System Architecture & Documentation

Complete technical overview of the CycleTrader market analysis platform

Stock Market Signal Processing System Documentation

The Bazaar

Course: BCS Integrated Calculus (MTH 1202)

Semester 2, 2025

Faculty of Computing & Informatics
Department of Computer Science
Mbarara University of Science and Technology

Generated on April 26, 2025 at 14:56

Declaration and Approval

Declaration

We, the undersigned members of The Bazaar, declare that this project documentation is our original work. All sources used have been properly acknowledged and referenced. This work has not been submitted for any other academic award.

Group Leader

Name: Amanya Peter
Signature: _______________
Date: _______________

Project Supervisor

Name: Mr. Buri Gershom
Signature: _______________
Date: _______________

Approval

This project documentation has been submitted with the approval of the project supervisor.

Head of Department

Name: _______________
Signature: _______________
Date: _______________

Acknowledgements

We would like to express our sincere gratitude to:

Academic Support

- **Mr. Buri Gershom** - For his invaluable guidance, supervision, and technical expertise throughout the project development
- The Faculty of Computing & Informatics at Mbarara University of Science and Technology
- The Department of Computer Science for providing the necessary resources and support

Technical Resources

- The open-source community for providing essential libraries and tools
- Financial data providers for access to market information
- Documentation contributors and technical writers

Special Thanks

- Our fellow students for their feedback and testing support
- The university administration for their institutional support
- Our families for their continuous encouragement and support

Abstract

The Stock Market Signal Processing Web Application is a sophisticated platform that leverages Fast Fourier Transform (FFT) algorithms to analyze stock market data and identify cyclic patterns. This system provides traders and analysts with powerful tools for detecting market cycles, generating trading recommendations, and analyzing market sentiment. By combining advanced signal processing techniques with an intuitive user interface, the application makes complex technical analysis accessible to users of all experience levels.

Problem Statement

Traditional technical analysis methods often struggle to identify hidden market cycles and patterns that could provide valuable trading insights. Traders and investors face several challenges:

- Difficulty in identifying underlying market cycles objectively
- Lack of quantitative tools for cycle-based trading decisions
- Challenge in combining technical analysis with sentiment indicators
- Need for automated, data-driven trading recommendations

Objectives

Primary Objectives

- Develop a robust system for detecting market cycles using FFT analysis
- Create accurate trading signals based on cycle analysis
- Integrate market sentiment analysis with technical indicators
- Provide an intuitive interface for complex technical analysis

Technical Objectives

- Implement efficient FFT algorithms for real-time analysis
- Develop accurate cycle detection methodologies
- Create reliable trading signal generation systems
- Build comprehensive visualization tools

User Experience Objectives

- Simplify complex technical analysis for all user levels
- Provide clear, actionable trading recommendations
- Enable easy data import and analysis
- Generate comprehensive, understandable reports

Literature Review

Signal Processing in Financial Markets

The application of signal processing techniques to financial markets has been extensively studied. Ehlers (2004) pioneered the use of digital filters and FFT analysis in technical trading, demonstrating how these tools can reveal hidden market cycles. Murphy (1999) established the foundational principles of technical analysis that remain relevant to modern algorithmic trading approaches.

Fourier Transform Applications

Smith (1997) provided comprehensive coverage of FFT algorithms and their applications in signal processing, which has been adapted for financial time series analysis. Oppenheim and Schafer (2009) detailed the mathematical foundations of discrete-time signal processing that underpin our cycle detection methodology.

Market Efficiency and Cycles

Lo's (2004) Adaptive Markets Hypothesis provides a theoretical framework supporting the existence of market cycles, suggesting that market efficiency varies over time and adapts to changing conditions. This supports our approach of dynamic cycle detection and analysis.

Market Sentiment Analysis

Recent research has shown strong correlations between market sentiment and price movements. Johnson et al. (2003) explored the complexity of financial markets and how sentiment indicators can provide early signals of market regime changes.

Gaps in Existing Systems

- Limited integration of signal processing with traditional technical analysis
- Lack of user-friendly interfaces for complex mathematical tools
- Insufficient combination of sentiment analysis with cycle detection
- Need for real-time processing and analysis capabilities

Our Contribution

This system addresses these gaps by providing an integrated platform that combines:

- Advanced FFT-based cycle detection algorithms
- Intuitive visualization of complex mathematical concepts

- Real-time sentiment analysis integration
- Automated trading recommendations based on multiple indicators

Existing Systems Analysis

Current Market Solutions

- **TradingView**: Offers traditional technical analysis tools but lacks advanced cycle detection
- **MetaTrader**: Focuses on forex trading with basic technical indicators
- **ThinkOrSwim**: Provides comprehensive charting but limited signal processing capabilities
- **Wave59 PRO2**: Implements basic cycle analysis without FFT integration

Limitations of Existing Systems

- Limited or no signal processing capabilities
- Lack of integrated sentiment analysis
- Complex interfaces requiring extensive training
- High cost of access to advanced features
- Limited customization options

System Uniqueness

Key Differentiators

- **Advanced Signal Processing**: First-of-its-kind implementation of FFT for market cycle detection
- **Integrated Approach**: Seamless combination of technical analysis, sentiment analysis, and cycle detection
- **Accessibility**: Complex mathematical concepts presented through an intuitive interface
- **Real-time Processing**: Instant analysis of market data and sentiment
- **Cost-effective Solution**: Professional-grade analysis tools at an accessible price point

Innovative Features

- **Multi-cycle Analysis**: Detection and analysis of multiple concurrent market cycles
- **Adaptive Algorithms**: Self-adjusting analysis based on market conditions
- **Comprehensive Visualization**: Interactive charts and intuitive visual representations
- **Portfolio Integration**: Cycle analysis across multiple assets simultaneously
- **Custom Data Support**: Analysis of user-provided historical data

Future Advantages

- Machine learning integration for improved cycle detection
- Advanced portfolio optimization based on cycle analysis
- Enhanced mobile accessibility
- Integration with additional data sources
- Expanded API capabilities for third-party integration

Methodology

1. Signal Processing Approach

Our methodology employs advanced signal processing techniques to analyze financial market data:

- **Data Preprocessing**: Raw price data is cleaned, normalized, and detrended
- **Fast Fourier Transform**: Price data is transformed into frequency domain
- **Cycle Detection**: Dominant frequencies are identified and analyzed
- **Signal Generation**: Trading signals are generated based on cycle analysis

2. Mathematical Foundation

The core mathematical components include:

- **Fourier Transform**: X(k) = Σ[n=0 to N-1] x(n) * e^(-j2πkn/N)
- **Amplitude Calculation**: |X(k)| = sqrt(real(X(k))^2 + imag(X(k))^2)
- **Phase Calculation**: φ(k) = atan2(imag(X(k)), real(X(k)))
- **Period Conversion**: T = N/k (where N is data length, k is frequency index)

3. Cycle Analysis Process

1. **Data Acquisition**
    - CSV file upload for historical data
    - Real-time API fetching for live data
    - Data validation and cleaning
2. **Signal Processing**
    - Application of FFT algorithm

  - Frequency spectrum analysis
  - Dominant cycle identification
3. **Pattern Recognition**
   - Cycle strength calculation
   - Phase analysis
   - Pattern validation

4. Decision Engine

Trading signals are generated through:

- **Multi-cycle Analysis**: Combining signals from multiple cycles
- **Strength Weighting**: Stronger cycles given higher importance
- **Confluence Detection**: Identifying areas where multiple cycles align
- **Confidence Calculation**: Based on signal clarity and cycle strength

5. Validation Process

- **Historical Backtesting**: Testing against historical data
- **Statistical Validation**: Ensuring statistical significance
- **Performance Metrics**: Measuring accuracy and reliability
- **Continuous Refinement**: Ongoing improvement of algorithms

Implementation Details

Core Algorithms

1. Fast Fourier Transform (FFT)

Our implementation uses the Fast Fourier Transform algorithm to decompose price data into its frequency components:

- **Purpose:** Convert time-domain price data into frequency-domain representation
- **Implementation:** Using SciPy's FFT module with Hann window function to reduce spectral leakage
- **Formula:** X(k) = Σ[n=0 to N-1] x(n) * e^(-j2πkn/N)
- **Application:** Identifies dominant cycle frequencies in price movements

2. Cycle Detection Algorithm

Custom peak detection algorithm for identifying significant market cycles:

- **Input:** FFT frequency spectrum
- **Process:**
  - Filter frequencies to trading-relevant range (2-252 days)
  - Calculate relative strength of each frequency component
  - Apply strength threshold (default 10%) to identify significant cycles
  - Calculate current phase position within each cycle
- **Output:** List of dominant cycles with period, strength, and phase

3. Trading Signal Generation

Decision engine algorithm for generating trading recommendations:

- **Inputs:**
  - Detected cycles and their current phases
  - Relative strength of each cycle
  - Recent price momentum
- **Process:**
  - Weight signals by cycle strength
  - Calculate days to next peak/trough for each cycle
  - Aggregate weighted signals for final recommendation
- **Output:** BUY/SELL/HOLD recommendation with confidence level

4. Sentiment Analysis Algorithm

Natural language processing approach for market sentiment detection:

- **Input:** Financial news text from multiple sources
- **Process:**
  - Keyword frequency analysis using predefined sentiment dictionaries
  - Weighted scoring of bullish/bearish/neutral terms
  - Normalization to 0-100 scale
- **Output:** Market sentiment score and mood classification

Data Processing Pipeline

The system processes data through the following stages:

1. Data acquisition and cleaning
2. FFT analysis for cycle detection
3. Trading signal generation
4. Visualization creation
5. Result storage and presentation

Market Sentiment Analysis

Sentiment analysis involves:

- Web scraping of financial news
- Keyword-based sentiment scoring
- Real-time sentiment updates

System Architecture

Core Components

- Data Processing Engine (FFT Analysis)
- Trading Recommendation System
- Market Sentiment Analysis
- Portfolio Management
- Visualization System

Technology Stack

- Backend: Python Flask, SQLAlchemy, NumPy, SciPy
- Frontend: Bootstrap 5, Plotly.js, Custom JavaScript
- Database: PostgreSQL with JSON support
- APIs: yfinance for market data

Features and Functionality

Functional Features

- Data Input Processing
  - CSV file upload capability
  - Real-time stock data fetching via API
  - Data validation and cleaning
- Technical Analysis
  - Fast Fourier Transform (FFT) analysis
  - Cycle detection and strength measurement
  - Trading signal generation
  - Price forecasting
- Market Sentiment Analysis
  - News sentiment processing
  - Bullish/Bearish indicator calculation
  - Real-time sentiment updates
- Portfolio Management
  - Multiple stock analysis
  - Portfolio performance tracking
  - Risk assessment
- Reporting and Export
  - PDF report generation
  - CSV data export
  - Interactive visualization

Non-Functional Features

- Performance
  - Fast data processing (< 5 seconds for standard analysis)
  - Efficient database operations
  - Optimized chart rendering
- Security
  - Secure file upload handling
  - Data validation and sanitization
  - Protected API endpoints
- Reliability
  - Error handling and recovery
  - Data backup and persistence
  - Stable API connections
- Usability
  - Intuitive user interface
  - Responsive design
  - Clear documentation
- Scalability
  - Efficient resource utilization

- Modular architecture
- Extensible design

ard">

Data Input Methods

- CSV File Upload: Historical price data analysis
- Stock Ticker Symbol: Real-time market data integration
- Portfolio Import: Multi-stock analysis capabilities

Analysis Capabilities

- Cycle Detection using FFT
- Market Sentiment Assessment
- Portfolio Optimization
- Trading Signal Generation

Visualization Tools

- Interactive Time Series Charts
- Frequency Domain Plots
- Sentiment Gauge Charts
- Portfolio Performance Visualization

Testing and Validation

Testing Methodology

- Unit Tests: Core algorithm validation
- Integration Tests: API and database functionality
- User Interface Testing: Responsive design validation
- Performance Testing: Load and stress tests

System Weaknesses and Limitations

Technical Limitations

- FFT analysis assumes cyclic patterns remain consistent over time
- System may not respond well to sudden market regime changes
- Processing speed limitations for very large datasets
- Memory constraints when analyzing multiple stocks simultaneously

Market-Related Challenges

- Cannot predict unexpected market events or news impacts
- Effectiveness varies in highly volatile market conditions
- Limited ability to detect non-cyclic market patterns
- Sentiment analysis may miss context or nuanced market news

Data Dependencies

- Relies on quality and consistency of input data
- Historical data may not reflect current market dynamics
- Real-time data delays can affect timing of signals
- API rate limits affect data refresh frequency

User Experience Constraints

- Learning curve for understanding FFT analysis concepts
- Complex configuration options may overwhelm new users
- Mobile interface limitations for detailed analysis
- Report generation time increases with data volume

Recommendations and Future Work

- Implementation of machine learning for improved cycle detection
- Enhanced portfolio management capabilities
- Advanced sentiment analysis using NLP
- Real-time data streaming integration
- Mobile application development

References

Mathematical Foundations and Fourier Analysis

1. Fourier, J. (1822). "The Analytical Theory of Heat" - Original foundational work on Fourier series
2. Körner, T.W. (1988). "Fourier Analysis" - Comprehensive treatment of Fourier series and transforms
3. Stein, E.M., Shakarchi, R. (2003). "Fourier Analysis: An Introduction" - Modern perspective on Fourier theory
4. Tolstov, G.P. (1976). "Fourier Series" - Classical text on Fourier series applications
5. Lanczos, C. (1988). "Discourse on Fourier Series" - Intuitive approach to Fourier analysis

Technical References

1. Smith, Steven W. (1997). "The Scientist and Engineer's Guide to Digital Signal Processing"
2. Murphy, John J. (1999). "Technical Analysis of the Financial Markets"
3. Edwards, Robert D., Magee, John (2007). "Technical Analysis of Stock Trends"
4. Oppenheim, Alan V., Schafer, Ronald W. (2009). "Discrete-Time Signal Processing"

Research Papers

1. Johnson, N. F., et al. (2003). "Financial market complexity"
2. Lo, Andrew W. (2004). "The Adaptive Markets Hypothesis"
3. Ehlers, John F. (2004). "Cybernetic Analysis for Stocks and Futures"

Software Libraries and Tools

1. NumPy Documentation - Mathematical computing tools
2. SciPy Documentation - Scientific computing library
3. Flask Documentation - Web framework
4. Plotly Documentation - Interactive visualization library
5. SQLAlchemy Documentation - Database toolkit

Course Instructor

Project Supervisor and Lecturer

Buri Gershom (bgershom@must.ac.ug)

- Project supervision and guidance
- Technical oversight and mentorship
- Academic evaluation and assessment

Authors and Contributors

Lead Developer and Project Manager

Amon Muhwezi (2024BCS109@std.must.ac.ug)

- System architecture and implementation
- Full-stack development
- Project deployment and management

Core Development Team

- **Amanya Peter (2024bcs033@std.must.ac.ug)**
  Group Leader and Project Manager
- **Doreen Miracle (2024bcs130@std.must.ac.ug)**
  Database design and management
- **Twinomugisha Nickson (2024bcs168@std.must.ac.ug)**
  API integration and testing
- **Asiimwe Shabellah (2024BCS049@std.must.ac.ug)**
  Data processing algorithms
- **Kangwagye Jonas (2024BCS194@std.must.ac.ug)**
  System testing and quality assurance

- **Maro Edly (2024BCS097@std.must.ac.ug)**
  Market sentiment analysis implementation
- **Nanjuki Daphine (2024BCS129@std.must.ac.ug)**
  Documentation and user guide development
- **Rwendeire Joshua Truth (2024BCS148@std.must.ac.ug)**
  Portfolio management features
- **Katusiime Moreen (2024bcs084@std.must.ac.ug)**
  Data visualization and reporting

System Features

Functional Features

- Data Import from CSV and API sources
- Fast Fourier Transform (FFT) Analysis
- Market Cycle Detection
- Trading Recommendations
- Interactive Data Visualization
- Portfolio Management
- Market Sentiment Analysis
- PDF Report Generation

Non-Functional Features

- Responsive Web Design
- Real-time Data Processing
- Secure Data Storage
- High Performance Analysis
- Cross-browser Compatibility
- Data Export Capabilities
- Error Handling and Logging
- User-friendly Interface