

1 Uvod

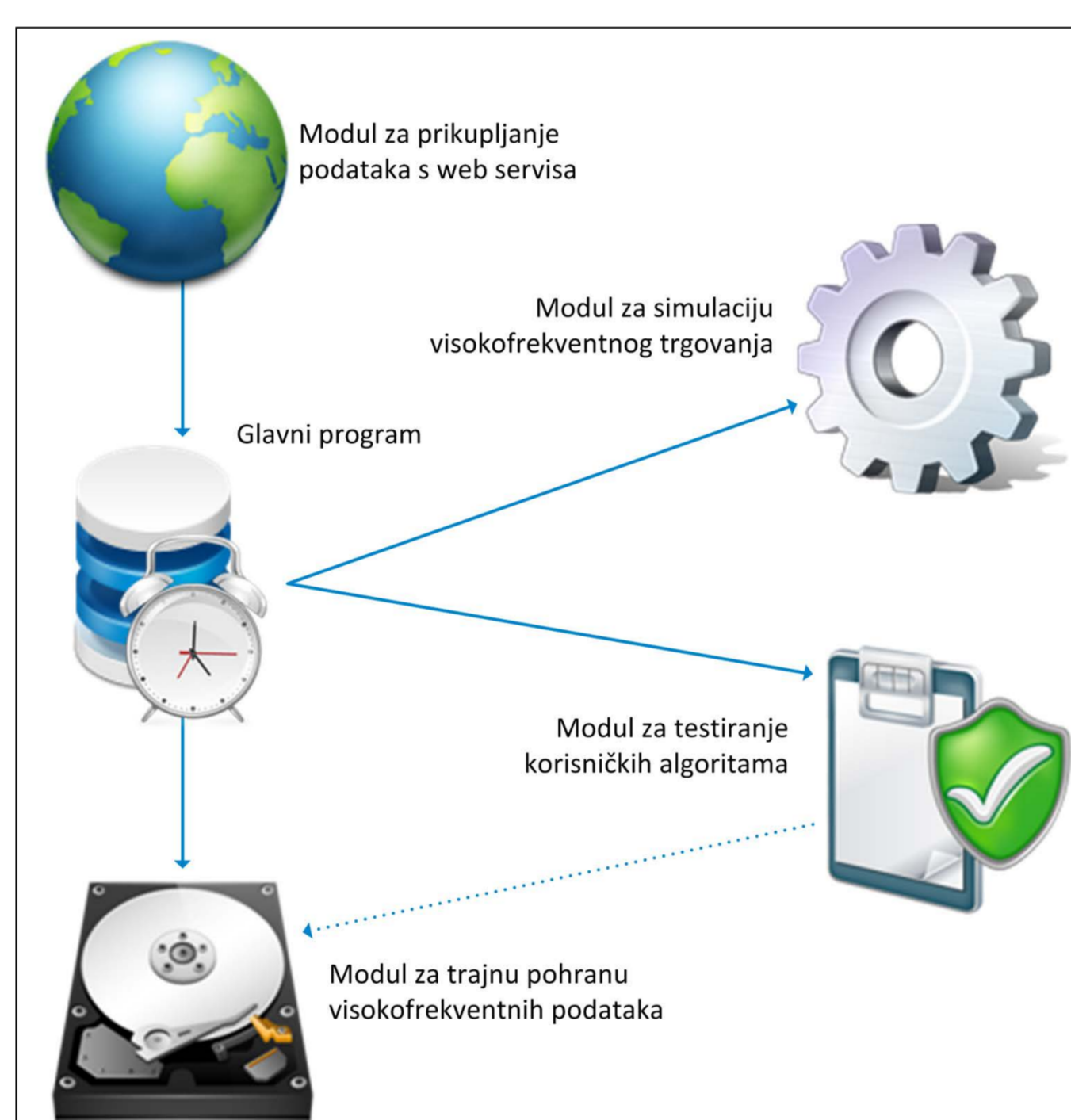
Tehnološkim napretkom i prelaskom burza na elektronički način trgovanja pojavilo se visokofrekventno trgovanje (engl. *HFT – High Frequency Trading*). Ovim načinom trgovanja se ostvarilo preko 60% volumena trgovanja u 2009. godini. HFT se temelji na velikom broju ostvarenih transakcija s relativno malim povratima, pri čemu se kupljene dionice zadržavaju svega od nekoliko minuta do najviše nekoliko dana. HFT algoritmi konstantno prate tržište kako bi uočili nepravilnosti u cijenama i ostvarili neposrednu dobit. Nažalost, visokofrekventno trgovanje je najčešće dostupno samo velikim fondovima i investicijskim bankama jer oni imaju potrebnu informatičku infrastrukturu i pristup podacima u stvarnom vremenu.

2 Opis problema

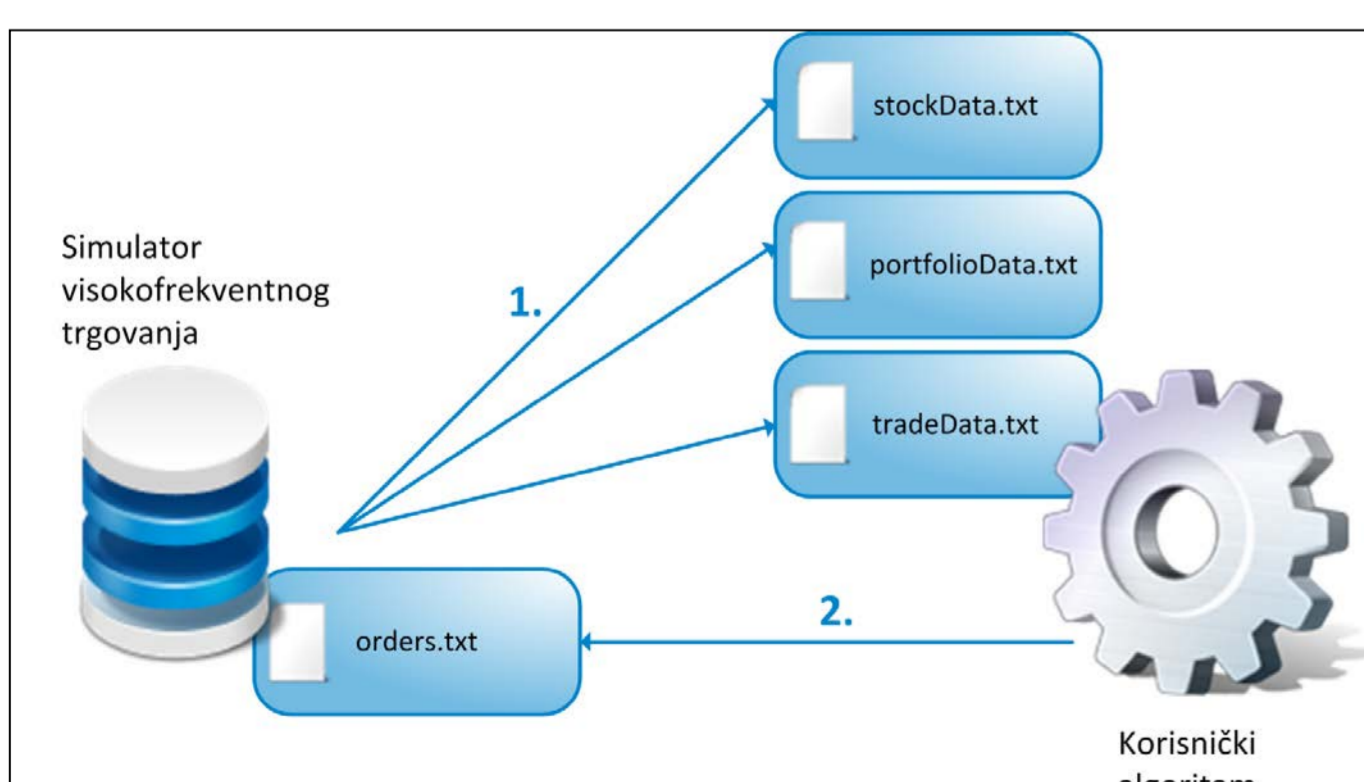
Cilj ovog diplomskog rada je ostvariti sustav za simuliranje visokofrekventnog trgovanja dionicama na Zagrebačkoj burzi koji bi bio namijenjen istraživačima, malim investitorima i entuzijastima. Dodatno, sustav bi trebao omogućiti testiranje korisničkih strategija na povijesnim podacima.

3 Arhitektura sustava

Sustav smo podijelili na glavni program i četiri pomoćna modula koji se oslanjaju na programski model domene. Modul za prikupljanje podataka u stvarnom vremenu koristi JSON web servise te osigurava da svaki modul sustava uvijek ima najnovije podatke o stanju na tržištu. Modul za trajnu pohranu podataka sprema podatke o tržištu.



Modul za simuliranje strategija visokofrekventnog trgovanja se brine o organizaciji i uređenju korisničkih algoritama. Također ovaj modul automatski pokreće simulacije i predstavlja njihove rezultate. Prije nego se ostvareni algoritmi aktiviraju, moguće ih je testirati, za što je odgovoran poseban modul koji provodi testiranje na povijesnim podacima koje je program dotada prikupio.

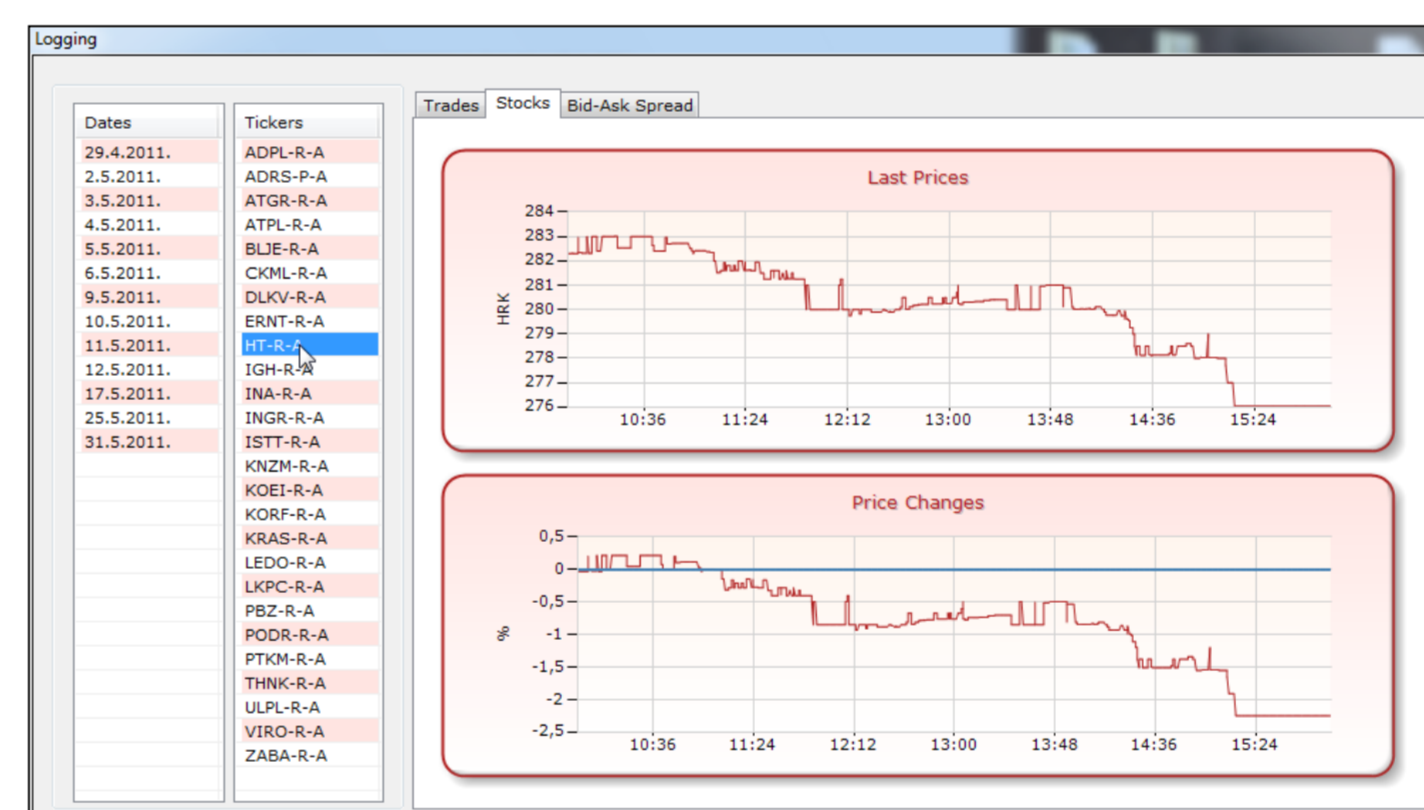
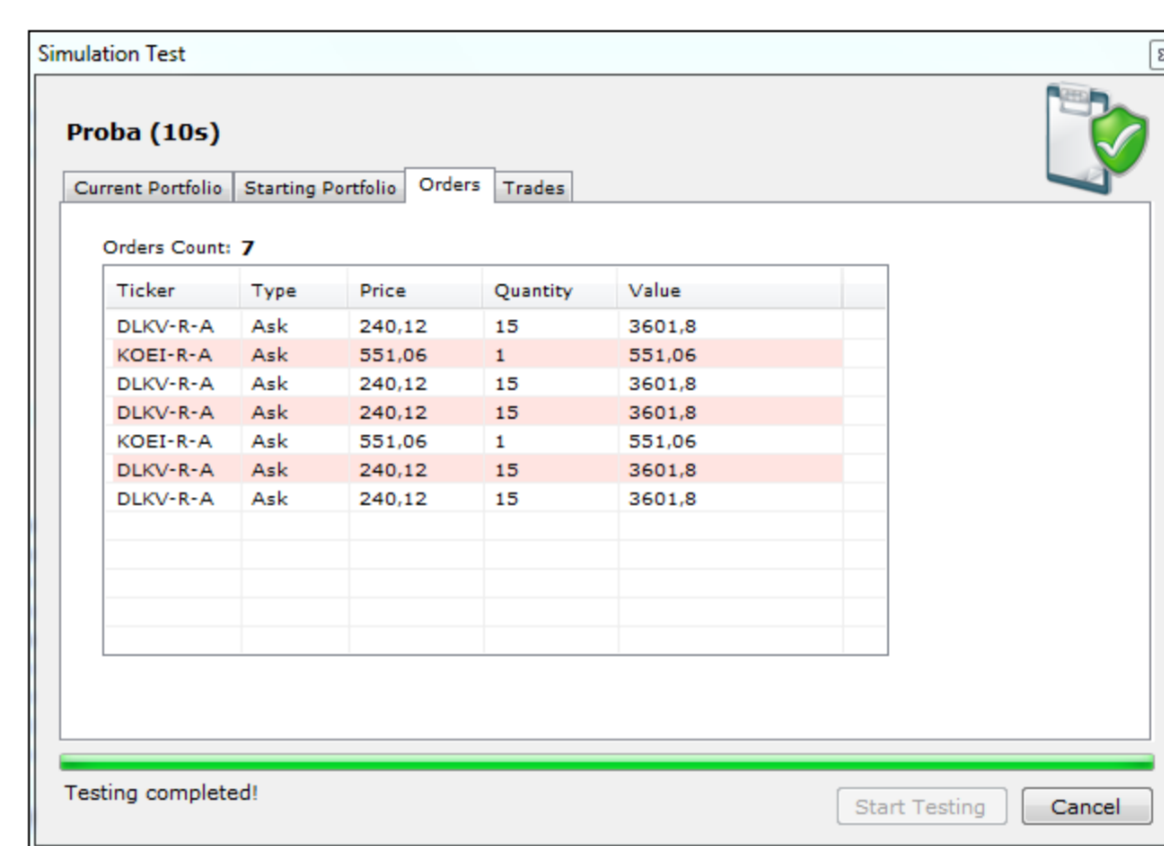
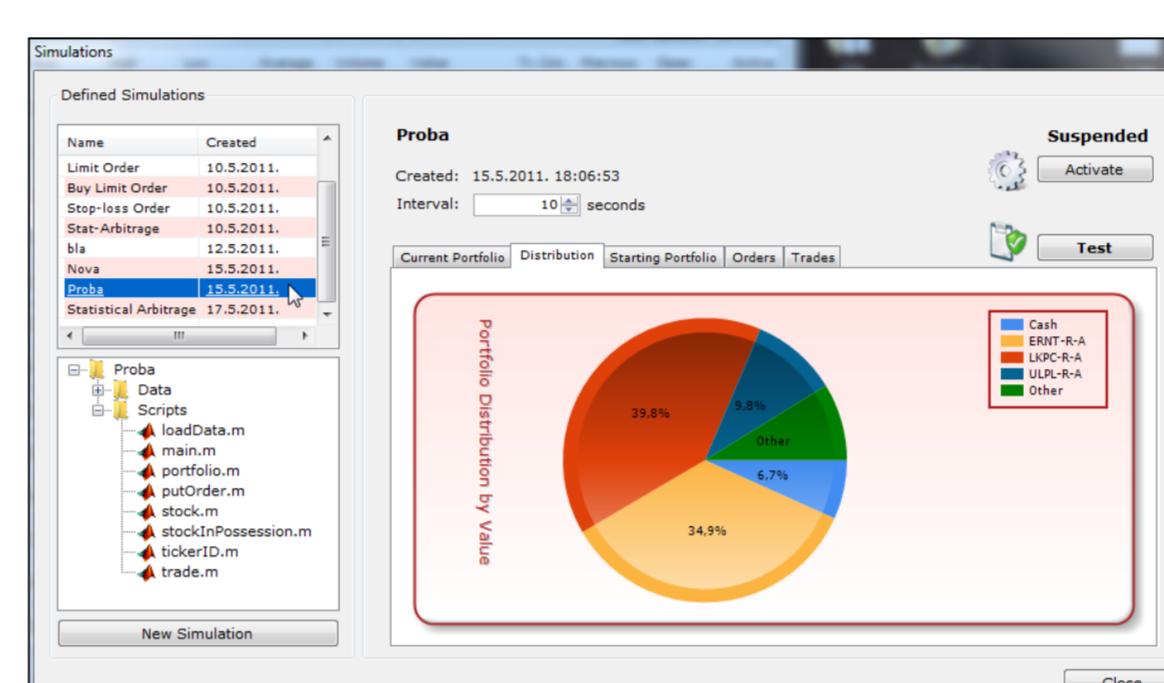
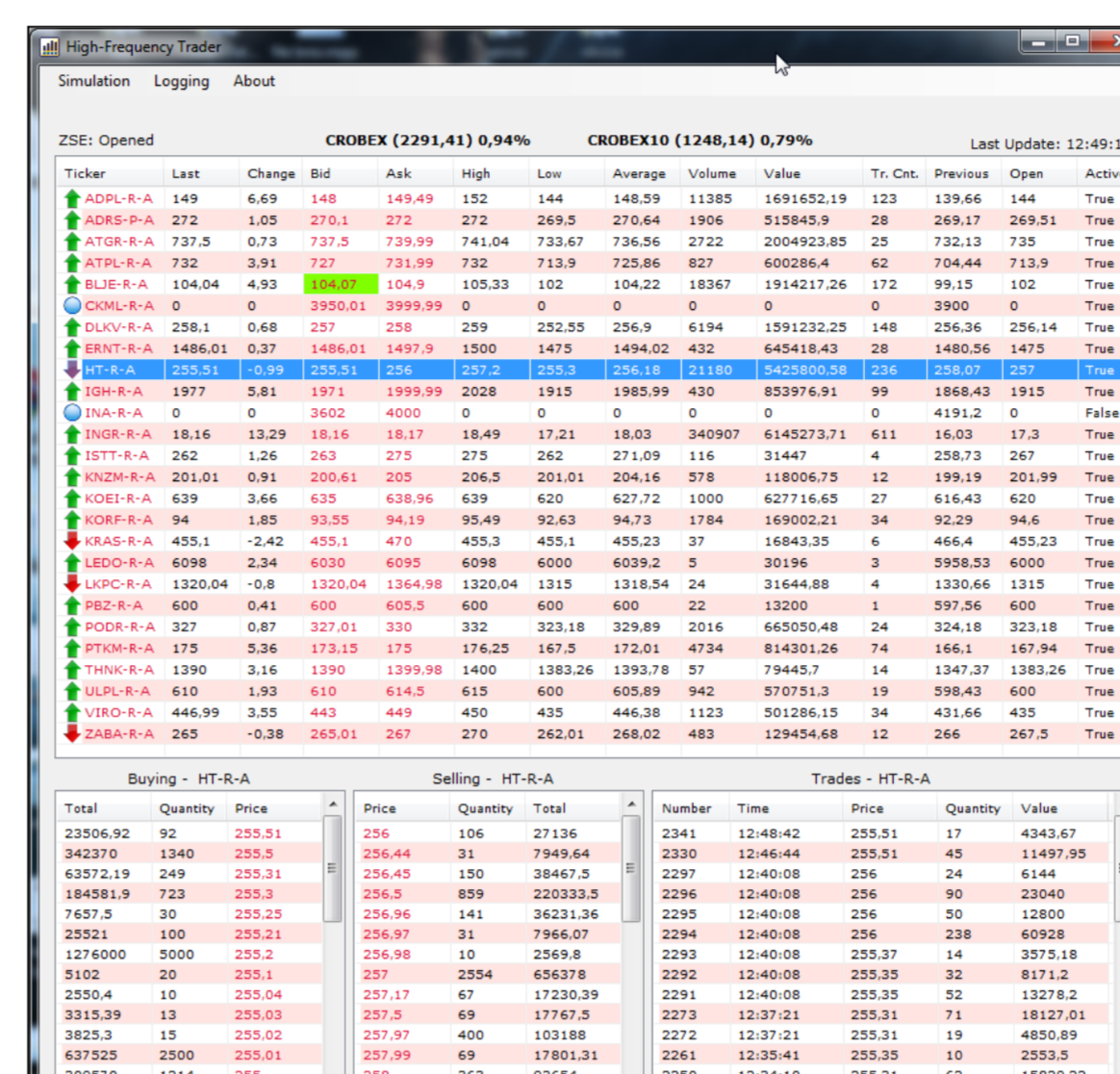


Korisnički algoritmi se implementiraju u programskom jeziku MATLAB. Razmjena podataka između simulatora i korisničkog algoritma se obavlja preko tekstualnih datoteka.

Algoritam ima na raspolaganju trenutne podatke o stanju na Burzi, obavljenim trgovanjima od početka radnog dana te o stanju korisničkog portfelja. Kao rezultat izvođenja, algoritam može zadati jedan ili više naloga za kupnju, odnosno prodaju.

Korištene tehnike i tehnologije:

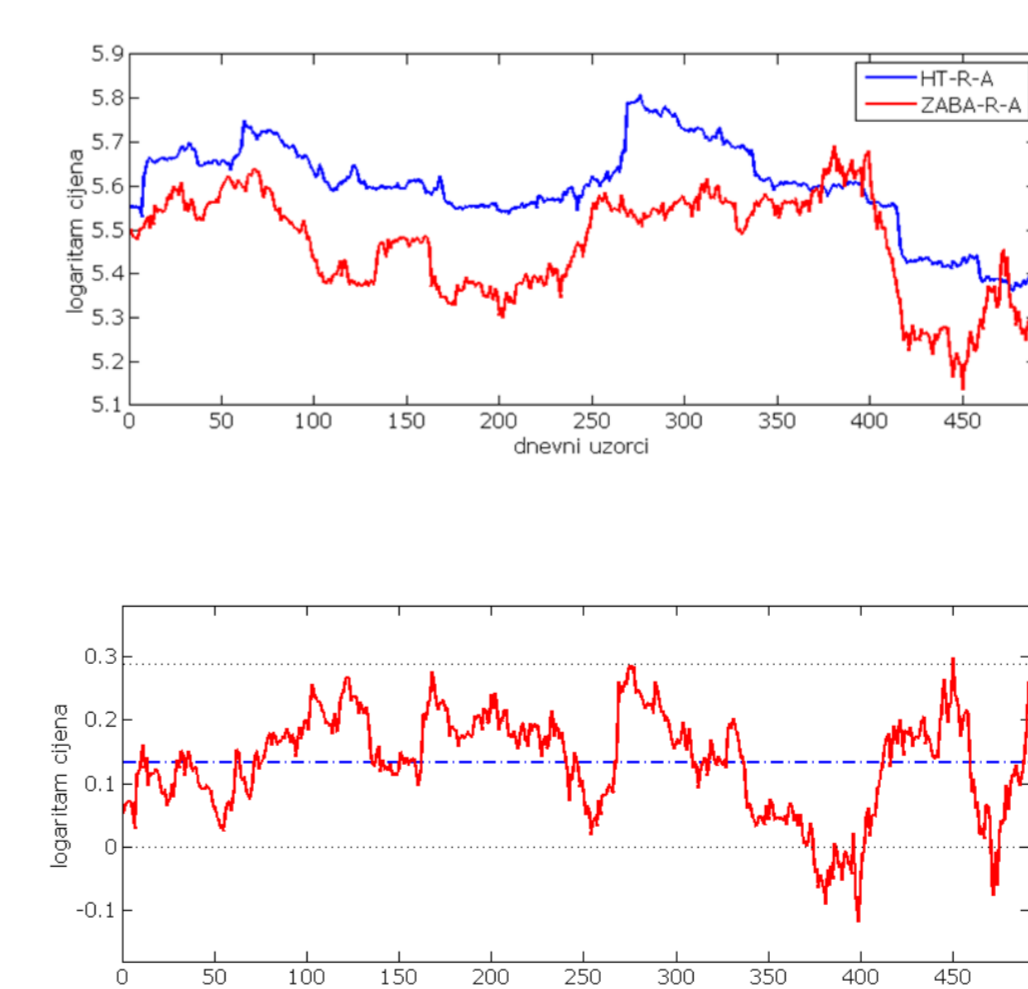
- C# .NET 4.0
- MATLAB
- JSON web servisi
- OO model domene
- Oblikovni obrasci
- Model-View-Controller



- Pregled stanja na Burzi
 - Cijene dionica
 - Aktivni nalozi za kupnju i prodaju
 - Izvršena trgovanja
- Stvaranje nove simulacije
 - Stvaranje početnog portfelja
 - Odabir frekvencije pokretanja
 - Predefinirane MATLAB skripte olakšavaju pisanje algoritama
 - Praćenje izvršavanja u stvarnom vremenu (nalozi, trgovanja, stanje portfelja, vizualizacija)

- Trajna pohrana visokofrekventnih serija
 - Mogućnost dodatnog analiziranja izvan sustava
 - Vizualizacija podataka i brzi uvid u stanje na tržištu (ukupni volumen i vrijednost trgovanja, kretanje cijena, bid-ask spread itd.)
- Testiranje korisničkih algoritama
 - Testiranje ispravnosti izvornog koda
 - Testiranje rezultata algoritma

Implementacija algoritma statističke arbitraže prilagođena za rad s ostvarenim simulatorom



```

% Primjer algoritma statističke arbitraže
mean = 0.1322;
std = 0.0782;

first = stocks(tickerID('HT-R-A'), tickers);
second = stocks(tickerID('ZABA-R-A'), tickers);
diff = log(first.LastPrice) - log(second.LastPrice);

% Prodaj prvu, kupi drugu
if (diff > (mean + 2*std))
    myFirst = myPortfolio.FindStock(first.Ticker);
    if (myFirst == 0)
        putOrder('Sell', first.Ticker, first.BestBid, myFirst.Volume);
    end

% Sav novac od prodaje uloji u kupnju
cash = first.BestBid*myFirst.Volume;
putOrder('Buy', second.Ticker, second.BestAsk, floor(cash/second.BestAsk));
end

% Kupi prvu, prodaj drugu ...
  
```

5 Zaključak

Razvijeni sustav je potpuno funkcionalan te se pomoću njega mogu simulirati HFT algoritmi različite složenosti. Osim HFT algoritama, korisnik ima slobodu izraditi proizvoljnu strategiju trgovanja. Primjerice, može ostvariti složenije vrste naloga kao što su *Stop-loss* i *Market-if-touch* nalozi.

Prilikom izrade sustava, od modeliranja arhitekture do konkretne implementacije, vodilo se računa o modularnosti i jednostavnosti uvođenja budućih poboljšanja i nadogradnji. Iako su moguća daljnja poboljšanja sustava, najveći izazovi kriju se u osmišljanju novih strategija trgovanja i stvaranju inovativnih modela kojima možemo „pobjediti tržište“.