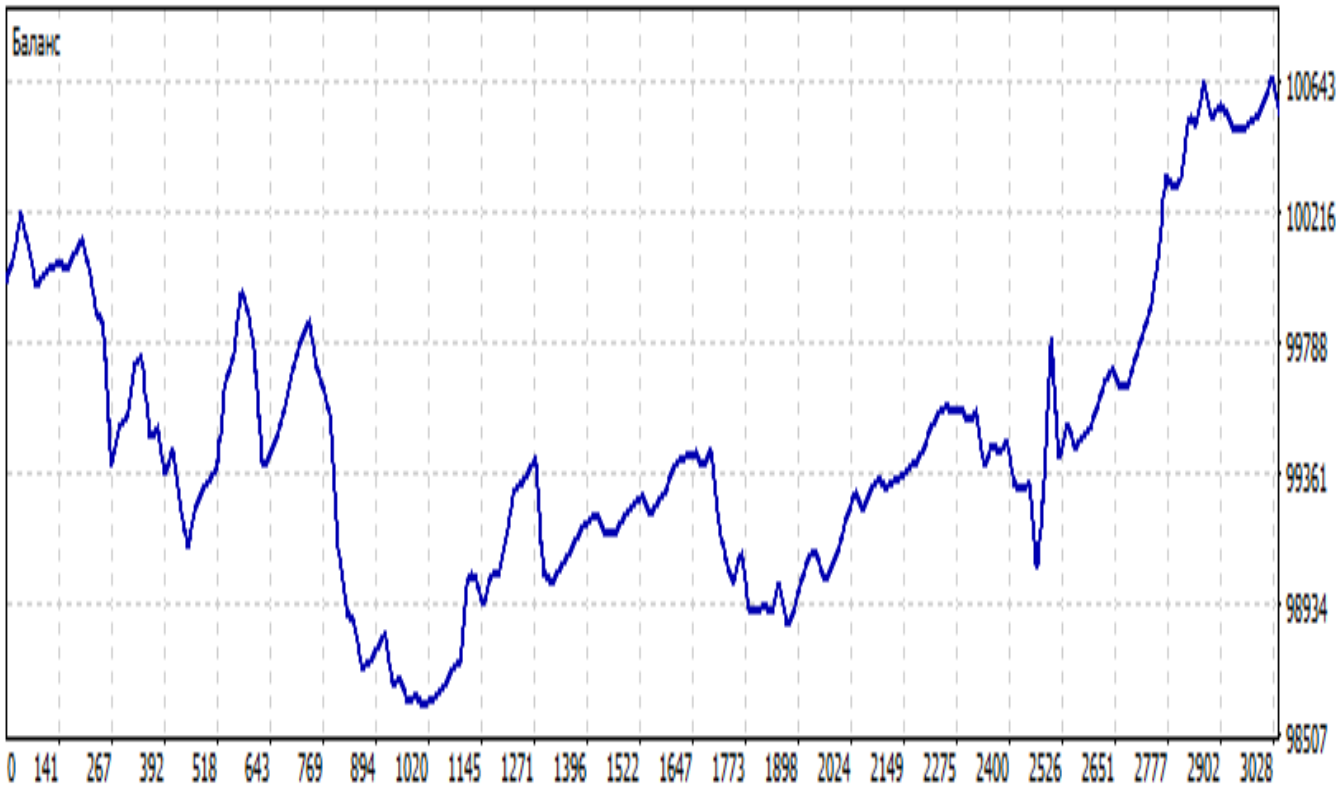

Рисунок 3.28 – Баланс, $ / сделки для индикатора Аллигатор

Таблица 2

Индикатор Огибающие Линии

Чистая прибыль, $	262.32
Общая прибыль, $	8 794.01
Общий убыток, $	8 531.69
Всего сделок	4452
Средняя прибыльная сделка, $	4.40
Прибыльные сделка, (% от всех)	1998 (44.88%)
Самая большая прибыльная сделка, $	25.39
Максимальное количество непрерывных выигрышей	126
Средний непрерывный выигрыш	43
Абсолютная просадка по средствам, $	210.11
Коэффициент Шарпа	0.01
Матожидание выигрыша	0.06
Максимальное количество непрерывных проигрышей	168
Средний непрерывный проигрыш	52
Убыточные сделки (% от всех):	2454 (55.12%)

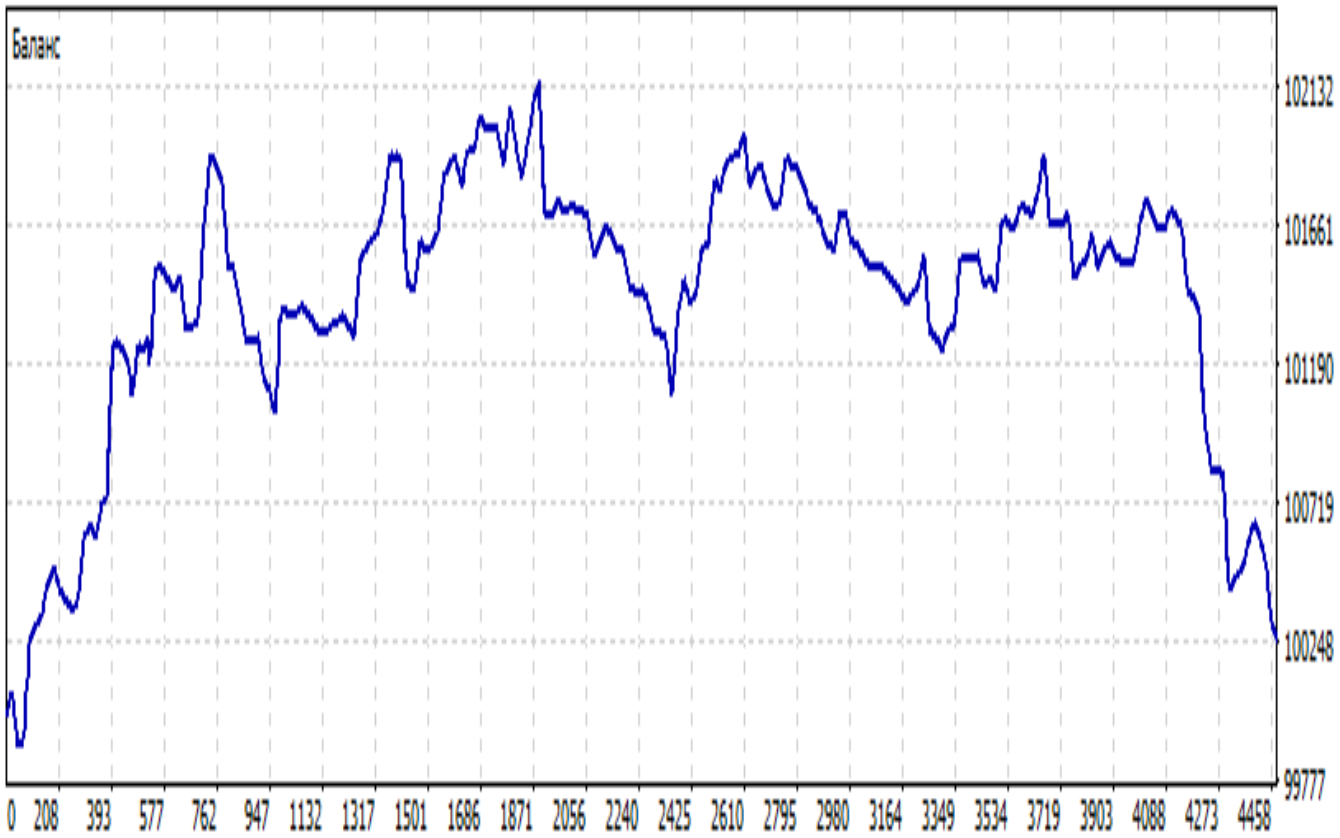

Рисунок 3.29 – Баланс, $ / сделки для индикатора Огибающие Линии

Таблица 3

Индикатор Стандартное отклонение

Чистая прибыль, $	-2 826.74
Общая прибыль, $	5 075.51
Общий убыток, $	7 902.25
Всего сделок	2142
Средняя прибыльная сделка, $	6.40
Прибыльные сделки, (% от всех)	793 (37.02%)
Самая большая прибыльная сделка, $	24.71
Максимальное количество непрерывных выигрышей	72
Средний непрерывный выигрыш	28
Абсолютная просадка по средствам, $	3 625.52
Коэффициент Шарпа	-0.15
Матожидание выигрыша	-1.32
Максимальное количество непрерывных проигрышей	162
Средний непрерывный проигрыш	48
Убыточные сделки (% от всех):	1349 (62.98%)

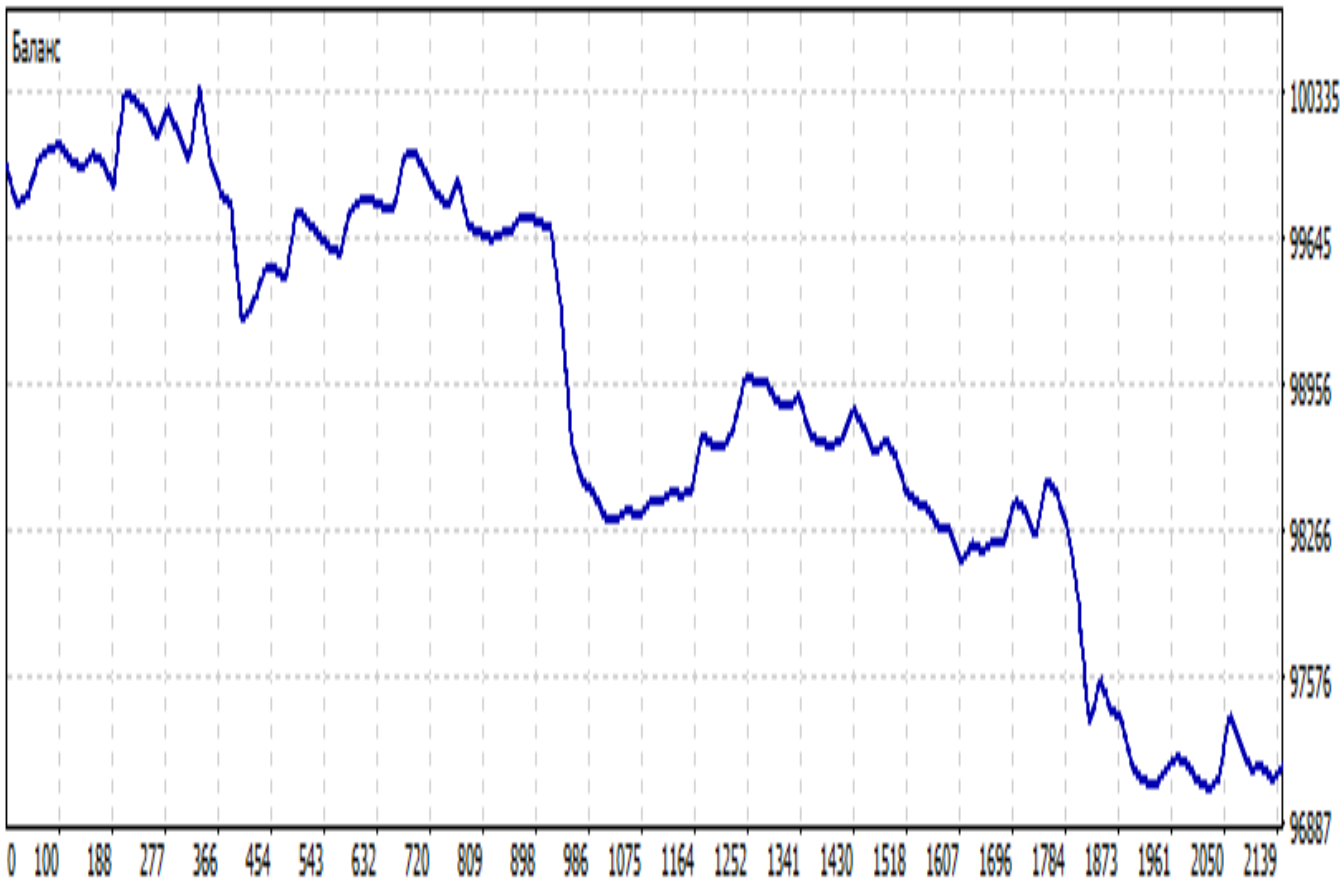

Рисунок 3.30 – Баланс, $ / сделки для индикатора Стандартное отклонение

Таблица 4

Индикатор Боллинджера

Чистая прибыль, $	-3 394.05
Общая прибыль, $	6 196.01
Общий убыток, $	9 590.06
Всего сделок	2499
Средняя прибыльная сделка, $	6.73
Прибыльные сделки, (% от всех)	920 (36.81%)
Самая большая прибыльная сделка, $	25. 38
Максимальное количество непрерывных выигрышей	84
Средний непрерывный выигрыш	27
Абсолютная просадка по средствам, $	3 942.77
Коэффициент Шарпа	-0.15
Матожидание выигрыша	-1.36
Максимальное количество непрерывных проигрышей	185
Средний непрерывный проигрыш	46
Убыточные сделки (% от всех):	1579 (63.19%)

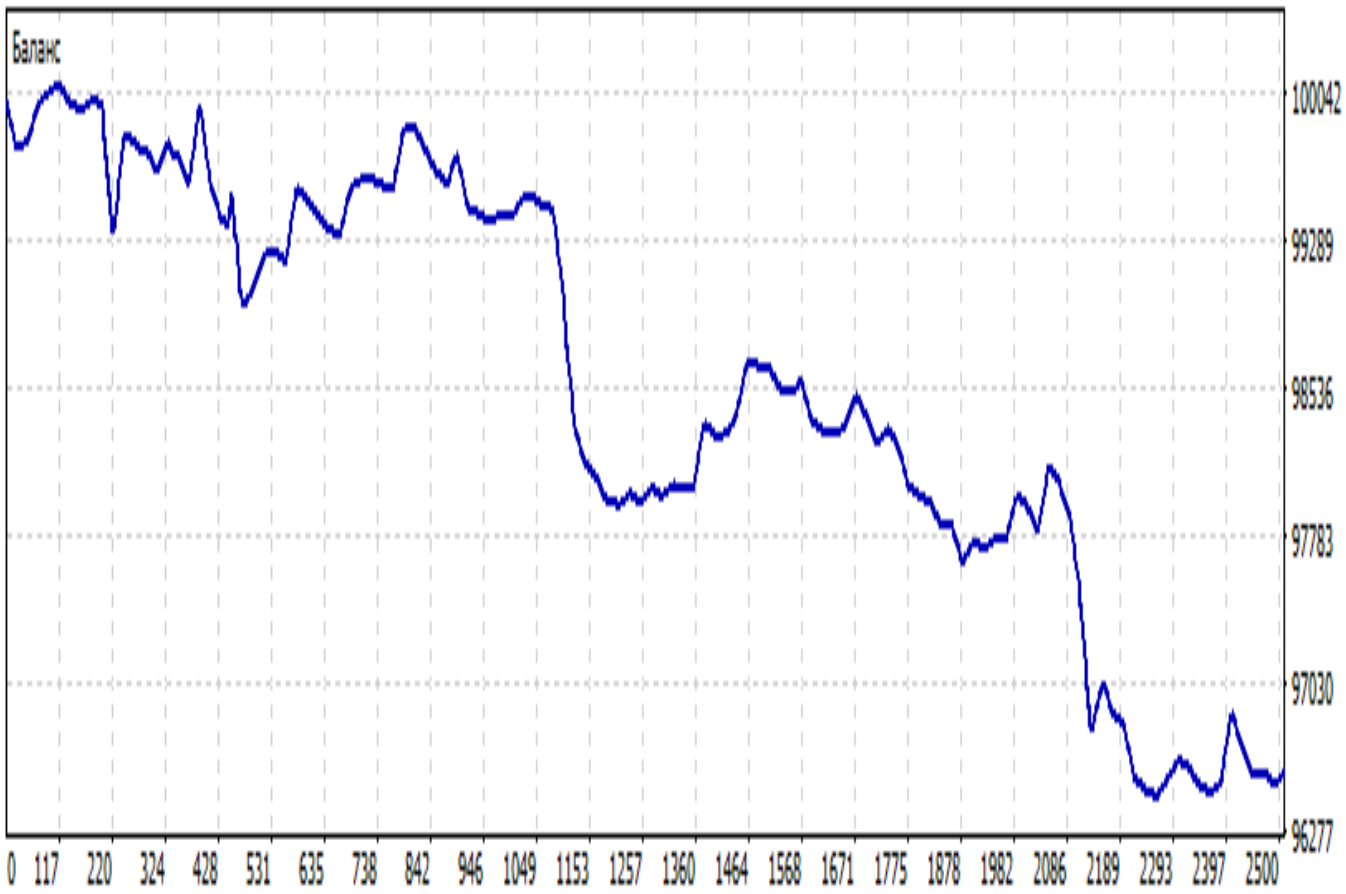

Рисунок 3.31 – Баланс, $ / сделки для индикатора Боллинджера

Таблица 5

Индикатор Процентный Диапазон Вильямса

Чистая прибыль, $	-549.79
Общая прибыль, $	3 076.54
Общий убыток, $	3 626.33
Всего сделок	3972
Средняя прибыльная сделка, $	4.21
Прибыльные сделки, (% от всех)	731 (36.81%)
Самая большая прибыльная сделка, $	27.51
Максимальное количество непрерывных выигрышей	30
Средний непрерывный выигрыш	10
Абсолютная просадка по средствам, $	842.00
Коэффициент Шарпа	-0.05
Матожидание выигрыша	-0.28
Максимальное количество непрерывных проигрышей	66
Средний непрерывный проигрыш	17
Убыточные сделки (% от всех):	1255 (63.19%)

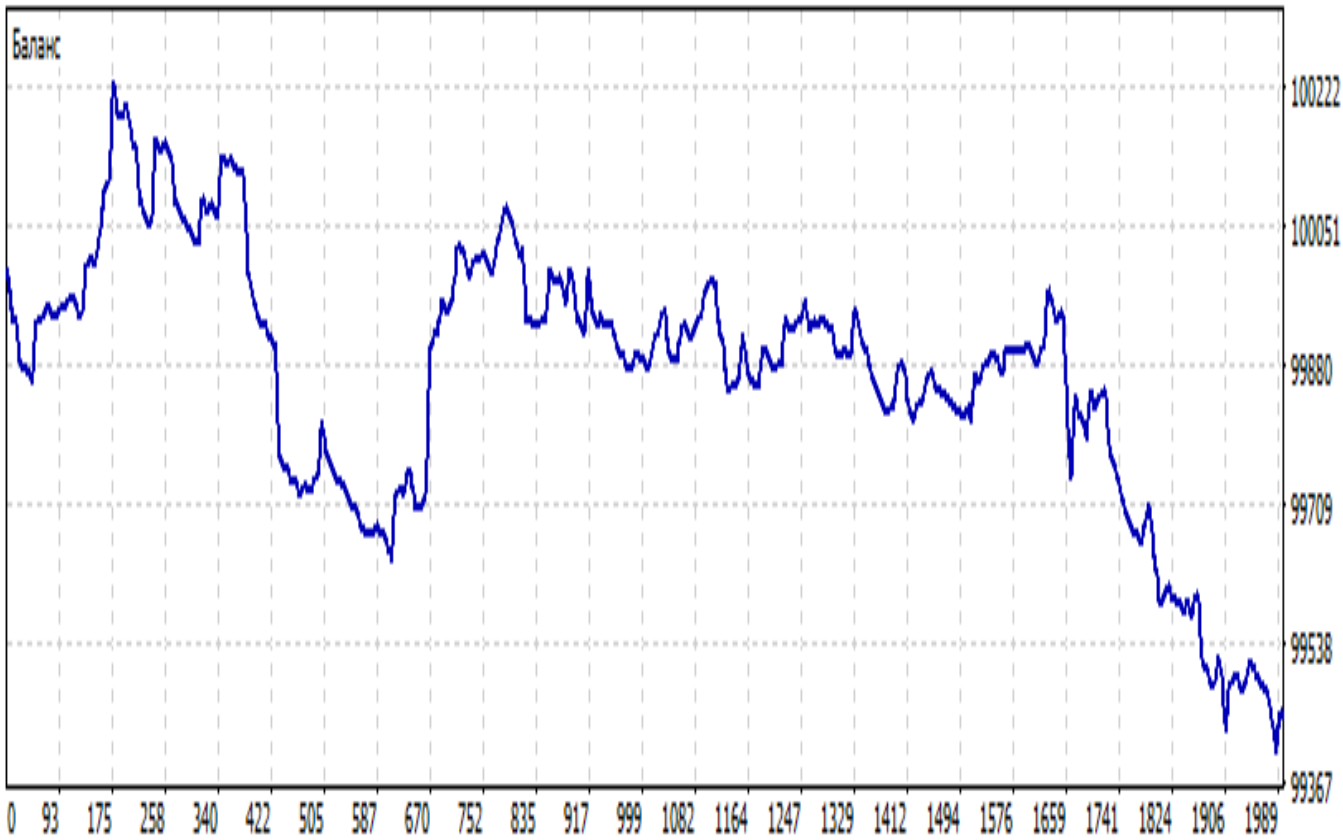

Рисунок 3.32 Баланс, $ / сделки для индикатора Процентный Диапазон Вильямса

Таблица 6

Индикатор Индекс Товарного Канала

Чистая прибыль, $	-396.17
Общая прибыль, $	3 082.97
Общий убыток, $	3 479.14
Всего сделок	2052
Средняя прибыльная сделка, $	3.65
Прибыльные сделки, (% от всех)	845 (41.18%)
Самая большая прибыльная сделка, $	26.77
Максимальное количество непрерывных выигрышей	36
Средний непрерывный выигрыш	10
Абсолютная просадка по средствам, $	847.22
Коэффициент Шарпа	-0.04
Матожидание выигрыша	-0.19
Максимальное количество непрерывных проигрышей	48
Средний непрерывный проигрыш	14
Убыточные сделки (% от всех):	1207 (58.82%)

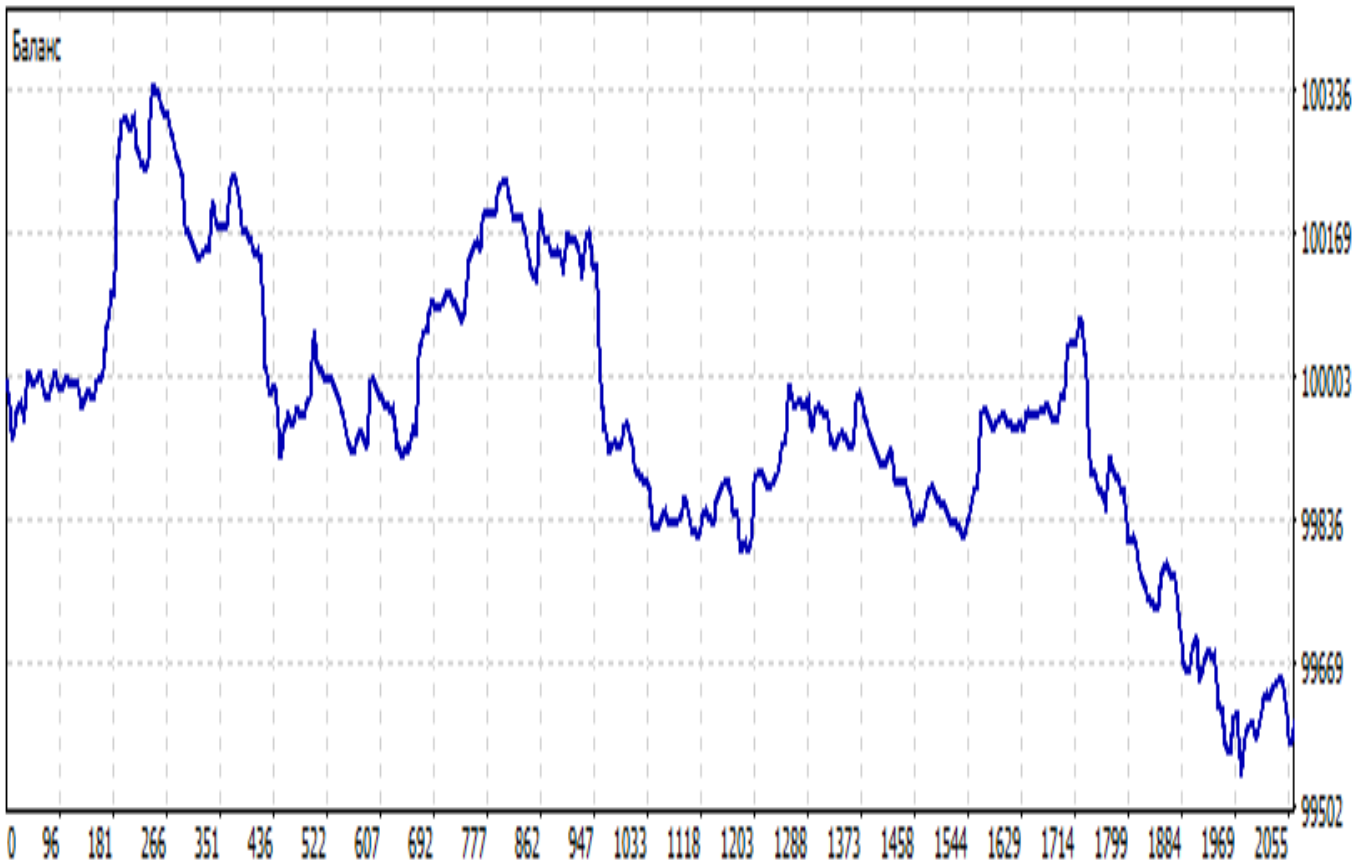

Рисунок 3.33 Баланс, $ / сделки для индикатора Индекс Товарного Канала

Таблица 7

Индикатор Индекс Относительной Силы

Чистая прибыль, $	-952.35
Общая прибыль, $	3 716.56
Общий убыток, $	4 668.91
Всего сделок	900
Средняя прибыльная сделка, $	9.83
Прибыльные сделки, (% от всех)	378 (42.00%)
Самая большая прибыльная сделка, $	29.39
Максимальное количество непрерывных выигрышей	72
Средний непрерывный выигрыш	34
Абсолютная просадка по средствам, $	1 732.79
Коэффициент Шарпа	-0.09
Матожидание выигрыша	-1.06
Максимальное количество непрерывных проигрышей	90
Средний непрерывный проигрыш	47
Убыточные сделки (% от всех):	522 (58.00%)

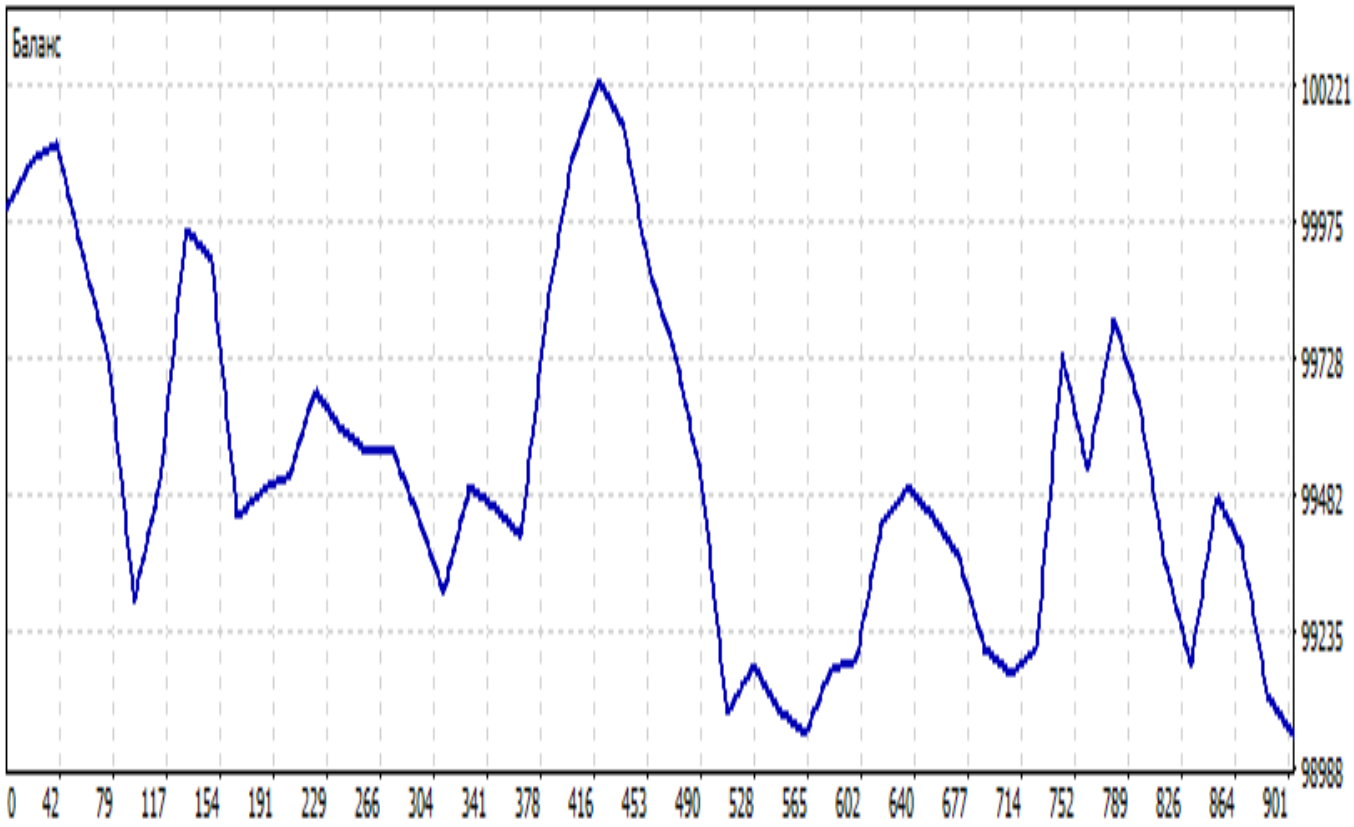

Рисунок 3.34 Баланс, $ / сделки для индикатора Индекс Относительной Силы

Таблица 8

Индикатор Адаптивное Скользящее Среднее

Чистая прибыль, $	-1 490.19
Общая прибыль, $	13 170.36
Общий убыток, $	14 660.55
Всего сделок	16110
Средняя прибыльная сделка, $	1.17
Прибыльные сделки, (% от всех)	11280 (70.02%)
Самая большая прибыльная сделка, $	13.38
Максимальное количество непрерывных выигрышей	208
Средний непрерывный выигрыш	56
Абсолютная просадка по средствам, $	2 352.74
Коэффициент Шарпа	-0.03
Матожидание выигрыша	-0.09
Максимальное количество непрерывных проигрышей	90
Средний непрерывный проигрыш	24
Убыточные сделки (% от всех):	4830 (29.98%)

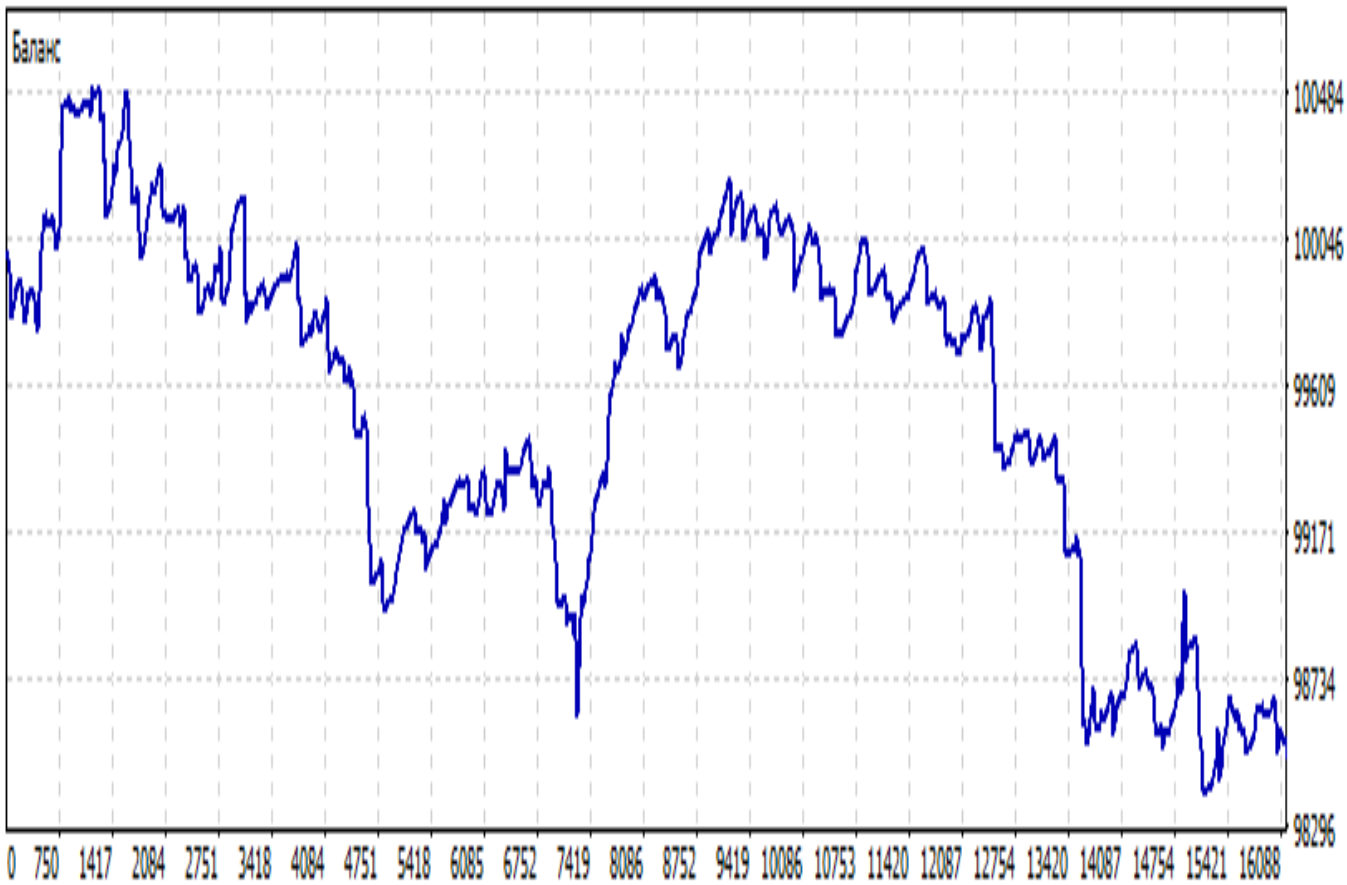

Рисунок 3.35 Баланс, $ / сделки для индикатора Адаптивное Скользящее Среднее

Таблица 9

Индикатор Стохастический Осциллятор

Чистая прибыль, $	-2 280.68
Общая прибыль, $	9 043.57
Общий убыток, $	11 324.25
Всего сделок	5850
Средняя прибыльная сделка, $	4.29
Прибыльные сделки, (% от всех)	2109 (36.05%)
Самая большая прибыльная сделка, $	34.15
Максимальное количество непрерывных выигрышей	108
Средний непрерывный выигрыш	26
Абсолютная просадка по средствам, $	3 755.86
Коэффициент Шарпа	-0.07
Матожидание выигрыша	-0.39
Максимальное количество непрерывных проигрышей	144
Средний непрерывный проигрыш	46
Убыточные сделки (% от всех):	3741 (63.95%)

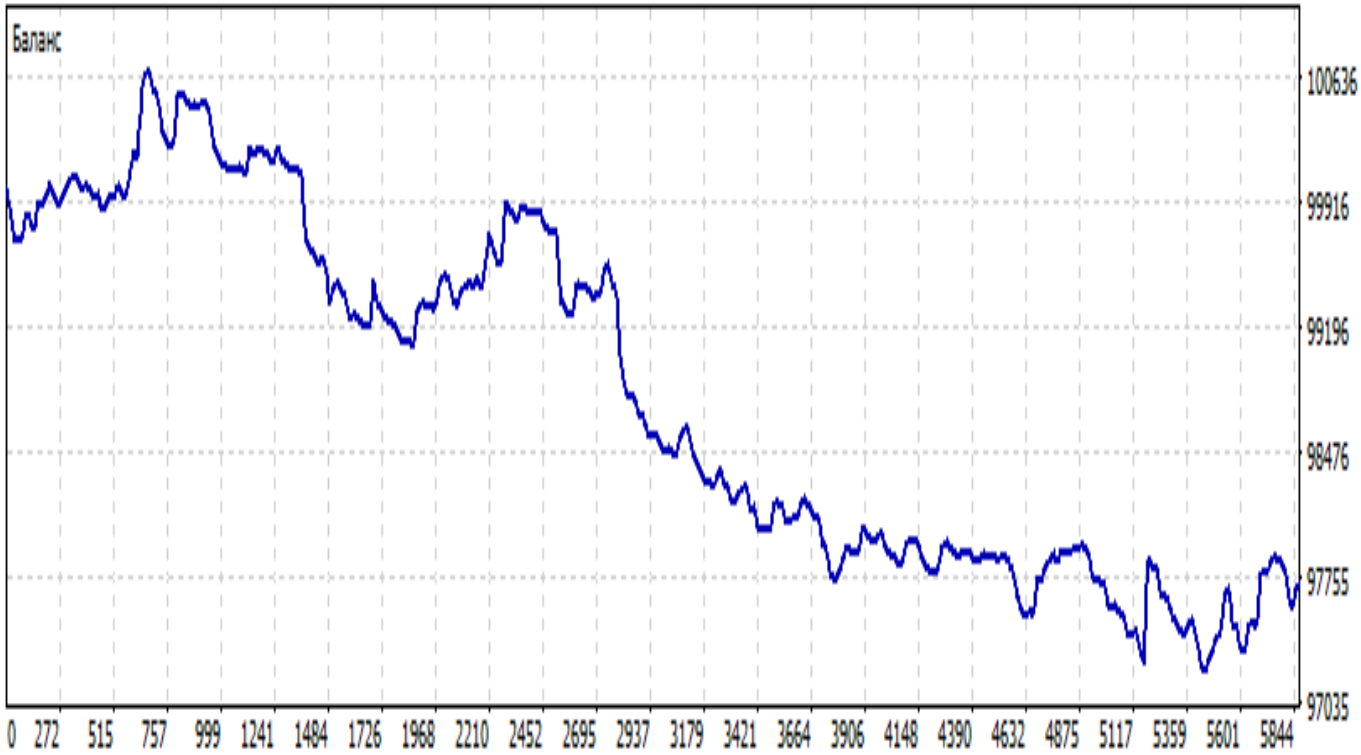

Рисунок 3.36 Баланс, $ / сделки для индикатора Стохастический Осциллятор

Таблица 10

Индикатор Схождение/Расхождение Скользящих Средних

Чистая прибыль, $	2 023.67
Общая прибыль, $	5 771.46
Общий убыток, $	3 747.79
Всего сделок	1224
Средняя прибыльная сделка, $	9.72
Прибыльные сделки, (% от всех)	594 (48.53%)
Самая большая прибыльная сделка, $	40.09
Максимальное количество непрерывных выигрышей	108
Средний непрерывный выигрыш	31
Абсолютная просадка по средствам, $	2 329.25
Коэффициент Шарпа	0.05
Матожидание выигрыша	1.65
Максимальное количество непрерывных проигрышей	90
Средний непрерывный проигрыш	33
Убыточные сделки (% от всех):	630

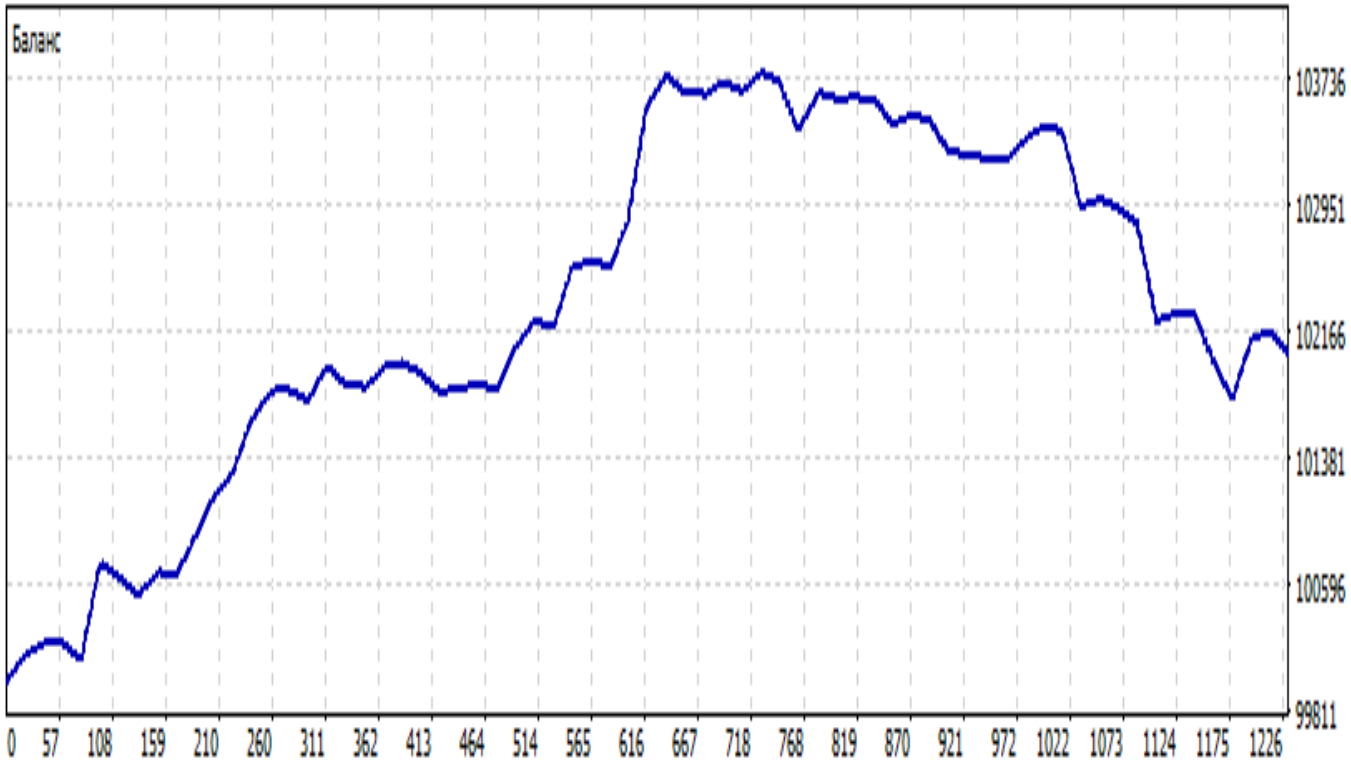

Рисунок 3.37 – Баланс, $ / сделки для индикатора Схождение/Расхождение Скользящих Средних

Таблица 11

Индикатор Скользящая Средняя

Чистая прибыль, $	2 122.81
Общая прибыль, $	12 760.79
Общий убыток, $	10 637.98
Всего сделок	8149
Средняя прибыльная сделка, $	2.64
Прибыльные сделки, (% от всех)	4828 (59.25%)
Самая большая прибыльная сделка, $	31.37
Максимальное количество непрерывных выигрышей	199
Средний непрерывный выигрыш	40
Абсолютная просадка по средствам, $	180.99
Коэффициент Шарпа	0.06
Матожидание выигрыша	0.26
Максимальное количество непрерывных проигрышей	90
Средний непрерывный проигрыш	27
Убыточные сделки (% от всех):	3321 (40.75%)

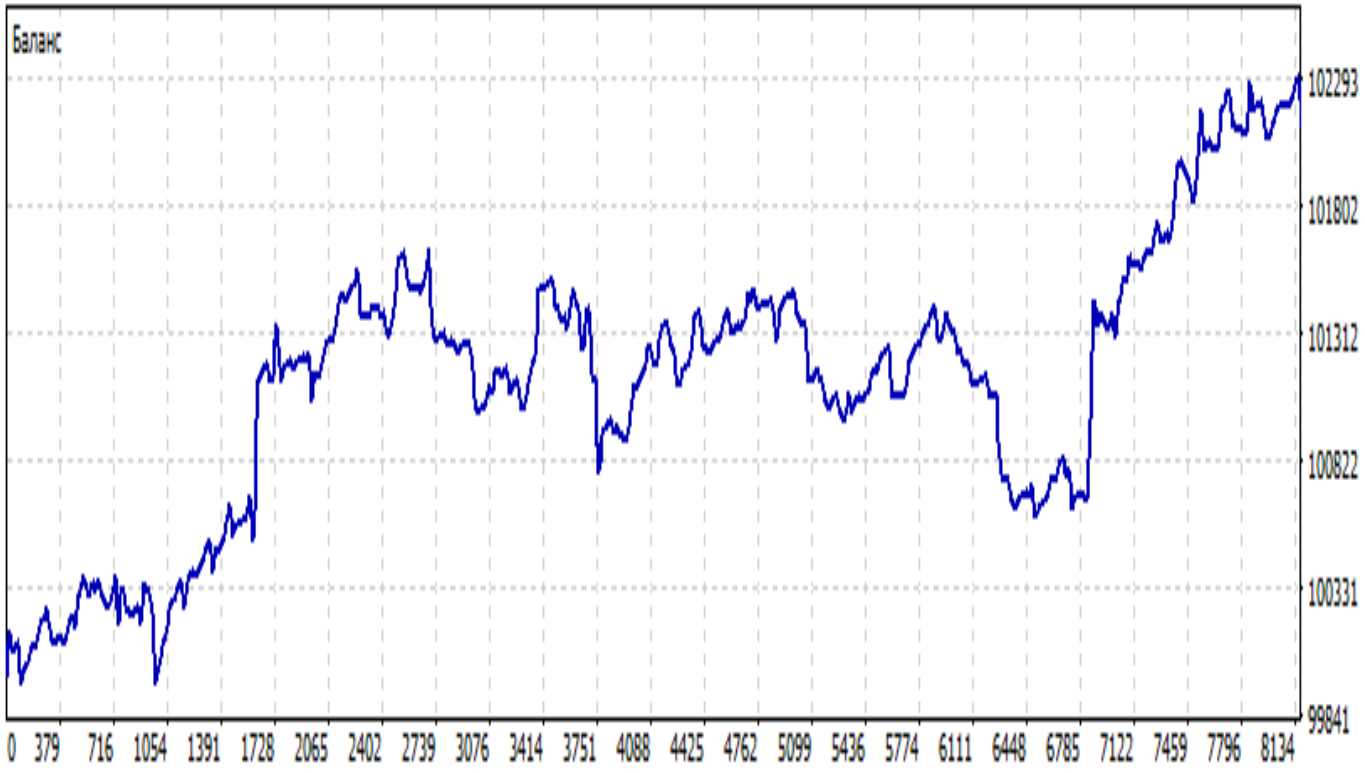

Рисунок 3.38 – Баланс, $ / сделки для индикатора Скользящая Средняя

3.3 Результаты алгоритма управления капиталом

Таблица 12

Голосование Большинства

Чистая прибыль, $	2 385.29
Общая прибыль, $	5 908.27
Общий убыток, $	3 522.98
Всего сделок	1476
Средняя прибыльная сделка, $	7.58
Прибыльные сделки, (% от всех)	779 (52.78%)
Самая большая прибыльная сделка, $	10.11
Максимальное количество непрерывных выигрышей	164
Средний непрерывный выигрыш	71
Абсолютная просадка по средствам, $	561.62
Коэффициент Шарпа	0.22
Матожидание выигрыша	1.62
Максимальное количество непрерывных проигрышей	40
Средний непрерывный проигрыш	31
Убыточные сделки (% от всех):	697 (47.22%)

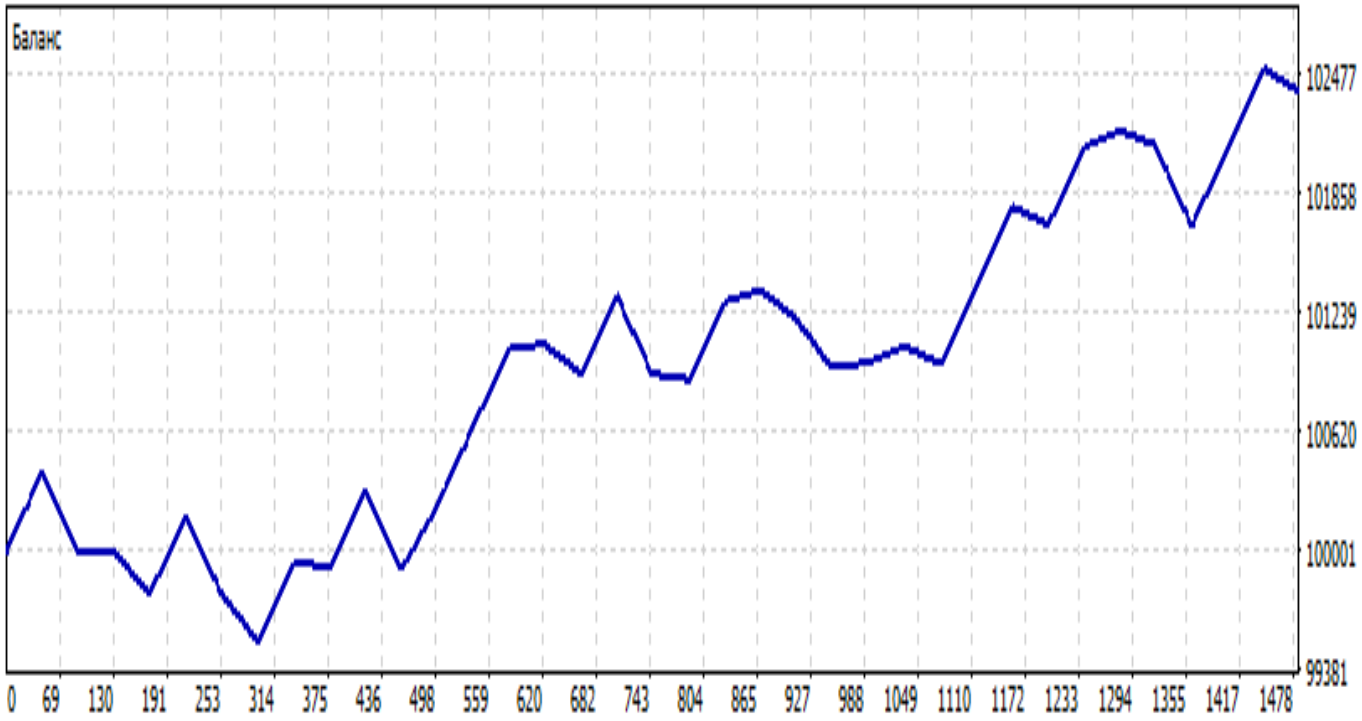

Рисунок 3.39 Баланс, $ / сделки для комбинации Голосование Большинства

Таблица 13

Правило К

Чистая прибыль, $	6 255.97
Общая прибыль, $	15 396.00
Общий убыток, $	9 140.03
Всего сделок	5048
Средняя прибыльная сделка, $	4.83
Прибыльные сделки, (% от всех)	3187 (63.13%)
Самая большая прибыльная сделка, $	10.52
Максимальное количество непрерывных выигрышей	217
Средний непрерывный выигрыш	80
Абсолютная просадка по средствам, $	26.65
Коэффициент Шарпа	0.23
Матожидание выигрыша	1.24
Максимальное количество непрерывных проигрышей	45
Средний непрерывный проигрыш	34
Убыточные сделки (% от всех):	1861 (36.87%)

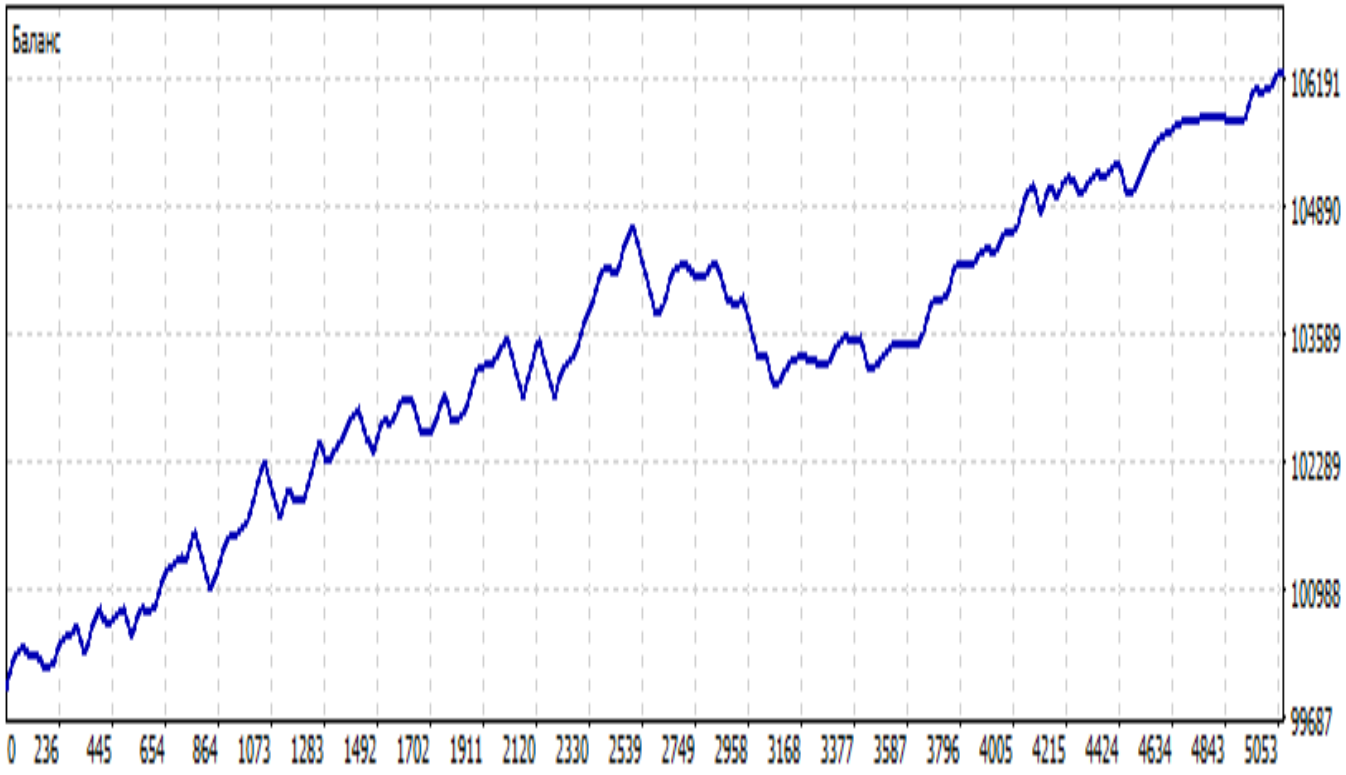

Рисунок 3.40 – Баланс, $ / сделки для комбинации сигналов по правилу К

3.4

Таблица 14

Правило B

Чистая прибыль, $	2 297.84
Общая прибыль, $	3 411.93
Общий убыток, $	1 114.09
Всего сделок	1395
Средняя прибыльная сделка, $	6.83
Прибыльные сделки, (% от всех)	907 (65.02%)
Самая большая прибыльная сделка, $	11.52
Максимальное количество непрерывных выигрышей	220
Средний непрерывный выигрыш	76
Абсолютная просадка по средствам, $	199.62
Коэффициент Шарпа	0.39
Матожидание выигрыша	1.65
Максимальное количество непрерывных проигрышей	43
Средний непрерывный проигрыш	32
Убыточные сделки (% от всех):	488 (34.98%)

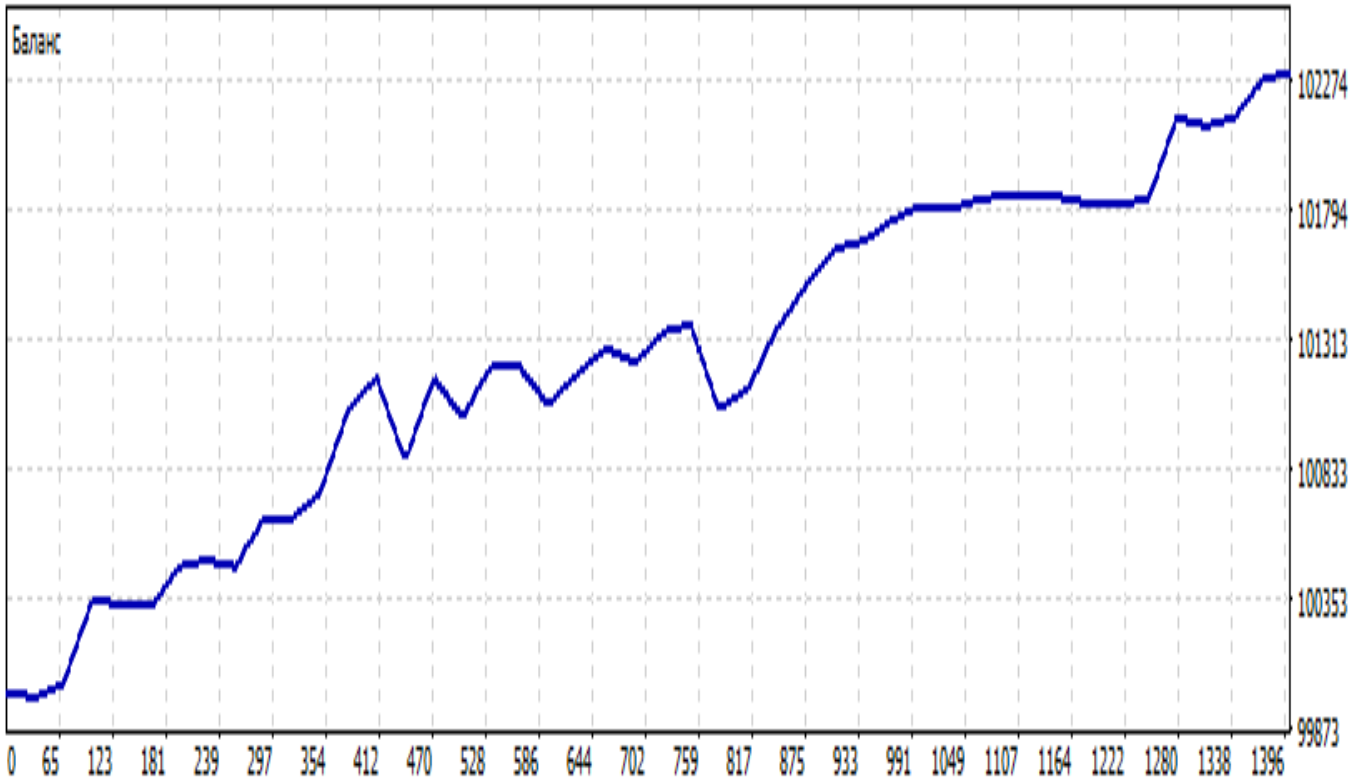

Рисунок 3.41 Баланс, $ / сделки для комбинации сигналов по Правилу B

3.5 Выводы по разделу

Результаты испытаний, проведенные с различными сигналами и их комбинацией, показывают, что использование АКМ не всегда увеличивает прибыль, однако уменьшает риски. Об этом свидетельствует Коэффициент Шарпа, показывающий соотношения дохода к принятым рискам. В ходе работы было выявлено, что АКМ, работающая с комбинацией сигналов по Правилу В, приносит прибыль соразмерную прибыли работы одиночного сигнала, однако коэффициент Шарпа превосходит в несколько раз лучшую одиночную модель. Что свидетельствует о меньшем риске при использовании АКМ.

ЗАКЛЮЧЕНИЕ

Успешная работа на финансовом рынке зависит от многих факторов, выбранной торговой стратегии и системы, психологии трейдера, его опытом и способностью быстро оценить меняющиеся условия торговли. Компьютерная программа, отслеживающая движение котировок на финансовом рынке, совершающая сделки, сопровождающая и закрывающая их, лишена усталости и способна работать круглосуточно, четко выполнять инструкции, заложенные в нее. Постоянно изменяющиеся условия рынка очень трудно описать, разработано достаточно методов и способов торговли.

В данной работе была представлена адаптивная комбинированная модель, которая приспосабливается к новым условиям. Адаптивная модель состоит из 11 базовых моделей. Был разработан алгоритм, запрограммирован модуль управления капиталом для платформы MetaTrader.

По результатам тестирования, проведенного в разделе 3, можно сделать вывод о том, что при работе адаптивной комбинированной модели снижаются риски, возникающие при торговле.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Распоряжение Правительства РФ от 29 декабря 2008 г. N 2043-р Об утверждении стратегии развития финансового рынка РФ на период до 2020 г.
2. Романов, В.П. Информационные технологии моделирования финансовых рынков. / В.П. Романов;– М.: Финансы и статистика, 2010. – 288 с.
3. Швагер, Д. Технический анализ / Д. Швагер; пер. с англ. А. Куницына, Б. Зуева. – М.: Альпина Бизнес Букс, 2007. – 805 с.
4. Торговая платформа MetaTrader 5. – Дата обновления: 26.07.2014. URL: https://www.metatrader5.com/ru/trading-platform (дата обращения: 05.02.2017).
5. Торговля и ордера в MetaTrader 5. – Дата обновления: 21.08.2014. URL: https://www.metatrader5.com/ru/trading-platform/trading (дата обращения: 08.02.2017).
6. Вайсман, Р. Механические торговые системы: Психология трейдинга и технический анализ. / Р. Вайсман; – М.: Альпина Паблишер, 2011. – 229 с.
7. Швагер, Д. Новые маги рынка: беседы с лучшими трейдерами Америки. / Д. Швагер; – М.: Альпина Паблишер, 2011. – 652 с.
8. Найман, Э. Малая энциклопедия трейдера. / Э. Найман; – М.: Альпина Паблишер, 2013. – 456 с.
9. Винс, Р. Математика управления капиталом: Методы анализа риска для трейдеров и портфельных менеджеров. / Р. Винс; – М.: Альпина Паблишер, 2012. – 400 с.
10. Тернер, Т. Краткосрочный трейдинг: Руководство для начинающих. / Т.Тернер; – М.: Альпина Паблишер, 2013. – 365 с.
11. Вилей, Д. Валютный и денежный рынок: Курс для начинающих. / Д. Вилей; – М.: Альпина Паблишер, 2012. – 344 с.
12. Флах, П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. / П. Флах; – М.: ДМК Пресс, 2015. – 400 с.
13. Галушкин, А.И. Нейронные сети: основы теории. / А.И. Галушкин; – М.: Горячая линия-Телеком, 2010. – 496 с.
14. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер.с польск.И.Д.Рудинского. / Д. Рудковская; – М.: Горячая линия-Телеком, 2013. – 384 с.
15. Горяинова, Е.Р. Прикладные методы анализа статистических данных. / Е.Р. Горяинова; – М.: Издательский дом Высшей школы экономики, 2012. – 310 с.
16. Замятин, А.В. Интеллектуальный анализ данных: учеб. пособие. / А.В. Замятин; – Томск: ТГУ, 2016. – 120 с.
17. Справочник по языку программирования MQL5 для клиентского терминала MetaTrader 5. – Дата обновления: 03.12.2015. URL: https://www.mql5.com/ru/docs (дата обращения: 28.01.2017).

18. METATRADER 5 Быстрый старт или краткий курс для начинающих. – Дата обновления: 14.09.2012. URL: https://www.mql5.com/ru/articles/496 (дата обращения: 30.01.2017).
19. Лукашин, Ю.П. Адаптивные методы краткосрочного прогнозирования. / Ю.П. Лукашин; – Москва: Статистика, 1979. – 254 с.
20. MetaTrader 5 Как создать торгового робота и не потерять время. – Дата обновления: 19.07.2012. URL: https://www.mql5.com/ru/articles/443 (дата обращения: 31.01.2017).
21. MetaTrader 5 - больше, чем можно представить! – Дата обновления: 01.06.2012. URL: https://www.mql5.com/ru/articles/384 (дата обращения: 31.01.2017).
22. Знакомство с MQL5: написание простого советника и индикатора – Дата обновления: 16.03.2010. URL: https://www.mql5.com/ru/articles/35 (дата обращения: 24.01.2017).

ПРИЛОЖЕНИЯ

Приложение 1 Руководство пользователя

Файл советника MoneyMaker.ex5 нужно поместить в папку MQL5/experts в каталоге данных вашего терминала.

Чтобы попасть в каталог данных, в терминале нажимаем Файл -> Открыть каталог данных. Демонстрация на рисунке 4.1

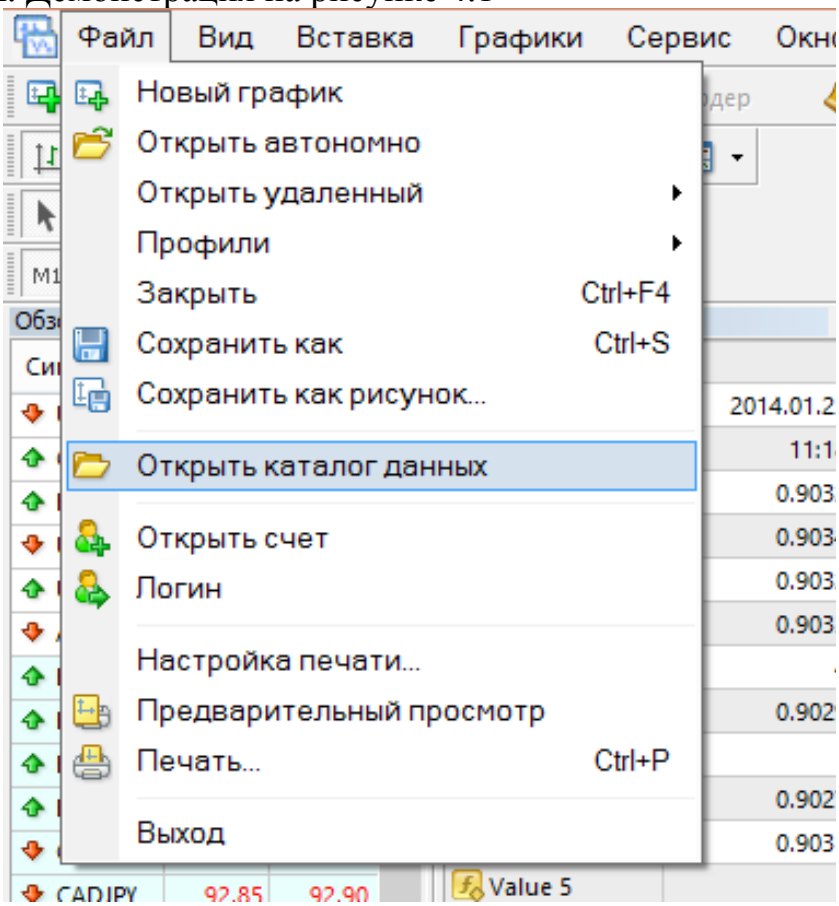

Рисунок 4.1 – Каталог данных

Откроется папка, в раздел MQL5 в каталог Experts копируем файлы советника. Закрываем папку, перезапускаем клиентский терминал MetaTrader.

Открываем торговый терминал, заходим Сервис->Настройки

Выбираем вкладку Советники и устанавливаем права для советника, «Разрешить автоматическую торговлю», «Разрешит импорт DLL», жмем ОК. Рисунок 4.2

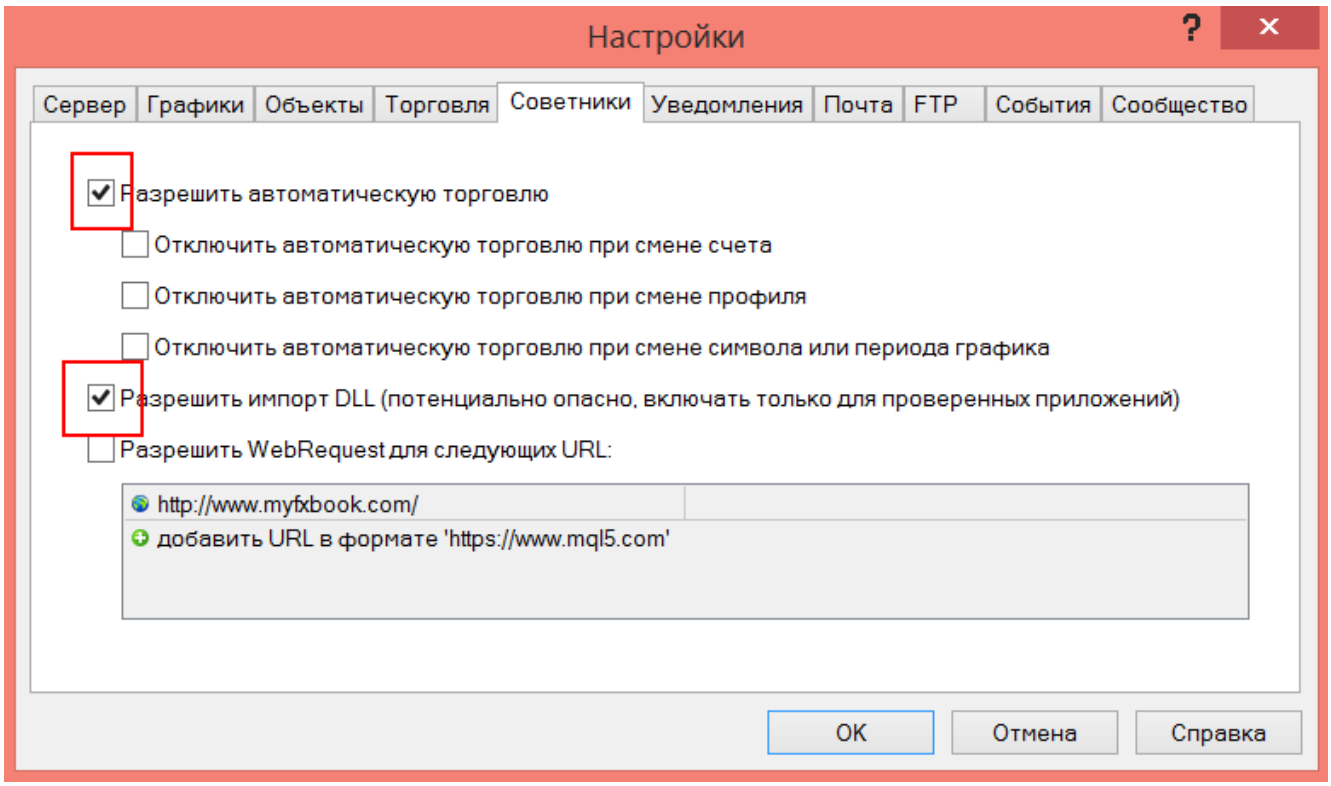

Рисунок 4.2 – Настройки терминала

В окне навигатора нажимаем плюсик напротив раздела Советники. Из выпавшего списка мышкой перетаскиваем нужный советник на заранее открытый график с валютной парой и таймфреймом, подходящими для работы эксперта.

Появляется окно настроек советника как на рисунке 4.3

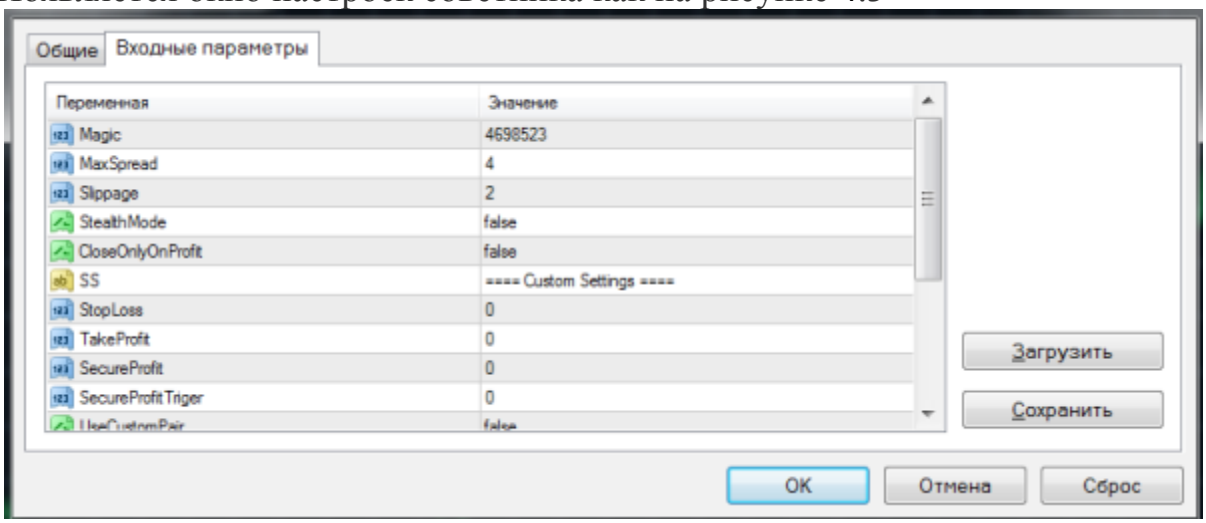

Рисунок 4.3 – Настройки Советника

После нажатия кнопки ОК советник начнет свою работу.

Приложение 2 Текст программы

Модуль управления капиталом

```
//+------------------------------------------------------------------+
//|                                                   MoneyMaker.mq5 |
//|                                                    Mayorov Maxim |
//|                                                mayorovmp@yandex.ru |
//+------------------------------------------------------------------+
#property copyright "Mayorov Maxim"
#property link      "mayorovmp@yandex.ru"
#property version   "1.00"

#include<Trade\AccountInfo.mqh>
#include<Trade\SymbolInfo.mqh>
#include<Trade\Trade.mqh>

CAccountInfo account;
CSymbolInfo symbol_info;//Класс символа торговой пары
CTrade  trade;//Класс для торговых операций

//+------------------------------------------------------------------+
//| Signals
//+------------------------------------------------------------------+
#include"Signals\MovingAverageIntersect.mqh"//+
#include"Signals\MacdIntersect.mqh"//+
#include"Signals\Stochastic.mqh"//+
#include"Signals\AMA.mqh"
#include"Signals\RSI.mqh"
#include"Signals\CCI.mqh"//+
#include"Signals\WPR.mqh"
#include"Signals\BANDS.mqh"
#include"Signals\StdDevChannel.mqh"
#include"Signals\Envelopes.mqh"
#include"Signals\Alligator.mqh"
//+------------------------------------------------------------------+
//| Global variables                                                 |
//+------------------------------------------------------------------+
#define SIGNALS_N 11
#define N 1000

ISignal *signals[SIGNALS_N];
OrderType predicts[SIGNALS_N][N];

double asks[N];//Цена продажи
double bids[N];//Цена покупки
int currentTick=0;
MqlTick last_tick;//Получение информации о котировках
//+------------------------------------------------------------------+
//|Инициализация
//+------------------------------------------------------------------+
int OnInit()
  {
   InitSignals();

   InitRulesB();

   InitTrade();
   Print("Init success");
   return(INIT_SUCCEEDED);
  }
```

```
 void InitSignals()
 {
   signals[0] = (new MovingAverageIntersect());
   signals[1] = (new MacdIntersect());
   signals[2] = (new Stochastic());
   signals[3] = (new AMA());
   signals[4] = (new RSI());
   signals[5] = (new CCI());
   signals[6] = (new WPR());
   signals[7] = (new BANDS());
   signals[8] = (new StdDevChannel());
   signals[9] = (new Envelopes());
   signals[10] = (new Alligator());
 }
//+------------------------------------------------------------------+
//| Вспомогательная функция сохранения состояния
//+------------------------------------------------------------------+
void RememberState()
  {
   SymbolInfoTick(Symbol(),last_tick);//Получение информации о текущем тике.
   asks[currentTick % N] = last_tick.ask;
   bids[currentTick % N] = last_tick.bid;
   for(int i=0; i<SIGNALS_N; i++)
     {
      OrderType predict=signals[i].calc();

      predicts[i][currentTick]=predict;
     }
   currentTick = (currentTick + 1)%N;
  }
//+------------------------------------------------------------------+
//|Голосование большинства
//+------------------------------------------------------------------+
int MajorityVote()
{
   int cntSell = 0, cntBuy = 0, cntWait = 0;
   int signalSell = -1, signalBuy = -1, signalWait = -1;
   for(int i=0; i<SIGNALS_N;i++)
   {
      if(predicts[i][(N+currentTick-1)%N] == BUY)
      {
         cntBuy++;
         signalBuy = i;
      }else if(predicts[i][(N+currentTick-1)%N] == SELL)
      {
         cntSell++;
         signalSell = i;
      }
      else
      {
         cntWait++;
         signalWait = i;
      }
   }
   if(cntSell > cntBuy)
      return signalSell;
   if(cntBuy > cntSell )
      return signalBuy;

   return signalWait;
}
```

```
//+------------------------------------------------------------------+
//|Поиск лучшего сигнала для Правила K
//+------------------------------------------------------------------+
#define K 10
double profits[SIGNALS_N];
int FindBestSignal()
{
   for(int i=0; i<SIGNALS_N;i++)
   {
      profits[i] = 0;
      for(int j = 1; j<= K;j++)
        {
         int oldTick=(N+currentTick-j)%N;
         if(predicts[i][oldTick]==BUY)
           {//Советник просигнализировал о покупке.
            //Купили по цене asks[i]. Продаем по цене last_tick.bid.
            profits[i]+=last_tick.bid-asks[oldTick];
           }
         if(predicts[i][oldTick]==SELL)
           {//Советник просигнализировал о продаже.
            //Продали по цене bids[i]. Покупаем по цене last_tick.ask.
            profits[i]+=last_tick.ask-bids[oldTick];
           }
        }
   }
   int bestSignal=0;
   for(int i=0; i<SIGNALS_N;i++)
      if(profits[i]>profits[bestSignal])
         bestSignal=i;
   return bestSignal;
}
//+------------------------------------------------------------------+
//| Правило K
//+------------------------------------------------------------------+
int RulesK()
  {
   if(currentTick<=K)//Не достаточно истории.
      return -1;

   int bestSignal = FindBestSignal();

   return bestSignal;
  }

//+------------------------------------------------------------------+
//| Правило B                                                        |
//+------------------------------------------------------------------+
int tau = 0;
double alfa = 0.5;
double B[SIGNALS_N];//
int RulesB()
  {
   if(currentTick <= tau)//Не достаточно истории.
      return -1;
   for(int i = 0;i < SIGNALS_N; i++)
   {
      double e = 0;
      if(predicts[i][currentTick - tau]==BUY)
           {//Советник просигнализировал о покупке.
            //Купили по цене asks[i]. Продаем по цене last_tick.bid.
            e = last_tick.bid-asks[currentTick - tau];
           }
```

```
         if(predicts[i][currentTick - tau]==SELL)
              {//Советник просигнализировал о продаже.
               //Продали по цене bids[i]. Покупаем по цене last_tick.ask.
               e = last_tick.ask-bids[currentTick - tau];
              }
         if(e > 0)
            e = 0;

         B[i] = (1 - alfa) * B[i] + alfa * e * e;
       }
       int bestSignal=0;
       for(int i=0; i<SIGNALS_N;i++)
          if(B[i] < B[bestSignal])
             bestSignal=i;
       return bestSignal;
      }

      void InitRulesB()
      {
       for(int i = 0;i < SIGNALS_N; i++)
          B[i] = 0;
       }
    //+------------------------------------------------------------------+
    //|Вспомогательная функция создания торгового ордера                 |
    //+------------------------------------------------------------------+
    OrderType lastOrder=WAIT;
    int openPos = 0;
    int maxOpenPos = 10;
    void MakeOrder(int signal)
      {
       if(signal == -1)return;

       if(predicts[signal][(N+currentTick-1)%N]==BUY)
         {
          if(lastOrder==SELL)
          {
             CloseAllPosition();
             openPos = 0;
          }
          if(openPos > maxOpenPos)
             return;
            if(!trade.Buy(0.01, NULL, 0.0, last_tick.ask - 0.01, last_tick.ask +
0.01))
           {
            //--- сообщим о неудаче
                                  Print("Метод   Buy()   потерпел   неудачу.   Код
возврата=",trade.ResultRetcode(),
                  ". Описание кода: ",trade.ResultRetcodeDescription());
           }
          else
           {
            openPos++;
            lastOrder=BUY;
           }

         }

       if(predicts[signal][(N+currentTick-1)%N]==SELL)
         {
          if(lastOrder==BUY)
          {
             CloseAllPosition();
```

```
            openPos = 0;
         }
         if(openPos > maxOpenPos)
            return;
           if(!trade.Sell(0.01, NULL, 0.0, last_tick.bid + 0.01, last_tick.ask -
0.01))
           {
            //--- сообщим о неудаче
                                  Print("Метод   Sell()   потерпел   неудачу.   Код
возврата=",trade.ResultRetcode(),
                  ". Описание кода: ",trade.ResultRetcodeDescription());
           }
         else
         {
            lastOrder=SELL;
            openPos++;
         }

        }
     }
   //+------------------------------------------------------------------+
   //| Инициализация торговли                                           |
   //+------------------------------------------------------------------+
   void InitTrade()
     {
   //---   установим   допустимое   проскальзывание   в   пунктах   при   совершении
покупки/продажи
      int deviation=10;
      trade.SetDeviationInPoints(deviation);
   //--- какую функцию использовать для торговли: true - OrderSendAsync(), false -
OrderSend()
      trade.SetAsyncMode(true);
   //---
     }

   //+------------------------------------------------------------------+
   //| Функция обработки тиков                                          |
   //+------------------------------------------------------------------+
   void OnTick()
     {
      RememberState();
      //int signal=RulesK();
      int signal = MajorityVote();
      //int signal=RulesB();
      MakeOrder(signal);
     }
   //+------------------------------------------------------------------+

   void OnDeinit(const int reason)
     {
      CloseAllPosition();
     }

   void CloseAllPosition()
      {
         while(trade.PositionClose(_Symbol));
      }
   //+------------------------------------------------------------------+
```

Интерфейс сигналов

```
//+------------------------------------------------------------------+
//|                                                      ISignal.mqh |
//|                                                   Mayorov Maxim |
//|                                              mayorovmp@yandex.ru |
//+------------------------------------------------------------------+
#property copyright "Mayorov Maxim"
#property link      "mayorovmp@yandex.ru"
#property version   "1.00"
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
enum OrderType{
   SELL = 1,   //Продажа
   WAIT = 0,   //Ожидание
   BUY = -1,   //Покупка
};
interface ISignal
  {
public:
                  OrderType calc();
  };
```

Сигналы

```
//+------------------------------------------------------------------+
//|                                                                  |
//|                                                   Mayorov Maxim |
//|                                              mayorovmp@yandex.ru |
//+------------------------------------------------------------------+
#property copyright "Mayorov Maxim"
#property link      "mayorovmp@yandex.ru"
#property version   "1.00"
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
#include"../ISignal.mqh"
class Alligator : public ISignal
  {
private:
   int h_al;
   double al1_buffer[], al2_buffer[], al3_buffer[], Close[];
public:
                   Alligator();
                  ~Alligator();
                  OrderType calc();
  };
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
Alligator ::Alligator()
  {
   h_al=iAlligator(Symbol(),Period(),13,0,8,0,5,0,MODE_SMMA,PRICE_MEDIAN);
  }
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
Alligator ::~Alligator()
  {
  }
//+------------------------------------------------------------------+
OrderType Alligator :: calc()
{
```

```
   OrderType sig=0;

   if(h_al==INVALID_HANDLE)
     {
      h_al=iAlligator(Symbol(),Period(),13,0,8,0,5,0,MODE_SMMA,PRICE_MEDIAN);
      return(0);
     }
   else
     {
      if(CopyBuffer(h_al,0,0,2,al1_buffer)<2)
         return(0);
      if(CopyBuffer(h_al,1,0,2,al2_buffer)<2)
         return(0);
      if(CopyBuffer(h_al,2,0,2,al3_buffer)<2)
         return(0);
      if(!ArraySetAsSeries(al1_buffer,true))
         return(0);
      if(!ArraySetAsSeries(al2_buffer,true))
         return(0);
      if(!ArraySetAsSeries(al3_buffer,true))
         return(0);
     }
//--- проводим проверку условия и устанавливаем значение для sig
   if(al3_buffer[1]>al2_buffer[1] && al2_buffer[1]>al1_buffer[1])
      sig=1;
   else if(al3_buffer[1]<al2_buffer[1] && al2_buffer[1]<al1_buffer[1])
      sig=-1;
   else sig=0;

//--- возвращаем торговый сигнал
   return(sig);
  }
//+------------------------------------------------------------------+
//|                                                          AMA.mqh |
//|                                                   Mayorov Maxim |
//|                                              mayorovmp@yandex.ru |
//+------------------------------------------------------------------+
#property copyright "Mayorov Maxim"
#property link      "mayorovmp@yandex.ru"
#property version   "1.00"
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
#include"../ISignal.mqh"

class AMA : public ISignal
  {
private:
   int h_ama;
   double ama_buffer[];
public:
                     AMA();
                    ~AMA();
                    OrderType calc();
  };
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
AMA::AMA()
  {
  h_ama=iAMA(Symbol(),Period(),9,2,30,0,PRICE_CLOSE);
  }
```

```
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
AMA::~AMA()
  {
  }
//+------------------------------------------------------------------+
OrderType AMA::calc()
{
   OrderType sig=0;

   if(h_ama==INVALID_HANDLE)
     {
      h_ama=iAMA(Symbol(),Period(),9,2,30,0,PRICE_CLOSE);
      return(0);
     }
   else
     {
      if(CopyBuffer(h_ama,0,0,3,ama_buffer)<3)
         return(0);
      if(!ArraySetAsSeries(ama_buffer,true))
         return(0);
     }
//--- проводим проверку условия и устанавливаем значение для sig
   if(ama_buffer[2]<ama_buffer[1])
      sig=1;
   else if(ama_buffer[2]>ama_buffer[1])
      sig=-1;
   else sig=0;

//--- возвращаем торговый сигнал
   return (sig);
}
//+------------------------------------------------------------------+
//|                                                        BANDS.mqh |
//|                                                  Mayorov Maxim |
//|                                              mayorovmp@yandex.ru |
//+------------------------------------------------------------------+
#property copyright "Mayorov Maxim"
#property link      "mayorovmp@yandex.ru"
#property version   "1.00"
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
#include"../ISignal.mqh"

class BANDS : public ISignal
  {
private:
   int h_bb;
   double bb1_buffer[], bb2_buffer[], Close[];

public:
                     BANDS();
                    ~BANDS();
                    OrderType calc();
  };
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
BANDS::BANDS()
  {
```

```
  h_bb=iBands(Symbol(),Period(),20,0,2,PRICE_CLOSE);
  }
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
BANDS::~BANDS()
  {
  }
//+------------------------------------------------------------------+
OrderType BANDS::calc()
  {
   OrderType sig=0;

   if(h_bb==INVALID_HANDLE)
     {
      h_bb=iBands(Symbol(),Period(),20,0,2,PRICE_CLOSE);
      return(0);
     }
   else
     {
      if(CopyBuffer(h_bb,1,0,2,bb1_buffer)<2)
         return(0);
      if(CopyBuffer(h_bb,2,0,2,bb2_buffer)<2)
         return(0);
      if(CopyClose(Symbol(),Period(),0,3,Close)<3)
         return(0);
      if(!ArraySetAsSeries(bb1_buffer,true))
         return(0);
      if(!ArraySetAsSeries(bb2_buffer,true))
         return(0);
      if(!ArraySetAsSeries(Close,true))
         return(0);
     }
//--- проводим проверку условия и устанавливаем значение для sig
   if(Close[2]<=bb2_buffer[1] && Close[1]>bb2_buffer[1])
      sig=1;
   else if(Close[2]>=bb1_buffer[1] && Close[1]<bb1_buffer[1])
      sig=-1;
   else sig=0;

//--- возвращаем торговый сигнал
   return(sig);
  }
//+------------------------------------------------------------------+
//|                                                          CI.mqh |
//|                                                  Mayorov Maxim |
//|                                              mayorovmp@yandex.ru |
//+------------------------------------------------------------------+
#property copyright "Mayorov Maxim"
#property link      "mayorovmp@yandex.ru"
#property version   "1.00"
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
#include"../ISignal.mqh"
class CCI : public ISignal
  {
private:
   int h_cci;
   double cci_buffer[];
public:
                     CCI();
```

```
                    ~CCI();
                    OrderType calc();
  };
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
CCI::CCI()
  {
  h_cci=iCCI(Symbol(),Period(),14,PRICE_TYPICAL);
  }
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
CCI::~CCI()
  {
  }
//+------------------------------------------------------------------+
OrderType CCI::calc()
  {
   OrderType sig=0;

   if(h_cci==INVALID_HANDLE)
     {
      h_cci=iCCI(Symbol(),Period(),14,PRICE_TYPICAL);
      return(0);
     }
   else
     {
      if(CopyBuffer(h_cci,0,0,3,cci_buffer)<3)
         return(0);

      if(!ArraySetAsSeries(cci_buffer,true))
         return(0);
     }
//--- проводим проверку условия и устанавливаем значение для sig
   if(cci_buffer[2]<-100 && cci_buffer[1]>-100)
      sig=1;
   else if(cci_buffer[2]>100 && cci_buffer[1]<100)
      sig=-1;
   else sig=0;

//--- возвращаем торговый сигнал
   return(sig);
  }
//+------------------------------------------------------------------+
//|                                                   Envelopes.mqh |
//|                                                   Mayorov Maxim |
//|                                             mayorovmp@yandex.ru |
//+------------------------------------------------------------------+
#property copyright "Mayorov Maxim"
#property link      "mayorovmp@yandex.ru"
#property version   "1.00"
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
#include"../ISignal.mqh"
class Envelopes : public ISignal
  {
private:
   int h_env;
   double env1_buffer[], env2_buffer[], Close[];
public:
```

```
                     Envelopes();
                    ~Envelopes();
                    OrderType calc();
  };
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
Envelopes::Envelopes()
  {
  h_env=iEnvelopes(Symbol(),Period(),28,0,MODE_SMA,PRICE_CLOSE,0.1);
  }
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
Envelopes::~Envelopes()
  {
  }
//+------------------------------------------------------------------+
OrderType Envelopes::calc()
  {
   OrderType sig=0;

   if(h_env==INVALID_HANDLE)
     {
      h_env=iEnvelopes(Symbol(),Period(),28,0,MODE_SMA,PRICE_CLOSE,0.1);
      return(0);
     }
   else
     {
      if(CopyBuffer(h_env,0,0,2,env1_buffer)<2)
         return(0);
      if(CopyBuffer(h_env,1,0,2,env2_buffer)<2)
         return(0);
      if(CopyClose(Symbol(),Period(),0,3,Close)<3)
         return(0);
      if(!ArraySetAsSeries(env1_buffer,true))
         return(0);
      if(!ArraySetAsSeries(env2_buffer,true))
         return(0);
      if(!ArraySetAsSeries(Close,true))
         return(0);
     }
//--- проводим проверку условия и устанавливаем значение для sig
   if(Close[2]<=env2_buffer[1] && Close[1]>env2_buffer[1])
      sig=1;
   else if(Close[2]>=env1_buffer[1] && Close[1]<env1_buffer[1])
      sig=-1;
   else sig=0;

//--- возвращаем торговый сигнал
   return(sig);
  }
//+------------------------------------------------------------------+
//|                                                MacdIntersect.mqh |
//|                                                   Mayorov Maxim |
//|                                              mayorovmp@yandex.ru |
//+------------------------------------------------------------------+
#property copyright "Mayorov Maxim"
#property link      "mayorovmp@yandex.ru"
#property version   "1.00"
//+------------------------------------------------------------------+
//|                                                                  |
```

```
//+------------------------------------------------------------------+
#include"../ISignal.mqh"
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
class MacdIntersect : public ISignal
  {
private:
   int h_macd;
   double macd1_buffer[],macd2_buffer[];

public:
                     MacdIntersect();
                    ~MacdIntersect();
                    OrderType calc();
  };
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
MacdIntersect::MacdIntersect()
  {
   h_macd=iMACD(Symbol(),Period(),12,26,9,PRICE_CLOSE);
  }
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
MacdIntersect::~MacdIntersect()
  {
  }
//+------------------------------------------------------------------+
OrderType MacdIntersect::calc()
{
   OrderType sig=0;

   if(h_macd==INVALID_HANDLE)
     {
      h_macd=iMACD(Symbol(),Period(),12,26,9,PRICE_CLOSE);
      return OrderType(0);
     }
   else
     {
      if(CopyBuffer(h_macd,0,0,2,macd1_buffer)<2)
         return OrderType(0);
      if(CopyBuffer(h_macd,1,0,3,macd2_buffer)<3)
         return OrderType(0);
      if(!ArraySetAsSeries(macd1_buffer,true))
         return OrderType(0);
      if(!ArraySetAsSeries(macd2_buffer,true))
         return OrderType(0);
     }

//--- проводим проверку условия и устанавливаем значение для sig
   if(macd2_buffer[2]>macd1_buffer[1] && macd2_buffer[1]<macd1_buffer[1])
      sig=1;
   else if(macd2_buffer[2]<macd1_buffer[1] && macd2_buffer[1]>macd1_buffer[1])
      sig=-1;
   else sig=0;

//--- возвращаем торговый сигнал
   return (sig);

}
```

```
//+------------------------------------------------------------------+
//|                                                    Signal_1.mqh |
//|                                                  Mayorov Maxim |
//|                                             mayorovmp@yandex.ru |
//+------------------------------------------------------------------+
#property copyright "Mayorov Maxim"
#property link      "mayorovmp@yandex.ru"
#property version   "1.00"
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
#include"../ISignal.mqh"
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
class MovingAverageIntersect:public ISignal
  {
private:

   int h_ma1,h_ma2;
   double ma1_buffer[],ma2_buffer[];

public:
                     MovingAverageIntersect();
                     ~MovingAverageIntersect();
                     OrderType calc();
  };
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
MovingAverageIntersect::MovingAverageIntersect()
  {
   h_ma1=iMA(Symbol(),Period(),8,0,MODE_SMA,PRICE_CLOSE);
   h_ma2=iMA(Symbol(),Period(),16,0,MODE_SMA,PRICE_CLOSE);
  }
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
MovingAverageIntersect::~MovingAverageIntersect()
  {
  }
//+------------------------------------------------------------------+
OrderType MovingAverageIntersect::calc()
 {
//--- ноль означает отсутствие сигнала
   OrderType sig=0;

//--- проверим хендлы индикаторов
   if(h_ma1==INVALID_HANDLE)//--- если хэндл невалидный
     {
      //--- создадим его снова
      h_ma1=iMA(Symbol(),Period(),8,0,MODE_SMA,PRICE_CLOSE);
      //--- выходим из функции
      return(0);
     }
   else //--- если хэндл валидный
     {
      //--- копируем значения из индикатора в массив
          if(CopyBuffer(h_ma1,0,0,3,ma1_buffer)<3)  //--- и если данных меньше
требуемых
           //--- выходим из функции
           return(0);
```

```
         //--- зададим индексацию в массиве как таймсерию
         if(!ArraySetAsSeries(ma1_buffer,true))
            //--- в случае ошибки индексации выходим из функции
            return(0);
        }

      if(h_ma2==INVALID_HANDLE)//--- если хэндл невалидный
        {
         //--- создадим его снова
         h_ma2=iMA(Symbol(),Period(),16,0,MODE_SMA,PRICE_CLOSE);
         //--- выходим из функции
         return(0);
        }
      else //--- если хэндл валидный
        {
         //--- копируем значения из индикатора в массив
            if(CopyBuffer(h_ma2,0,0,2,ma2_buffer)<2) //--- и если данных меньше
требуемых
            //--- выходим из функции
            return(0);
         //--- зададим индексацию в массиве как таймсерию
         if(!ArraySetAsSeries(ma1_buffer,true))
            //--- в случае ошибки индексации выходим из функции
            return(0);
        }

   //--- проводим проверку условия и устанавливаем значение для sig
      if(ma1_buffer[2]<ma2_buffer[1] && ma1_buffer[1]>ma2_buffer[1])
         sig=1;
      else if(ma1_buffer[2]>ma2_buffer[1] && ma1_buffer[1]<ma2_buffer[1])
         sig=-1;
      else sig=0;

   //--- возвращаем торговый сигнал
      return (sig);
     }
   //+------------------------------------------------------------------+
   //|                                                          RSI.mqh |
   //|                                                   Mayorov Maxim |
   //|                                                mayorovmp@yandex.ru |
   //+------------------------------------------------------------------+
   #property copyright "Mayorov Maxim"
   #property link      "mayorovmp@yandex.ru"
   #property version   "1.00"
   //+------------------------------------------------------------------+
   //|                                                                  |
   //+------------------------------------------------------------------+
   #include"../ISignal.mqh"
   class RSI : public ISignal
     {
   private:
      int h_rsi;
      double rsi_buffer[];
   public:
                        RSI();
                       ~RSI();
                       OrderType calc();
     };
   //+------------------------------------------------------------------+
   //|                                                                  |
   //+------------------------------------------------------------------+
   RSI::RSI()
```

```
  {
  h_rsi=iRSI(Symbol(),Period(),14,PRICE_CLOSE);
  }
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
RSI::~RSI()
  {
  }
//+------------------------------------------------------------------+
OrderType RSI::calc()
{
OrderType sig=0;

   if(h_rsi==INVALID_HANDLE)
     {
      h_rsi=iRSI(Symbol(),Period(),14,PRICE_CLOSE);
      return(0);
     }
   else
     {
      if(CopyBuffer(h_rsi,0,0,3,rsi_buffer)<3)
         return(0);

      if(!ArraySetAsSeries(rsi_buffer,true))
         return(0);
     }
//--- проводим проверку условия и устанавливаем значение для sig
   if(rsi_buffer[2]<30 && rsi_buffer[1]>30)
      sig=1;
   else if(rsi_buffer[2]>70 && rsi_buffer[1]<70)
      sig=-1;
   else sig=0;

//--- возвращаем торговый сигнал
   return(sig);
}
//+------------------------------------------------------------------+
//|                                              StdDevChannel.mqh |
//|                                                  Mayorov Maxim |
//|                                             mayorovmp@yandex.ru |
//+------------------------------------------------------------------+
#property copyright "Mayorov Maxim"
#property link      "mayorovmp@yandex.ru"
#property version   "1.00"
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
#include"../ISignal.mqh"
class StdDevChannel : public ISignal
  {
private:
   int h_sdc;
   double sdc1_buffer[], sdc2_buffer[], Close[];
public:
                     StdDevChannel();
                    ~StdDevChannel();
                    OrderType calc();
  };
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
```

```
   StdDevChannel::StdDevChannel()
     {
      h_sdc=iCustom(Symbol(),Period(),"Examples\\StdDev",14,0,MODE_SMA,PRICE_CLOSE
,2.0);
     }
   //+------------------------------------------------------------------+
   //|                                                                  |
   //+------------------------------------------------------------------+
   StdDevChannel::~StdDevChannel()
     {
     }
   //+------------------------------------------------------------------+
   OrderType StdDevChannel:: calc()
     {
      OrderType sig=0;

      if(h_sdc==INVALID_HANDLE)
        {
         h_sdc=iCustom(Symbol(),Period(),"Examples\\StdDev",14,0,MODE_SMA,PRICE_CL
OSE,2.0);
         return(0);
        }
      else
        {
         if(CopyBuffer(h_sdc,0,0,2,sdc1_buffer)<2)
            return(0);
         if(CopyBuffer(h_sdc,1,0,2,sdc2_buffer)<2)
            return(0);
         if(CopyClose(Symbol(),Period(),0,3,Close)<3)
            return(0);
         if(!ArraySetAsSeries(sdc1_buffer,true))
            return(0);
         if(!ArraySetAsSeries(sdc2_buffer,true))
            return(0);
         if(!ArraySetAsSeries(Close,true))
            return(0);
        }
   //--- проводим проверку условия и устанавливаем значение для sig
      if(Close[2]<=sdc2_buffer[1] && Close[1]>sdc2_buffer[1])
         sig=1;
      else if(Close[2]>=sdc1_buffer[1] && Close[1]<sdc1_buffer[1])
         sig=-1;
      else sig=0;

   //--- возвращаем торговый сигнал
      return(sig);
     }
   //+------------------------------------------------------------------+
   //|                                                  Stochastic.mqh |
   //|                                                   Mayorov Maxim |
   //|                                                mayorovmp@yandex.ru |
   //+------------------------------------------------------------------+
   #property copyright "Mayorov Maxim"
   #property link      "mayorovmp@yandex.ru"
   #property version   "1.00"
   //+------------------------------------------------------------------+
   //|                                                                  |
   //+------------------------------------------------------------------+
   #include"../ISignal.mqh"
   //+------------------------------------------------------------------+
   //|                                                                  |
   //+------------------------------------------------------------------+
```

```
class Stochastic : public ISignal
  {
private:
   double stoh_buffer[];
   int h_stoh;
public:
                     Stochastic();
                    ~Stochastic();
                    OrderType calc();
  };
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
Stochastic::Stochastic()
  {
   h_stoh=iStochastic(Symbol(),Period(),5,3,3,MODE_SMA,STO_LOWHIGH);
  }
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
Stochastic::~Stochastic()
  {
  }
//+------------------------------------------------------------------+
OrderType Stochastic:: calc()
{
OrderType sig=0;

   if(h_stoh==INVALID_HANDLE)
     {
      h_stoh=iStochastic(Symbol(),Period(),5,3,3,MODE_SMA,STO_LOWHIGH);
      return OrderType(0);
     }
   else
     {
      if(CopyBuffer(h_stoh,0,0,3,stoh_buffer)<3)
         return OrderType(0);

      if(!ArraySetAsSeries(stoh_buffer,true))
         return OrderType(0);
     }
//--- проводим проверку условия и устанавливаем значение для sig
   if(stoh_buffer[2]<20 && stoh_buffer[1]>20)
      sig=1;
   else if(stoh_buffer[2]>80 && stoh_buffer[1]<80)
      sig=-1;
   else sig=0;

//--- возвращаем торговый сигнал
   return (sig);
}



//+------------------------------------------------------------------+
//|                                                         WPR.mqh |
//|                                                   Mayorov Maxim |
//|                                              mayorovmp@yandex.ru |
//+------------------------------------------------------------------+
#property copyright "Mayorov Maxim"
#property link      "mayorovmp@yandex.ru"
#property version   "1.00"
```

```
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
#include"../ISignal.mqh"
class WPR : public ISignal
  {
private:
   int h_wpr;
   double wpr_buffer[];
public:
                     WPR();
                    ~WPR();
                    OrderType calc();
  };
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
WPR::WPR()
  {
  h_wpr=iWPR(Symbol(),Period(),14);
  }
//+------------------------------------------------------------------+
//|                                                                  |
//+------------------------------------------------------------------+
WPR::~WPR()
  {
  }
//+------------------------------------------------------------------+
OrderType WPR::calc()
  {
   OrderType sig=0;

   if(h_wpr==INVALID_HANDLE)
     {
      h_wpr=iWPR(Symbol(),Period(),14);
      return(0);
     }
   else
     {
      if(CopyBuffer(h_wpr,0,0,3,wpr_buffer)<3)
         return(0);

      if(!ArraySetAsSeries(wpr_buffer,true))
         return(0);
     }
//--- проводим проверку условия и устанавливаем значение для sig
   if(wpr_buffer[2]<-80 && wpr_buffer[1]>-80)
      sig=1;
   else if(wpr_buffer[2]>-20 && wpr_buffer[1]<-20)
      sig=-1;
   else sig=0;

//--- возвращаем торговый сигнал
   return(sig);
  }
```