

**ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA  
SISTEMAS ELETRÔNICOS**

**SISTEMA DE APOIO A DECISÃO PARA COMPRA E VENDA  
DE ATIVOS EMPREGANDO LÓGICA FUZZY**

Thiago Mizuta

Orientador:  
Flavio A. M. Cipparrone

São Paulo  
2009

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA  
SISTEMAS ELETRÔNICOS

**SISTEMA DE APOIO A DECISÃO PARA COMPRA E VENDA  
DE ATIVOS EMPREGANDO LÓGICA FUZZY**

Monografia Final do Trabalho de Formatura  
da Escola Politécnica da Universidade de  
São Paulo

Thiago Mizuta

Nº USP 5174720

e-mail: [thiago.mizuta@poli.usp.br](mailto:thiago.mizuta@poli.usp.br)



Orientador:

Flavio A. M. Cipparrone

Departamento PSI

e-mail: [flavioamc@gmail.com](mailto:flavioamc@gmail.com)

São Paulo  
2009

OK

## RESUMO

Este trabalho reúne um projeto de implementação de filtros e Lógica Fuzzy. O texto apresenta como os filtros serão implementados, a partir de séries discretas de preços de ativos praticados no mercado, para a utilização no Mercado Financeiro. E também, como a Lógica Fuzzy entrará no processo de modo a ser responsável por gerar um sistema auxiliador, que devolverá sugestões de decisões como compra ou venda do ativo em questão, sempre visando a maximização dos lucros. Uma importante regra a ser pesquisada e inserida na Lógica Fuzzy será relativo a dados históricos dos volumes negociados deste ativo. Para a calibragem dos parâmetros do sistema, um processo iterativo varrerá toda base histórica de preços deste ativo, e retornará os que obtiverem melhor resultado.

# SUMÁRIO

1 INTRODUÇÃO .....	1
2 JUSTIFICATIVA .....	2
3 METODOLOGIA .....	3
3.1 Implementação do Filtro .....	3
3.1.1 Tipos de Filtros .....	3
3.1.2 Equação do Filtro .....	3
3.1.3 Viabilidade .....	4
3.1.4 Implementação.....	4
3.2 Implementação da Lógica Fuzzy .....	13
3.2.1 Introdução .....	13
3.2.2 Conceito Fuzzy .....	14
3.2.3 Sistemas Fuzzy .....	15
3.2.4 Lógica Fuzzy no Sistema .....	18
3.2.5 Viabilidade .....	18
3.3 Preço da Posição dos Grandes Investidores .....	19
3.3.1 Conceito .....	19
3.3.2 Viabilidade .....	20
3.4 Títulos Públicos.....	20
3.4.1 Letras do Tesouro Nacional.....	21
3.4.2 Notas do Tesouro Nacional.....	22
3.5 Implementação.....	24
4 Cronograma .....	33
5 DISCUSSÃO .....	34
6 CONCLUSÃO .....	35
7 BIBLIOGRAFIA .....	36
Anexo 1 .....	37
Anexo 2 .....	41
Anexo 3 .....	43

# 1 INTRODUÇÃO

Durante grande parte do curso de Sistemas Eletrônicos, estudou-se o comportamento de sistemas e seus respectivos sinais. Neste projeto, apresentarei uma de suas diversas possíveis aplicações, tendo como base os conhecimentos adquiridos neste período.

Detectar tendências e padrões no mercado financeiro tem sido alvo de muita pesquisa. Desenvolver uma ferramenta que auxilia na tomada de decisões de investidores, na área de finanças, seria de grande utilidade para a maximização de seus lucros.

A decisão será gerada com base em Lógica Fuzzy, onde as entradas do sistema serão os preços dos ativos, e suas respectivas respostas quando processadas por determinados filtros.

Para otimizar, ainda mais os resultados, dados relativos aos volumes negociados em um determinado período e seus respectivos preços serão utilizados como regras na Lógica Fuzzy.

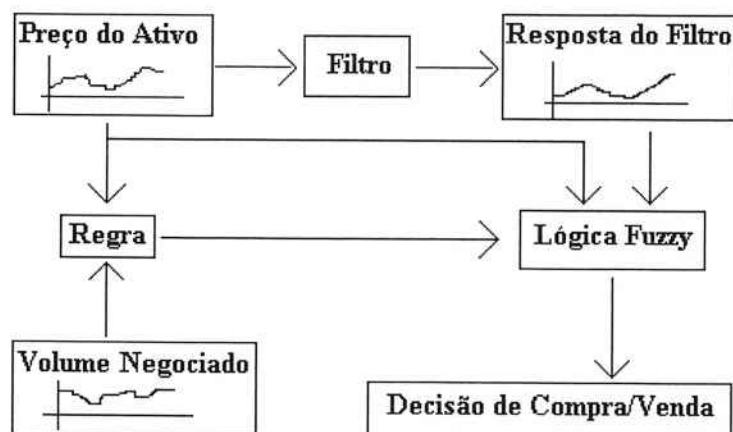


Figura 1: Diagrama referente à idéia geral do projeto.

## **2 JUSTIFICATIVA**

A principal motivação para a escolha deste tema é a área em que estou estagiando, e que pretendo permanecer depois de concluído o curso. Durante este período, foi possível verificar a complexidade do Mercado Financeiro, e observar que o emprego de uma sistemática consistente seria de grande utilidade na rotina dos analistas financeiros.

Devido à própria dinâmica do mercado, um sistema automatizado, com atualizações em tempo real, serviria de apoio às decisões, que devem ser tomadas agilmente. A cada nova amostra disponível na base de preços, o sistema irá re-ajustar os parâmetros do sistema. Isto tornaria a ferramenta consistente e atualizada.

O tratamento dos preços dos ativos como sinais discretos, somando os estudos realizados durante o curso de Engenharia Elétrica na área de Filtros, Sistemas e Sinais, resultarão em um projeto de Engenharia para a área de Mercado Financeiro.

## 3 METODOLOGIA

### 3.1 Implementação do Filtro

Analistas do mundo inteiro pesquisam indicadores de tendência de alta e baixa dos preços do mercado. No indicador ideal, o investidor saberia o momento exato para a compra ou venda do ativo.

O que se buscará nesta metodologia serão exatamente os parâmetros para criar um indicador que se aproxime dos resultados gerados pelo ideal.

Este indicador terá como base a resposta de um filtro que será implementado.

#### 3.1.1 Tipos de Filtros

Por se tratar de uma ferramenta que devolve decisões em tempo real, a resposta do filtro no instante  $n$  só poderá depender de entradas no instante  $n$  e anteriores a ele, uma vez que os instantes posteriores a  $n$  ainda não ocorreram, ou seja, o filtro a ser implementado tem de ser *causal*.

A utilização em filtro passa baixa, que calcula a média de uma janela móvel, tem uso consagrado na área de análise técnica, que pressupõe que os preços futuros dos ativos têm relação com o passado, e será a utilizada no sistema.

#### 3.1.2 Equação do Filtro

O filtro a ser implementado, tem a seguinte equação:

$$y(n) = \frac{\sum_{i=1}^m (p(m-i+1).x(n-i))}{\sum_{i=1}^m p(m-i+1)}$$

Eq. 3.1.2.1



Onde  $y(n)$  é a resposta do filtro,  $x(n-i)$  são as amostras de preços a serem analisadas,  $m$  é a ordem do filtro.

A função  $p$  é o devido peso que se dará a cada amostra, uma vez que possivelmente o preço do ativo no dia  $n$ , tem mais relação com o preço do dia  $n-1$  do que com o do dia  $n-2$  (mas isso será visto mais pra frente, na parte de otimização).

O atraso do filtro é diretamente proporcional à ordem  $m$ , filtros de ordem mais alta têm atrasos maiores. Para os investidores, decisões atrasadas são de total ineficiência. Porém filtros de ordem mais baixa, por conterem menos amostras a serem analisadas, podem capturar tendências erroneamente, tornando o sistema ineficaz.

### **3.1.3 Viabilidade**

O estudo de filtros e sinais já foi efetuado com eficácia no programa MatLab. A implementação e otimização do filtro seguirão os mesmos procedimentos aprendidos durante o curso, que se mostrou de completa viabilidade.

### **3.1.4 Implementação**

A partir de estudos em bases históricas, será possível encontrar os melhores parâmetros que resultarão no melhor investimento ao usuário. Um algoritmo em Matlab será responsável por essa tarefa.

Inúmeras combinações das variáveis serão testadas, e a que resultar em uma predição mais eficiente, será utilizadas para tomar a decisão no momento. Após a disponibilização de um novo preço, um novo dado entrará para a base e novas combinações serão testadas e recalculadas, obtendo assim, sempre os melhores parâmetros para o dado momento.

Basicamente, serão dois os parâmetros a serem analisados: a ordem do filtro, que representa a quantidade de pontos a serem consideradas para a predição, e o vetor de pesos dados às amostras passadas.

Toda a base histórica servirá como treinamento e teste dos parâmetros a serem pesquisados. Seja  $h(n)$  a função do filtro a ser implementado:

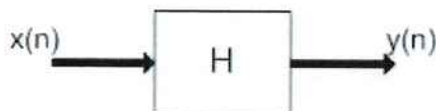


Fig. 3.1.4.1

Sendo  $\beta$  a base histórica dos preços com  $k$  elementos, e o filtro  $h$  de ordem  $m$ , o sistema otimizador varrerá os  $k-m$  elementos  $\alpha$  vezes, onde  $\alpha$  é o número de variações do filtro que serão analisadas. Estas  $\alpha$  variações resultam da combinação de  $n$  alterações na ordem  $m$  do filtro e de  $q$  modificações na função  $p$  de pesos na amostras, logo se chega ao número de  $(k-m)*\alpha$  varreduras sobre a base  $\beta$ . Cada um dos ciclos de  $k-m$  varreduras pode ser visualizado da seguinte forma:

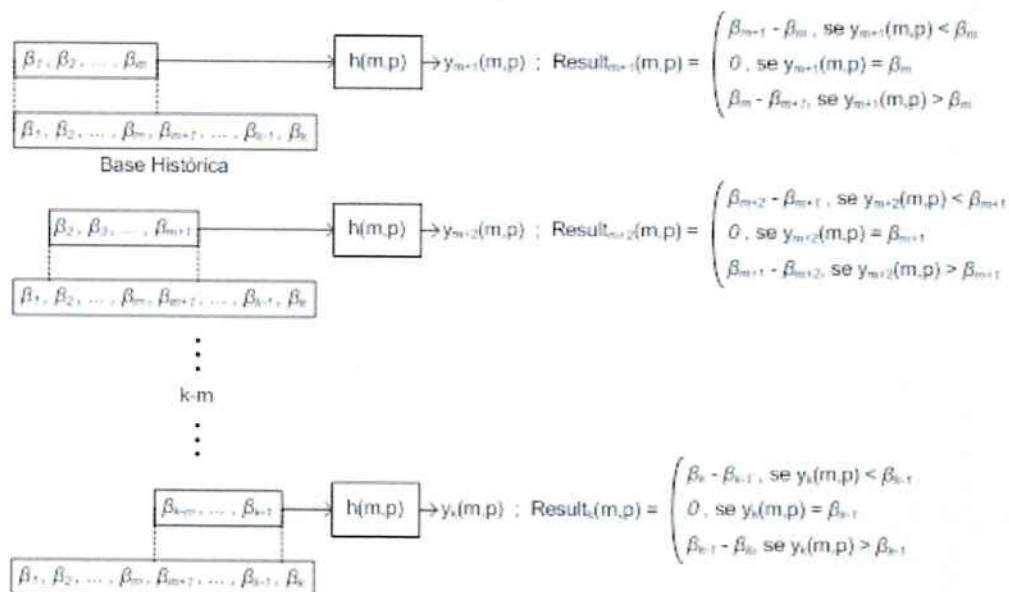


Fig. 3.1.4.2

As condições para determinar o resultado gerado por cada um dos  $k-m$  elementos do vetor de *result* foram tomadas pela seguinte suposição: se a resposta do filtro  $h(m,p)$  for menor que a amostra de menor atraso na entrada, o preço do ativo em questão está em uma tendência de alta e o

sistema sugerirá que o investidor compre o ativo, gerando o resultado líquido de  $\beta_i - \beta_{i-1}$ ; porém se a resposta do filtro  $h(m,p)$  for maior que a amostra de menor atraso na entrada, o preço do ativo em questão está em uma tendência de queda e o sistema sugerirá que o investidor venda o ativo, gerando o resultado líquido de  $\beta_{i-1} - \beta_i$ , caso a resposta do filtro for igual a amostra de menor atraso, o investidor será sugerido a não tomar posição.

$$Total(m, p) = \sum_{i=m+1}^k Result_i(m, p)$$

Eq. 3.1.4.1

Este processo produzirá o valor  $Total(m,p)$ , que é o resultado que o filtro  $h(m,p)$ , com ordem  $m$  e função de peso  $p$ , gerou na base histórica  $\beta$ .

O sistema otimizador calculará os  $\alpha$  diferentes valores de  $Total(m,p)$ , provenientes das  $n$  variações no valor de  $m$  e das  $q$  variações no valor de  $p$ , e utilizará o modelo que resultar no maior valor.

Como a ordem do filtro é uma variável discreta, basta selecionar um valor máximo de amostras passadas que pode ter influencia sobre o preço atual, e colocar o otimizador em funcionamento.

Para a função de peso, será utilizado um vetor linear de pesos, e as  $q$  variações serão provenientes das alterações do coeficiente angular  $\theta$  deste vetor. O vetor base tem  $\theta$  igual a zero, ou seja, todas as amostras têm o mesmo peso. Cada uma das  $q$  modificações neste vetor aumentará em 10% o peso da amostra de menor atraso em relação à de maior atraso.

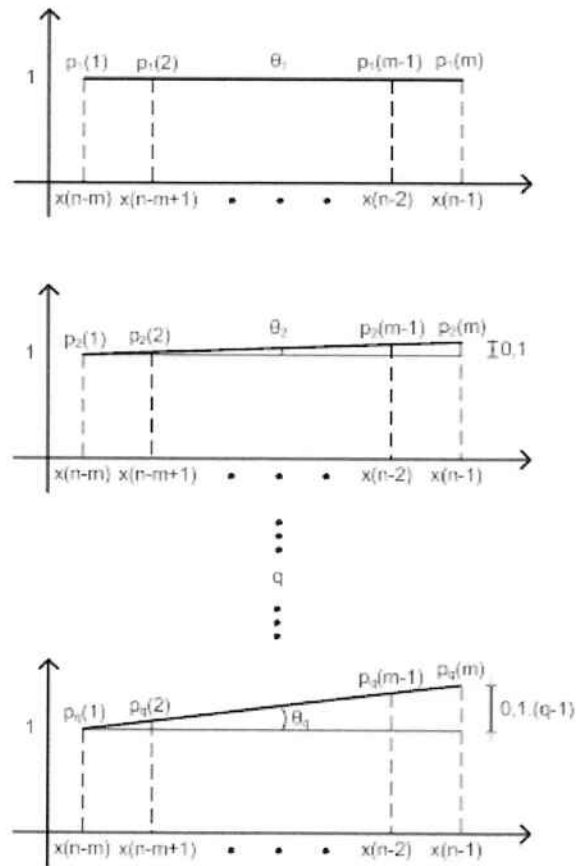


Fig. 3.1.4.3

Para cada uma das  $q$  variações de  $p$ , se chega à seguinte equação da inclinação  $\theta_q$ :

$$\tan(\theta_q) = \frac{0,1 \cdot (q-1)}{m-1}$$

Eq. 3.1.4.2

Fixado esta inclinação, os elementos do vetor  $p$  têm a seguinte equação:

$$p_q(i) = 1 + \tan(\theta_q) \cdot (i-1)$$

Eq. 3.1.4.3

Estabelecido os parâmetros  $m$  e  $p$ , a próxima etapa é calcular a matriz  $Total(m,p)$  variando os valores destes e verificar o seu ponto máximo, que

serão os parâmetros do filtro a serem utilizados para o modelo para comparar com a próxima amostra. Tendo este próximo ponto, esse procedimento será repetido, agora com um ponto a mais na base, e se encontrarão novos parâmetros.

As funções e comandos utilizadas para esta otimização podem ser encontrada ao final do trabalho no Anexo 1. A partir de uma base de dados e os limites dos valores de  $n$  e  $q$ , o Programa Principal consolida as funções descritas na sequência para encontrar os melhores parâmetros do filtro.

A primeira função utilizada é a *funcao\_peso*. Esta é responsável, dado número  $m$  de amostras a serem implementadas no filtro e a inclinação  $q$  de peso, por devolver um vetor linear com  $m$  pontos, sendo que o primeiro ponto vale 1 e o ponto na última posição,  $1 + 0,1.m$ .

A próxima função da sequência é a *filtro*. A partir da base  $\beta$ , da ordem  $m$  e o do vetor de pesos encontrado no passo anterior, esta função devolve o vetor de saída do filtro  $h(m,p)$ , nomeado como  $y$  na figura 3.1.4.2.

Tendo a resposta do filtro e a base histórica, a próxima etapa é gerar um dos  $n \times q$  pontos da matriz *Total*. A função *calcula* verifica as condições citadas na figura 3.1.4.2 e calcula o resultado que o modelo gera no próximo ponto da base.

Para exemplificar esta matriz de uma implementação, foram escolhidos  $n$  e  $q$  iguais a 100 e a taxa de um título público como base histórica. A seguir, o gráfico da base a ser utilizada neste exemplo:

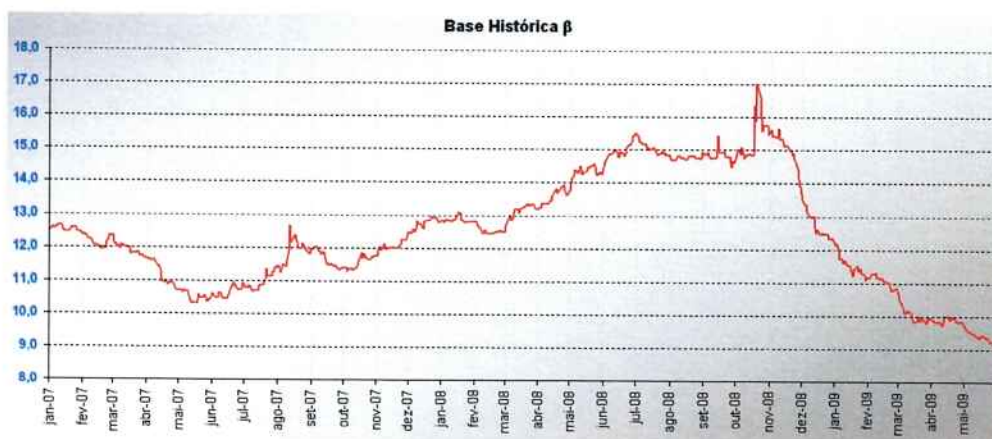


Fig. 3.1.4.4



Tendo a base e os limites dos parâmetros estabelecidos (ambos iguais 100), já se pode começar o processo de otimização seguindo o diagrama em 3.1.4.2. Lembrando que este ciclo servirá para determinar apenas um ponto da matriz, no processo inteiro, o programa executará, neste exemplo, 10.000 vezes este fluxograma, ou seja, serão testados dez mil filtros diferentes. Portanto, a seleção dos máximos valores dos parâmetros não deve ser aleatória.

A etapa final do processo de otimização é a leitura do máximo ponto da matriz gerada e os parâmetros que resultaram neste ponto. A seguir, a superfície visualizada a partir desta matriz. Para sua visualização, utilizaram-se os comandos no Anexo1 intitulados Comandos Geradores de Gráfico.

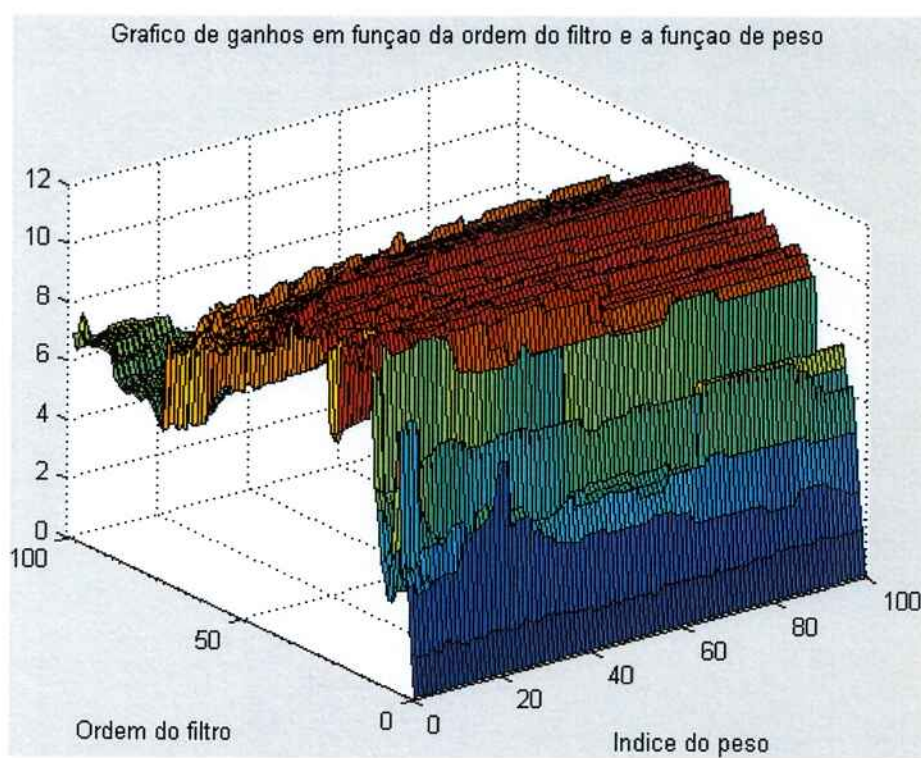


Fig. 3.1.4.5

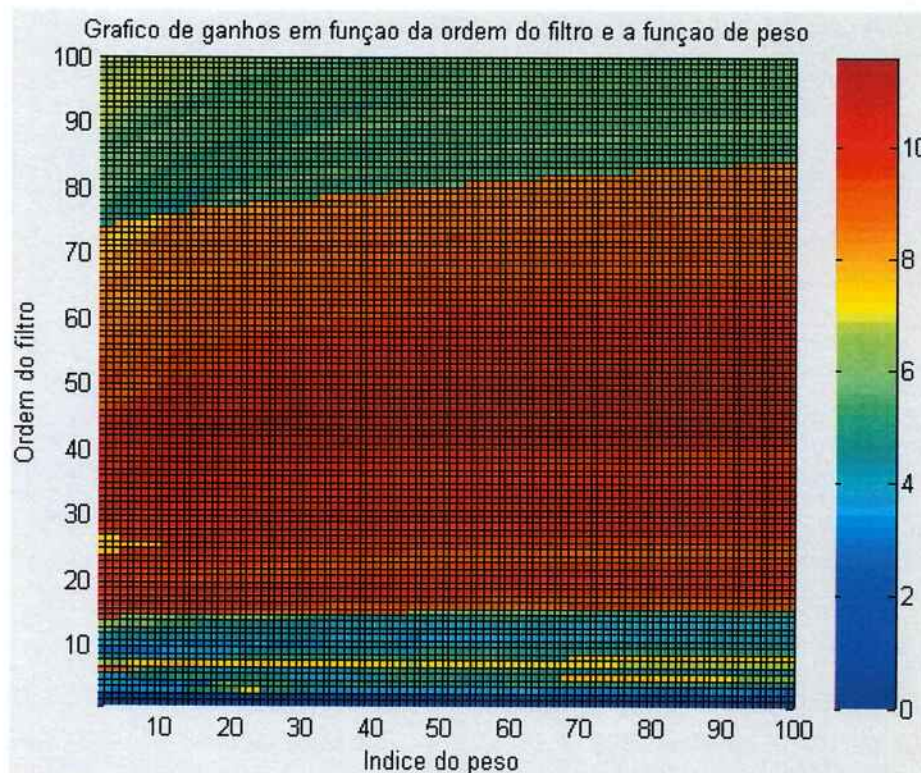


Fig. 3.1.4.6

Neste exemplo, o ponto de maior ganho será de 11,58, onde a ordem do filtro é de 41 e índice de peso, de 35. Os códigos e funções escritos em MatLab estão todos listados no anexo 1.

Com estes parâmetros, é possível construir a equação do filtro que será utilizada para a próxima amostra substituindo o valor de 41 na variável  $m$  da equação 3.1.2.1.:

$$y(n) = \frac{\sum_{i=1}^{41} (p(41-i+1).x(n-i))}{\sum_{i=1}^{41} p(41-i+1)}$$

Eq. 3.1.4.4

Para a função de pesos, substitui-se o valor de 35 na variável  $q$  da equação 3.1.4.2. e chega-se à seguinte equação para determinar a inclinação de sua reta:



$$\tan(\theta_q) = \frac{0,1.(35-1)}{41-1} = \frac{3,4}{40} = 0,085$$

Eq. 3.1.4.5

Logo a função de pesos estabelecida na equação 3.1.4.3, substituindo as variáveis encontradas, é:

$$p_q(i) = 1 + 0,085.(i - 1)$$

Eq. 3.1.4.6

Encontrada a equação do filtro otimizado, é possível visualizar um gráfico da saída gerada por este filtro:



Fig. 3.1.4.7

Com estas duas curvas e a lógica da figura 3.1.4.2, traça-se a curva de resultado:



Fig. 3.1.4.8

Uma importante observação é que só é possível encontrar este filtro, e conseqüentemente, gerar este resultado, tendo a base inteira de preços.



Para simular uma atuação real deste modelo usar-se-á a seguinte metodologia: para cada ponto da amostra, o processo de otimização será utilizado e o resultado que se acumulará será o que este modelo produzirá na amostra seguinte.

Como o processo de otimização demorou cerca de 1 minuto para gerar um modelo, o método para encontrar a real atuação do modelo irá demorar em torno de 600 minutos, já que a base tem mais de 600 pontos. Então para esta simulação, será feita uma simplificação e a resolução de funções de pesos irá diminuir: ao invés de 100 funções diferentes de peso, usar-se-á apenas 10 com o mesmo valor máximo de inclinação, ou seja, a cada iteração, a função de pesos andar-á a passos maiores.

Os novos comandos a serem utilizados podem ser encontrados no Anexo 2. Estas linhas de comandos nada mais são do que uma chamada do programa principal no Anexo 1 a cada um dos pontos da base histórica. Lembrando que este processo será realizado com muita pouca frequência, uma vez que servirá apenas para medir a performance do modelo na base histórica.

Uma outra informação derivada deste processo é a variação dos parâmetros com o crescimento da base histórica. Tanto o vetor de ordem, quanto o de peso do filtro otimizado ficarão guardados no vetor *vetor\_imax* e no vetor *vetor\_jmax*, respectivamente.

Após cerca de 25 minutos de simulação, chegou-se ao seguinte resultado:

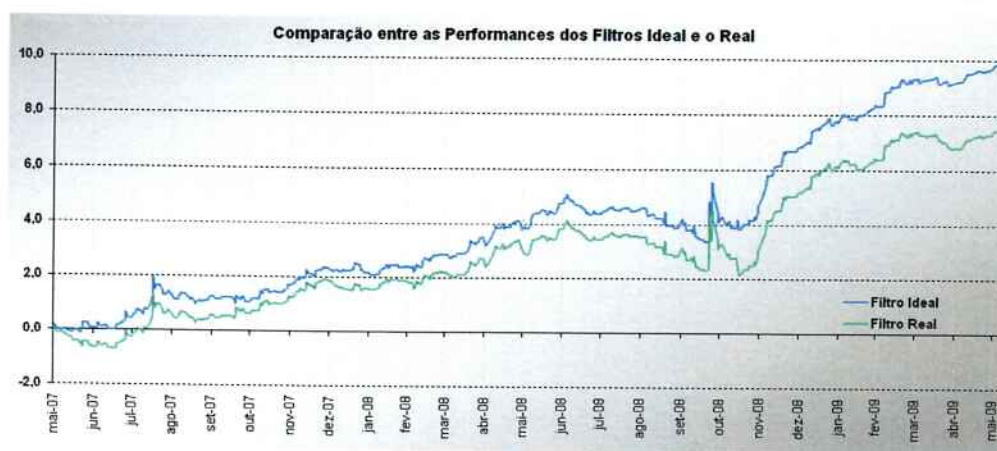


Fig. 3.1.4.9

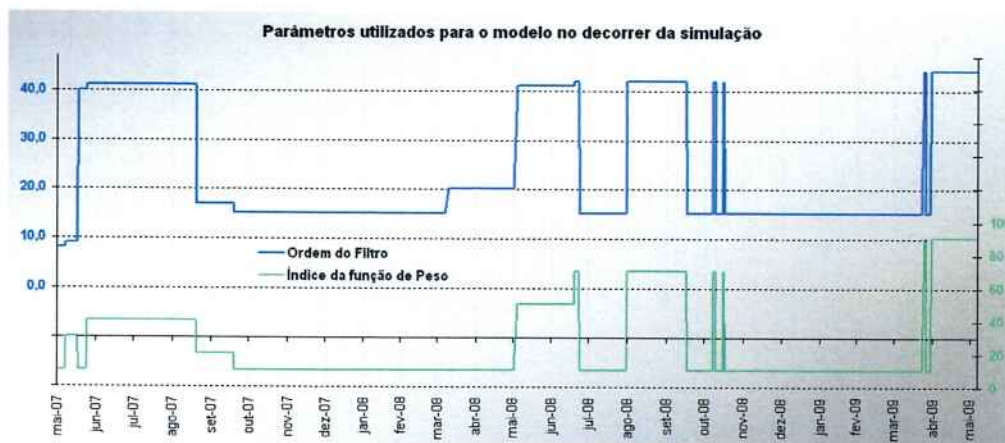


Fig. 3.1.4.10

Observa-se uma oscilação entre praticamente dois pontos, tanto na ordem do filtro quanto no índice de peso. Isso mostra que o modelo transita entre duas regiões da matriz de resultado. Na figura 3.1.4.6, verifica-se estas duas regiões com uma tonalidade mais escura.

O resultado obtido na simulação real da implementação do modelo foi bem próximo ao do filtro ideal, encontrado apenas ao final da base. Nas etapas seguintes, com a utilização da ferramenta Fuzzy, se tentará a melhora destes resultados.

## 3.2 Implementação da Lógica Fuzzy

### 3.2.1 Introdução

Estas últimas décadas foram testemunhas de um rápido crescimento da quantidade e variedade das aplicações em lógica fuzzy. Estas aplicações vão desde utilitários domésticos, como câmeras fotográficas, máquinas de lavar pratos e forno micro-ondas, até as aplicações de processos industriais, instrumentos medicinais e sistemas de auxílio a decisões mais variadas.

A lógica fuzzy pode ser entendida como dois tipos básicos de ferramenta. A primeira, um sistema lógico que trabalha com múltiplas entradas e saídas. A segunda, mais genérica, uma teoria que atribui classes, e seus respectivos graus de pertinência, aos objetos.

O que torna a lógica fuzzy tão interessante, é o fato de que ele se comporta como o raciocínio humano, então seu entendimento fica claro e lógico. As funções relacionadas à condição são como utilizadas comumente entre as pessoas.

### 3.2.2 Conceito Fuzzy

A teoria de conjuntos Fuzzy foi originalmente apresentada por Zadeh em 1965. Uma de suas principais características é a adoção do chamado “índice de pertinência”.

Seja o conjunto  $U$  formado por  $a_1, a_2, a_3, \dots, a_n$  e os subconjuntos  $S_1, S_2, S_3, \dots, S_m$  de  $U$  disjuntos, ou seja, a intersecção entre si é vazia. Considere ainda  $0 \leq \mu_{S_i(a_j)} \leq 1$ ,  $i = 1, 2, \dots, m$  e  $j = 1, 2, \dots, n$ , tais que, para todo  $j = 1, 2, \dots, n$ , tem-se:

$$\sum_{i=1}^m \mu_{S_i(a_j)} = 1$$

Eq. 3.2.2.1

Deste modo,  $\mu_i(a_j)$  é dito como índice de pertinência do elemento  $a_j$  no subconjunto  $S_i$ , onde  $\mu$  é a regra estabelecida.

Como exemplo, seja o conjunto  $U$  formado pelos valores no intervalo  $-1$  e  $1$ , inclusive, e os subconjuntos  $P$ , dado pelos números maiores do que zero, e  $N$ , dado pelos números menores do que zero.

Na teoria clássica de conjuntos,  $1$  e  $0,5$  pertenceriam ao subconjunto  $P$ . E graficamente:

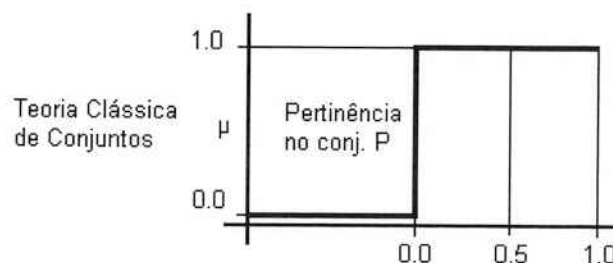


Fig. 3.2.2.1

Porém, na teoria de conjuntos Fuzzy, a associação do índice de pertinência aos elementos de  $U$ , introduz idéias como 1 pertence mais ao conjunto  $P$  do que 0,5, ou seja,  $\mu_{P(1)} > \mu_{P(0,5)}$ , onde  $\mu$ , por exemplo, é a regra dada pela distância do elemento ao zero. Se  $\mu_{P(1)} = 1$ , sabendo que  $\mu_{P(1)} + \mu_{N(1)} = 1$ , temos que  $\mu_{N(1)} = 0$ . Neste caso,  $\mu_{P(0,5)} = 0,75$  e  $\mu_{N(0,5)} = 0,25$ , respeitando a condição imposta de  $\mu_{P(0,5)} + \mu_{N(0,5)} = 1$ . E graficamente:

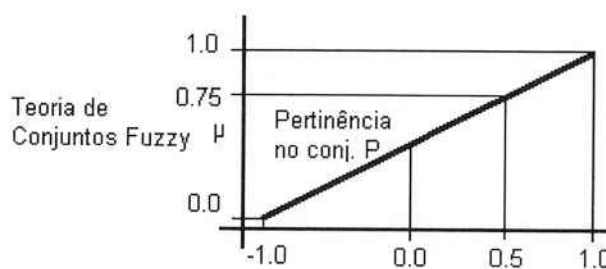


Fig. 3.2.2.2

### 3.2.3 Sistemas Fuzzy

Um Sistema Fuzzy consiste em um sistema de entrada e saída, onde o tratamento destes segue o conceito de conjuntos Fuzzy citado anteriormente. Visualizando o diagrama de blocos, tem-se:



Fig. 3.2.3.1

Onde Fuzificar é devolver o índice de pertinência em cada subconjuntos de entrada existentes do valor capturado. E Defuzificar significa, através dos índices de pertinência em cada classe de saída que o sistema Fuzzy devolver, traduzi-lo para um valor de saída.

O caixa de ferramentas do Fuzzy apresenta alguns tipos pré-determinados de funções de pertinência, das quais podem-se mudar seus parâmetros, e/ou combiná-los, e conseguir a função adequada para sua utilização.

Para combinar as funções, os operadores lógicos serão utilizados, porém como as variáveis não são binárias, operações como “e” e “ou” serão traduzidas para mínimo e máximo, respectivamente.

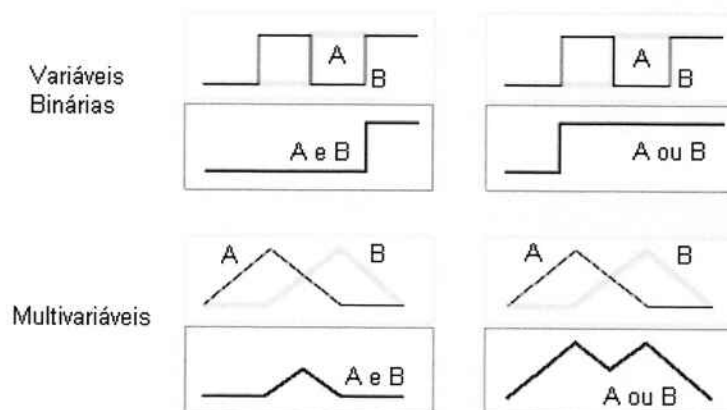


Fig.3.2.3.2

Estas operações serão fundamentais para adquirir a função de pertinência desejada, caso se queira combinar as que estão presentes na caixa de ferramentas do Fuzzy.

As regras no Sistema Fuzzy são escritas na forma condicional e descreve a dependência de um valor de saída em relação ao de entrada. O operador usado nestas regras é o “se... então...”. Na prática, supondo a implicação *se A então B*, o resultado seria o mínimo entre A e B.

Para visualizar estas operações, suponha a regra: Se  $x_1$  está  $A_1$  e  $x_2$  está em  $A_2$  então y está em B. Dadas a funções de pertinência  $A_1$ ,  $A_2$  e B, chega-se a:

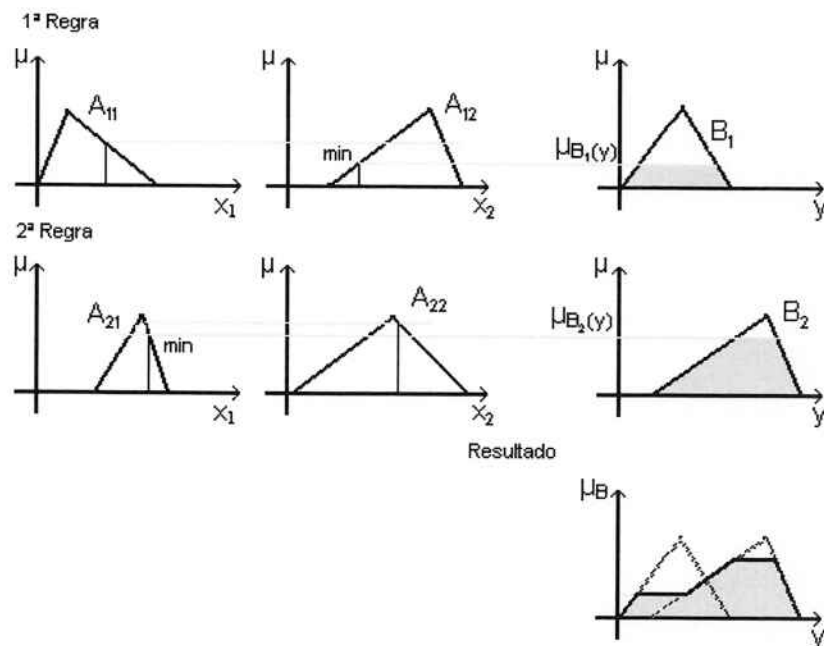


Fig. 3.2.3.3

A Defuzificação ocorre no momento em que se obtém os índices de pertinência resultantes das regras implementadas. Nesta etapa, os dados tipo Fuzzy são traduzidos para os dados literais que são descritos na saída do sistema.

Uma das maneiras de Defuzificar é pelo método do Centróide dado por:

$$y_{saída} = \frac{\int \mu_B(y) dy}{\int dy}$$

Eq. 3.2.3.1

Para melhor detalhar o sistema Fuzzy, a figura 3.2.3.1 pode ser expandida para o seguinte diagrama:



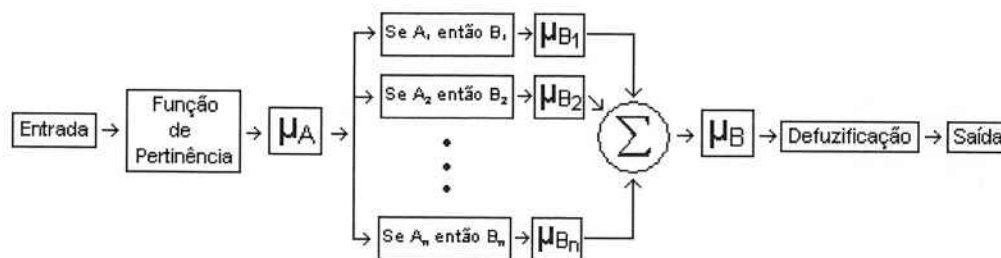


Fig. 3.2.3.4

### 3.2.4 Lógica Fuzzy no Sistema

O sistema Fuzzy captará os sinais de entrada e os classificará nos dois subconjuntos de saída, o de compra e o de venda do ativo, devolvendo o índice de pertinência do estado atual em cada um dos subconjuntos.

Para a classificação, são necessárias regras que atribuirão o devido subconjunto ao cenário em questão. No trabalho de formatura I, foram apresentadas regras iniciais e neste trabalho, serão otimizadas.

Em conjunto com os profissionais da área de Mercado Financeiro, pesquisaram-se as regras que comumente são seguidas na prática.

Como regra adicional, será implementado um conceito pouco pesquisado na área, e se medirá o impacto dela na eficiência do sistema.

### 3.2.5 Viabilidade

Assim como na implementação de filtros, o estudo, análise e inserção da Lógica Fuzzy no sistema ocorrerá por meios do programa MatLab. Este aplicativo possui uma caixa de ferramenta específica para o Fuzzy. Isto o torna, o estudo, análise e implementação, viável e aplicável no sistema a ser projetado.

A Lógica Fuzzy também pode ser implementada através de linhas de comando. Isto tornará a parte de otimização viável, das quais serão necessárias as elaborações de algoritmos, que também serão escritas em linhas de comando, possibilitando a sua comunicação.

### 3.3 Preço da Posição dos Grandes Investidores

#### 3.3.1 Conceito

Todo e qualquer preço de ativo negociado no mercado, depende exclusivamente de oferta e demanda. Quando a oferta é maior que a demanda, o preço cai. E na ponta oposta, quando a demanda é maior que a oferta, o preço sobe.

Deste modo, todo investidor consegue influenciar, no presente e no futuro, o preço do mercado, uns mais, e outros menos. O seu poder de influência é diretamente ligado ao tamanho do capital investido neste ativo. Quanto maior o capital, maior a influência.

A atuação dos grandes investidores é consequência de um alto nível de conscientização. Na tentativa de capturar os grandes investidores, será utilizado o volume negociado em cada dia.

Porém, o dado mais importante a ser analisado será o preço médio que este investidor negociou para adquirir esta posição. Assim, podem-se detectar momentos em que o investidor defende a sua posição, ou se desfaz dela, trazendo a predição requerida pelo sistema.

Para obter este preço médio, será feita uma ponderação do seu preço e seu volume negociado no dia sobre o volume total de contratos em aberto, uma vez que o tamanho do capital do investidor é relativo ao capital total existente:

$$PPGI(n) = \frac{Volume(n) * Preço(n) + PPGI(n-1) * \sum_{i=1}^{n-1} Volume(i)}{\sum_{i=1}^n Volume(i)}$$

Eq 3.3.1.1

Onde  $PPGI(n)$  é o preço médio da data  $n$ ,  $Volume(n)$  é a quantidade de volume negociado na data  $n$  e  $Preço(n)$  é o preço do ativo na data  $n$ .



### **3.3.2 Viabilidade**

O volume de negociação de um dia é dado pela quantidade do ativo, em números de contratos ou em valor financeiro, que é negociada no dia. Tanto o valor do volume quanto o do preço do ativo estão disponíveis em seus respectivos *sites* de suas instituições, como por exemplo, na BM&F ou no *site* do tesouro nacional. Para o caso das ações, será utilizado a *Bloomberg*, que é acessível pelo local de estágio no Banco Itaú BBA.

### **3.4 Títulos Públicos**

Os títulos públicos são ativos de renda fixa que se constituem em boa opção de investimento para a sociedade. Os títulos públicos possuem a finalidade primordial de captar recursos para o financiamento da dívida pública, bem como para financiar atividades do Governo Federal, como educação, saúde e infra-estrutura.

Para um exemplo de implementação será dado, considerando a ativo como sendo o título público NTN-F com vencimento em janeiro de 2012. A venda ocorre pelo próprio tesouro e é possível que a pessoa física compra ou venda o título seguramente pelo próprio site do tesouro nacional ([www.tesouro.fazenda.gov.br/tesouro\\_direto/](http://www.tesouro.fazenda.gov.br/tesouro_direto/)):



Fig. 3.4.1

Esta facilidade abriu as portas para os pequenos investidores, tornando o ativo bastante líquido e de fácil acesso para sociedade. Então ele foi o escolhido para o exemplo.

Este título público se caracteriza basicamente pela taxa de juros, que é estabelecida no momento da compra do contrato, que o proprietário do ativo irá receber. Então quanto maior a taxa de juros, melhor para o investidor.

A seguir, serão apresentados alguns dos títulos públicos que podem ser adquiridos e suas respectivas características.

### 3.4.1 Letras do Tesouro Nacional (LNTN)

Este título tem a rentabilidade definida no momento da compra. O investidor saberá exatamente quanto irá receber no futuro se permanecer com o título até o vencimento.

As Letras do Tesouro Nacional possuem fluxo simples de pagamento. Isso significa que se escolher esse título, o investidor receberá o valor investido mais os juros somente a data de vencimento do título. O importante

a destacar é que só se receberá a rentabilidade informada se o investidor permanecer até o final do vencimento.

Na prática, o valor que se receberá no futuro é fixo e denominado de lote, este com valor de mil reais. Então, o que se busca é pagar o menor preço à vista por este lote, deste modo, os juros a receber será maior.

O fluxo de caixa da compra de um lote deste título é representado por:

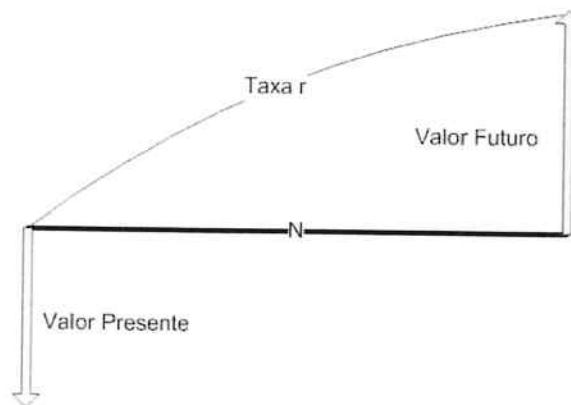


Fig. 3.4.1.1

E o valor presente é dado por:

$$Valor\ Presente = \frac{ValorFuturo}{(1+r)^N}$$

Eq. 3.4.1.1

Onde  $r$  é a taxa de juros do título e  $N$  o tempo até o vencimento em anos.

### 3.4.2 Notas do Tesouro Nacional – Série F (NTN-F)

Do mesmo modo que as Letras do Tesouro Nacional, este título tem a rentabilidade definida no momento da compra. Porém o investidor irá receber uma parte do rendimento a cada seis meses através de cupons semestrais. E no vencimento, receberá o valor investido mais os rendimentos de juros.

Os valores dos cupons são pré-estabelecidos e correspondem ao equivalente de 10% ao ano do valor futuro de cada lote, assim como a LTN, de mil reais.

A partir destas informações, chega-se ao seguinte fluxo de caixa:

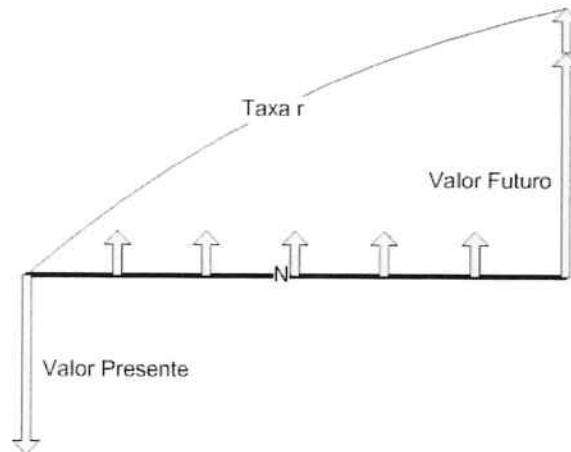


Fig. 3.4.2.1

Onde cada cupom tem valor futuro de:

$$Cupom = ValorFuturo \cdot ((1 + 0,1)^{0,5} - 1)$$

Eq. 3.4.2.1

E o valor presente é dado por:

$$Valor\ Presente = \frac{Cupom_1}{(1+r)^{N_1}} + \frac{Cupom_2}{(1+r)^{N_2}} + \dots + \frac{Cupom_n}{(1+r)^{N_n}} + \frac{ValorFuturo}{(1+r)^N}$$

Eq. 3.4.2.2

Onde  $N_i$  é o prazo de vencimento do Cupom  $i$  em anos e  $r$  é a taxa de rendimento anual do título.

A compra de títulos públicos pode ser feita em qualquer dia e hora. A venda já é mais restrita: apenas um ou dois dias na semana. Mas isto não o torna menos líquido, já que o vencimento costuma ser de anos.

Estes títulos são muito negociados no mercado, isto faz com que as taxas de juros deles variem diariamente. Se o investidor compra o papel a

juros alto, ele estará pagando um preço baixo. No momento que os títulos ficarem mais ofertados, os preços irão subir, ou seja, as taxas diminuir. Neste cenário, se outro investidor quiser comprar estes títulos do Tesouro, ele acordará uma taxa menor que a do primeiro investidor, ou seja, ele pagará mais caro por um papel que resultará num mesmo valor futuro. Como os títulos podem ser revendidos para o Tesouro semanalmente, se o primeiro investidor resolver vender, ele captará um recurso maior do que investido. É este o lucro que o sistema buscará.

### 3.5 Implementação

O sistema que está se pesquisando, irá apontar o momento em que a taxa está alta o suficiente para comprar, e na outra ponta, o momento em que se deve vender de volta ao tesouro.

Os dados de entrada no sistema, que são a taxa de juros histórica deste ativo e a posição detida pelo tesouro, são abertos e encontrados no site da Andima ([www.andima.com.br](http://www.andima.com.br)) neste formato (em planilha de Excel):

IMA-B - Série Histórica (Carteira)				
Titulos	Data de Referência	Data de Vencimento	Taxa Indicativa (% a.a.)	Quantidade (1.000 tit.)
NTN-F	29/5/2009	1/7/2010	9,4823	19.269,96
NTN-F	29/5/2009	1/1/2011	9,8601	9.239,78
NTN-F	29/5/2009	1/1/2012	10,8492	71.636,10
NTN-F	29/5/2009	1/1/2013	11,5086	10.067,19
NTN-F	29/5/2009	1/1/2014	11,7688	11.034,72
NTN-F	29/5/2009	1/1/2017	11,9749	27.403,41

Fig. 3.5.1

Onde é possível encontrar a taxa de juros que o título iria pagar se fosse negociado na data de referência e a quantidade que o tesouro nacional emitiu.

Para um exemplo de implemetação, serão utilizadas regras pesquisadas em conjunto com os analistas financeiros:

- Se Spread for negativo, posição é vendido.
- Se Spread for positivo, posição é comprado.



Onde Spread é dado pela diferença entre o preço do ativo e a resposta do filtro, e é considerado negativo quando:

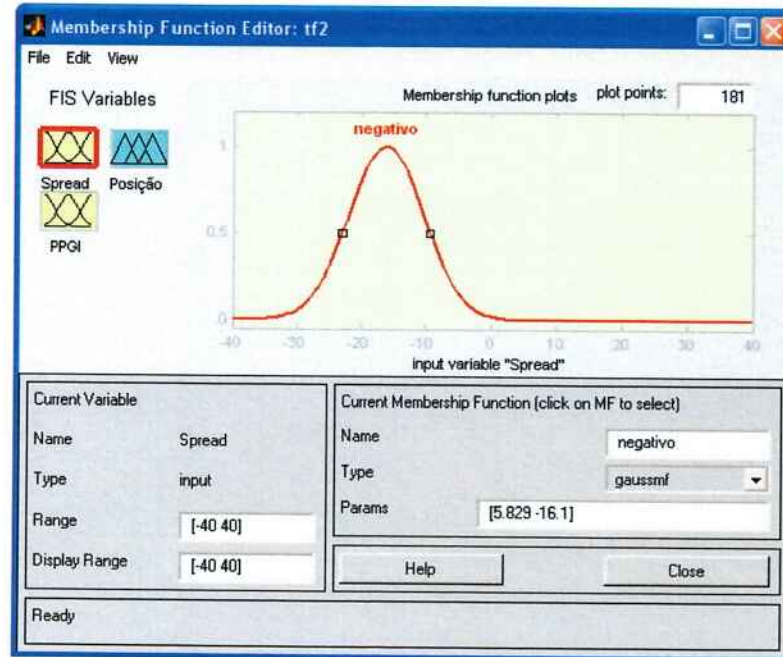


Fig. 3.5.2

E considerado positivo quando:

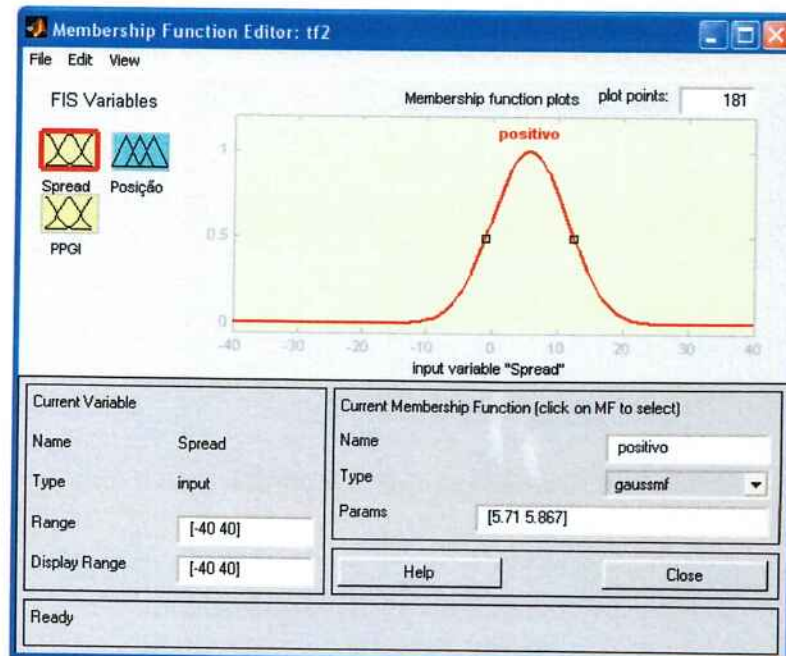


Fig. 3.5.3

Para os adjetivos de saída, foram considerados as seguintes funções:

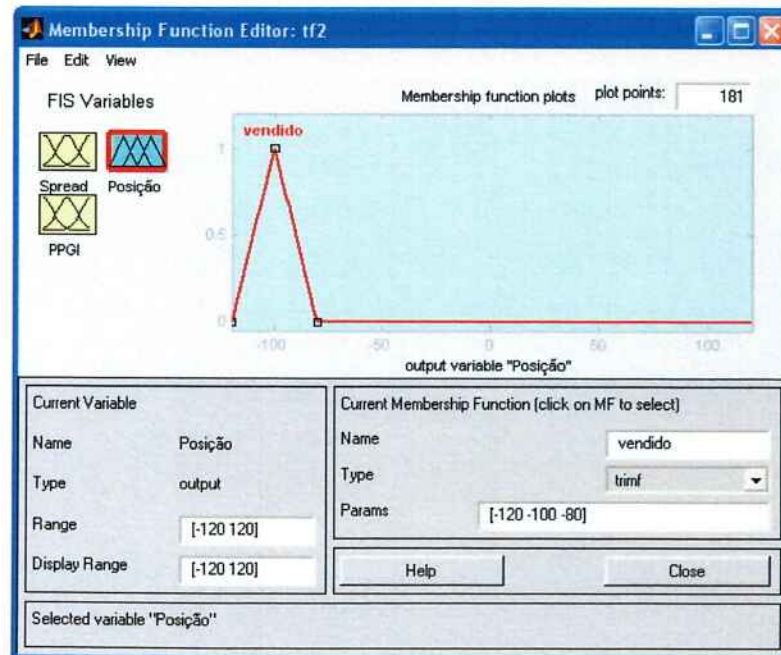


Fig. 3.5.4

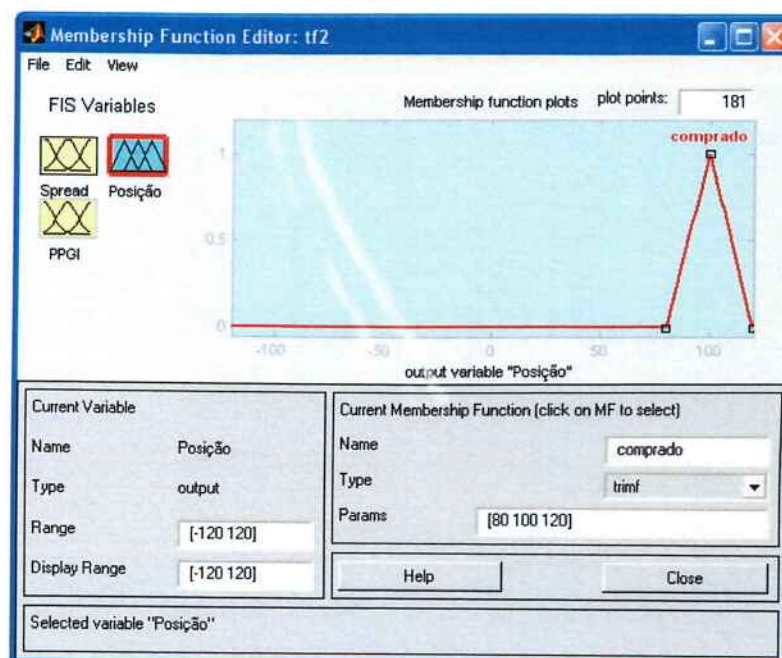


Fig. 3.5.5

Dados estes adjetivos e a seguinte regra:

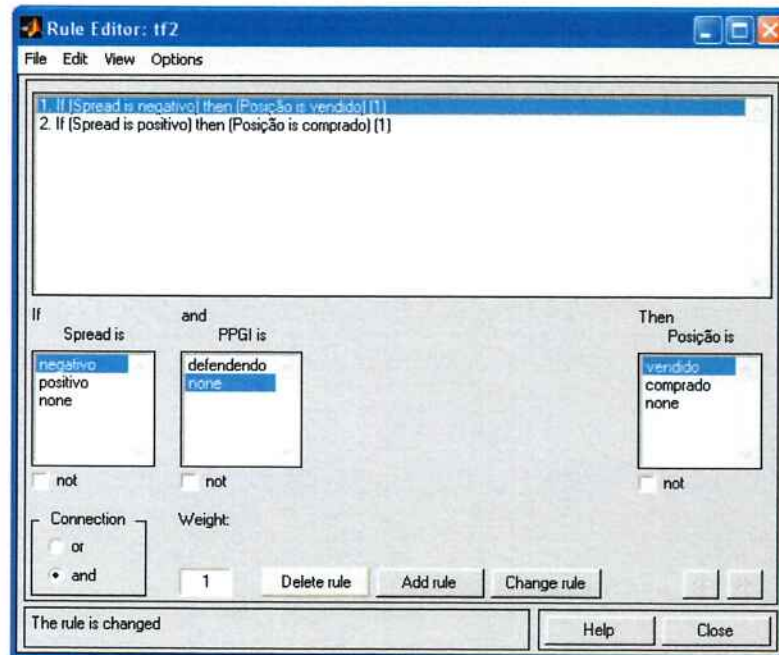


Fig. 3.5.6

Chega-se ao seguinte gráfico final do Spread pela posição:

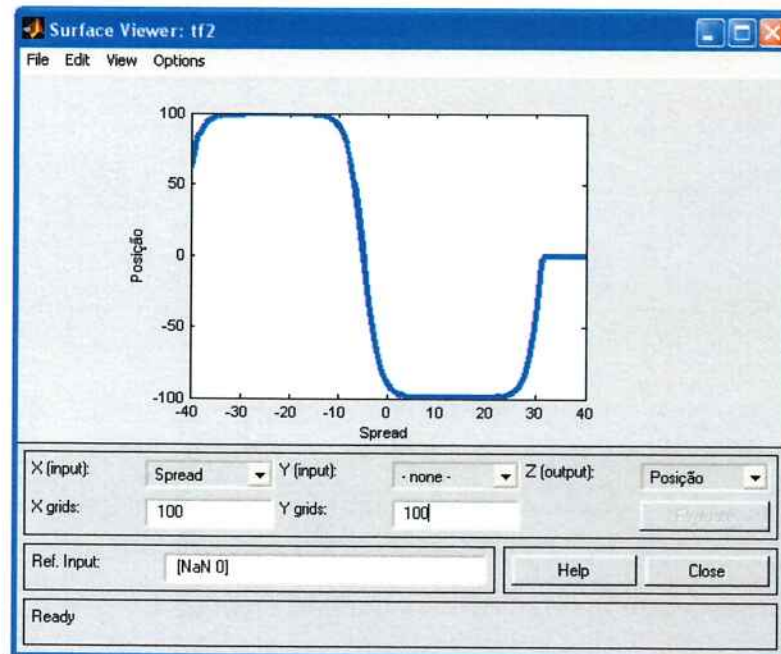


Fig. 3.5.7



Os pontos interessantes a observar são onde os Spreads são muito altos ou muito baixos, a posição é nula. Ou seja, nestas regiões, não se sabe a tendências do preço do ativo.

Para uma implementação inicial, foi utilizada a seguinte regra adicional: se o preço estiver alcançando o PPGI, o tesouro nacional, na tentativa de proteger o seu investimento, vai alocar o capital na tentativa de inverter a tendência.

Então, chega-se os seguinte gráfico do PPGI:

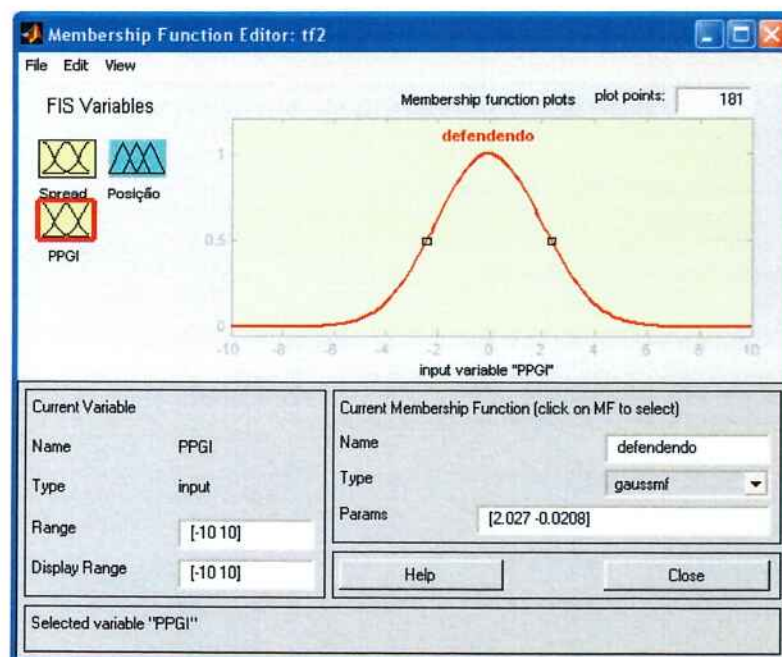


Fig. 3.5.8

Onde PPGI é a diferença do preço atual do ativo para o preço da posição do grande investidor.

Para as regras consolidando todos os adjetivos, foram utilizadas as seguintes sentenças:

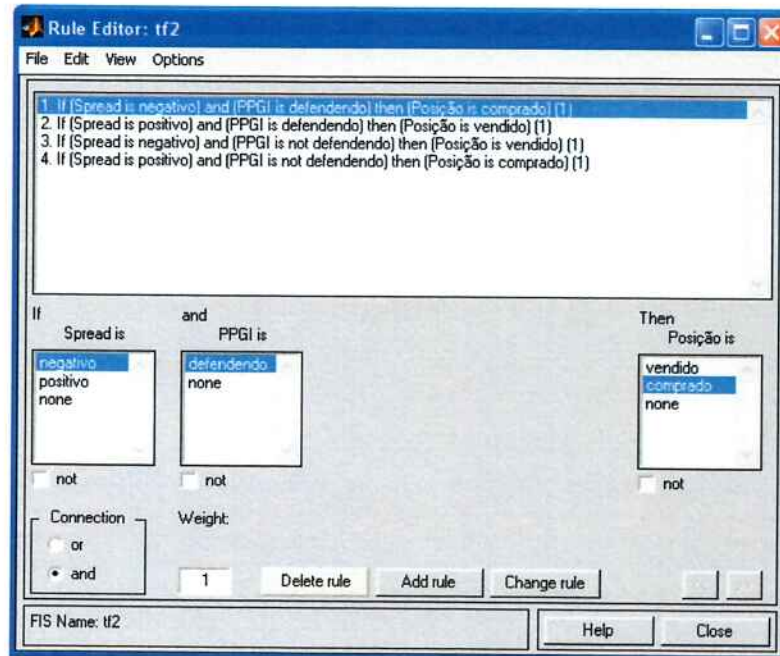


Fig. 3.5.8

Consolidando estas regras, chegamos à seguinte superfície:

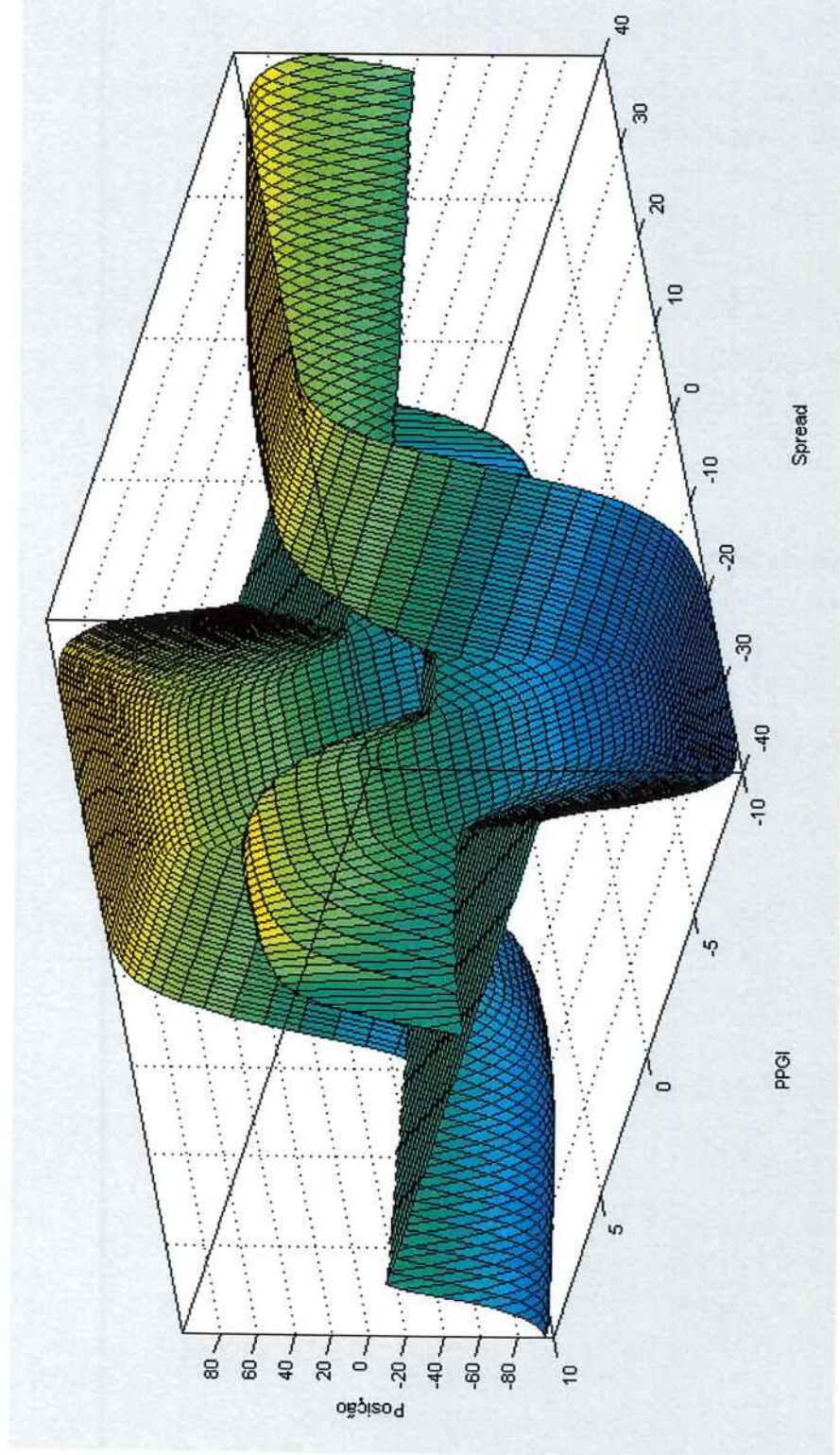


Fig. 3.5.9



Para implementar esta superfície no exemplo do tópico 3.1.4, é necessário obter a curva de PPGI através da equação 3.3.1.1. Graficamente chega-se às seguintes curvas:

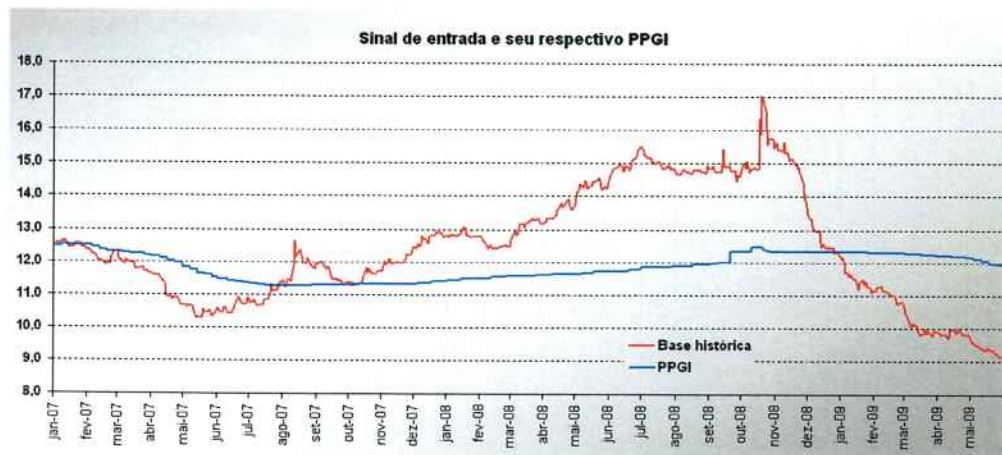


Fig. 3.5.10

A partir deste gráfico e do gráfico na figura 3.1.4.7, chega-se às seguintes curvas que servem de variáveis de entrada para a superfície 3.5.6:



Fig. 3.5.11

Aplicando estes sinais de entrada na superfície utilizando a função Retorna\_posicao no Anexo 3, chega-se ao seguinte gráfico de posição:



Fig. 3.5.12

Para checar a sua performance, utiliza-se a função de cálculo do Anexo 3. Observa-se que esta função é muito semelhante ao utilizado no cálculo do Anexo 1, a diferença relevante é a metodologia utilizada para o cálculo do resultado.

Graficamente, chega-se ao seguinte resultado:

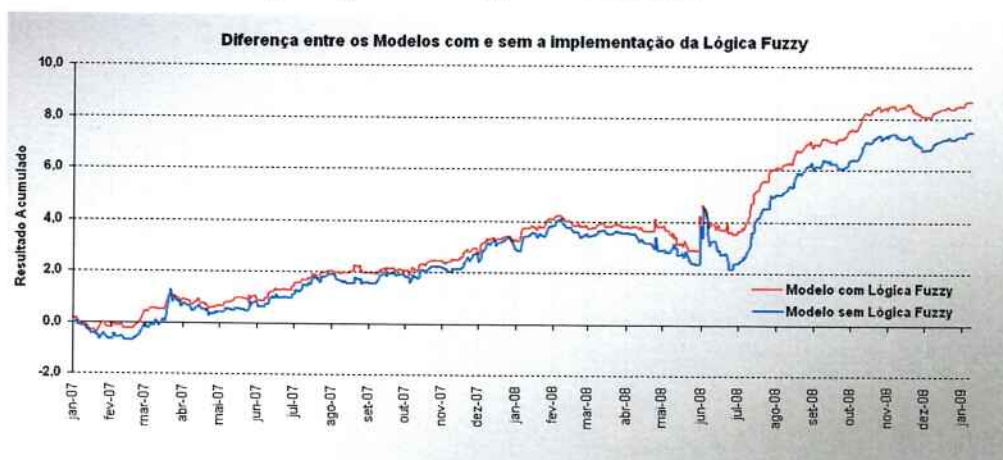





Fig. 3.5.13

Observa-se um melhor desempenho no modelo que utiliza a lógica fuzzy. Esta melhoria, neste exemplo, representa algo em torno de 10%. Ela pode ser explicada pela ponderação que a lógica Fuzzy faz nos momentos de indecisão do sistema. Quando isso ocorre, o modelo reage e reduz o risco para o investidor, diminuindo seu prejuízo.

## 4 CRONOGRAMA

	Revisão Bibliográfica			Implementação			Otimização		
	Filtro	Lógica Fuzzy	PPGI	Filtro	Lógica Fuzzy	PPGI	Filtro	Lógica Fuzzy	PPGI
Março									
Abril									
Maio									
Junho									
Julho									
Agosto									
Setembro									
Outubro									
Novembro									
Dezembro									

 Não realizado  
 Em elaboração  
 Concluído

## 5 DISCUSSÃO

Em busca do filtro que se encaixa no contexto, chegou-se ao filtro do tipo passa baixa, uma vez que se mostrou bastante eficiente na área de análise técnica. Para incrementá-lo, foi adicionada uma função para atribuir pesos para amostra que, supostamente, tem mais relação do que outras.

A ordem do filtro a ser implementado é um ponto a ser destacado. O comprometimento entre o tempo de resposta de detecção de tendência, que é menor para filtro de ordem menor, e a eficiência de detectá-la, que é maior para filtro de ordem alta, faz dessa questão fundamental.

Para encontrar os melhores parâmetros deste filtro, foi utilizado o programa MatLab, uma vez que este já foi utilizado nas matérias deste curso e se mostrou bastante eficaz. A robustez e a velocidade do MatLab foi essencial para a conclusão deste trabalho, já que o método utilizado para encontrar os parâmetros ideais foi o de busca exaustiva.

As regras da Lógica Fuzzy, e como elas vão interferir no sistema, são outros pontos decisivos para um funcionamento eficaz do sistema. São estes que vão interpretar os dados capturados e calculados, e devolver as sugestões mais lucrativas para o investidor. A otimização destas regras será essencial para a maximização dos rendimentos do sistema.

## 6 CONCLUSÃO

Seguindo rigidamente o cronograma pretendido no Memorial Descritivo, foram determinados o filtro e sua respectiva equação, com parâmetros otimizados por métodos computacionais, com o auxílio do MatLab.

Na área da Lógica Fuzzy, foram efetuadas pesquisas bibliográficas para adquirir um melhor conhecimento sobre o assunto, e analisar a Teoria de conjuntos Fuzzy, assim como suas nomenclaturas e conceito teórico.

Definiu-se uma ferramenta chave que servirá como regra para a Lógica Fuzzy. O preço da posição dos grandes agentes será fundamental para tomada de decisões do sistema em desenvolvimento.

Um ponto importante a citar é que a mesma metodologia pode ser utilizada para diferentes séries de preços a serem analisadas. Os parâmetros do filtro e os dados da Lógica Fuzzy serão diferentes de uma série para outra. Porém o sistema irá se auto-ajustar para cada sequência de amostras.



## 7 BIBLIOGRAFIA

- ABAR, C. O Conceito "Fuzzy". Disponível em: [www.pucsp.br/~logica/Fuzzy.htm](http://www.pucsp.br/~logica/Fuzzy.htm). Acesso em março de 2009.
- AGUIAR, R. A. Um Modelo Fuzzy Comportamental para Análise de Sobre-Reação e Sub-Reação no Mercado de Ações Brasileiro. Disponível em: [www.scielo.br/pdf/rae/v48n3/a02v48n3.pdf](http://www.scielo.br/pdf/rae/v48n3/a02v48n3.pdf). Acesso em abril de 2009.
- AMENDOLA, M. Manual do uso da teoria dos conjuntos Fuzzy no MatLab 6.5. Disponível em: [www.ime.unicamp.br/~laeciocb/manual\\_fuzzy\\_matlab.pdf](http://www.ime.unicamp.br/~laeciocb/manual_fuzzy_matlab.pdf). Acesso em: março de 2009.
- JUNGERS, L. C. D. Introdução a Lógica Fuzzy. Disponível em: [s2i.das.ufsc.br/tikiwiki/apresentacoes/logica-fuzzy.pdf](http://s2i.das.ufsc.br/tikiwiki/apresentacoes/logica-fuzzy.pdf). Acesso em março de 2009.
- SOUTO, M. Lógica Fuzzy. Disponível em: [www.dimap.ufrn.br/~marcilio/IA/IA2004.1/IA-aula-12-sistemas-fuzzy.ppt](http://www.dimap.ufrn.br/~marcilio/IA/IA2004.1/IA-aula-12-sistemas-fuzzy.ppt). Acesso em março de 2009.
- TANSCHKEIT, R. Sistemas Fuzzy. Disponível em: [www.ica.ele.puc-rio.br/Downloads/41/LN-Sistemas%20Fuzzy.pdf](http://www.ica.ele.puc-rio.br/Downloads/41/LN-Sistemas%20Fuzzy.pdf). Acesso em março de 2009.
- Tesouro Nacional. Disponível em: <http://www.tesouro.fazenda.gov.br/index.asp>. Acesso em Outubro de 2009.

## **Anexo 1**

### **Programa Principal**

```
%Importar vetor de base historica na variavel base  
%Inputar valores da ordem do filtro em n e vetores de peso em q para a  
iteração
```

```
i = 2;
```

```
j = 1;
```

```
max = 0;
```

```
while i <= n
```

```
    while j <= q
```

```
        %cria um vetor de pesos com i elementos
```

```
        peso = funcao_peso(i,j);
```

```
        %gera resposta do filtro para a base inteira dada a ordem do filtro
```

```
        %e o vetor de pesos
```

```
        resposta = filtro(base,i,peso);
```

```
        %calcula o resultado gerado pelo filtro implementado
```

```
        total(i,j) = calcula(base,resposta);
```

```
        %guarda os parametros do melhor filtro
```

```
        if total(i,j) > max
```

```
            max = total(i,j);
```

```
            imax = i;
```

```
            jmax = j;
```

```
        end
```

```
        %aumenta inclinacao da funcao de pesos
```

```
        j = j + 1;
```

```
    end
```

```

    %aumenta ordem do filtro
    i = i + 1;

    %reseta angulação da funcao de peso
    j = 1;
end

```

```

%guarda função de peso otimizada
peso = funcao_peso(imax,jmax);

```

```

%guarda resposta do filtro otimizado
resposta = filtro(base,imax,peso);

```

### **Função de Peso**

```

function [peso] = funcao_peso(m,q)

```

```

% funcao_peso(ordem do filtro m, indice q do vetor peso)
% cria um vetor linear onde o ultimo ponto vale  $1 + 0.1*(q-1)$  mais do que o
% primeiro

```

```

i = m;

```

```

while i > 0
    peso(i) =  $1 + 0.1*(q-1)*(i-1)/(m-1)$ ;
    i = i - 1;
end

```

### **Função do Filtro**

```

function [resposta] = filtro(base,ordem,peso)

```

```

% filtro(base historica, ordem do filtro m, vetor p de pesos)

```

```

% dada a base historica e os parametros do filtro,

```

```
% esta função gera um vetor de mesma escala que a base historica  
% com a resposta deste filtro
```

```
tamanho = length(base);  
n = 1;
```

```
% tratamento para as amostras iniciais  
while n < ordem  
    resposta(n) = mean(base(1:n));  
    n = n + 1;  
end
```

```
% resposta do filtro  
while n <= tamanho  
    resposta(n) = sum((base(n-ordem+1:n).*peso))/sum(peso);  
    n = n + 1;  
end
```

```
resposta = resposta';
```

### **Função do Cálculo**

```
function [total] = calcula(base,resposta)
```

```
% calcula(base historica, vetor repostas do filtro)
```

```
% dadas a base historica e a resposta do filtro,  
% esta funcao calcula o resultado que este filtro geraria ao investidor,  
% seguindo a logica de que se a repostas estiver abaixo da amostra, o ativo  
% esta com tendencia de alta e o investidor deve comprar o ativo e vice  
% versa
```

```
tamanho = length(base);  
n = 1;
```

```

while n < tamanho
    if resposta(n) == base(n)
        result(n) = 0;
    elseif resposta(n) < base(n)
        result(n) = base(n+1) - base(n);
    else
        result(n) = base(n) - base(n+1);
    end
    n = n + 1;
end

```

```
total = sum(result);
```

### **Comandos Geradores de Gráfico**

```

mesh(total)
pcolor(total)
colorbar
xlabel('Indice do peso')
ylabel('Ordem do filtro')
title('Grafico de ganhos em função da ordem do filtro e a função de peso')

surf(total)
xlabel('Indice do peso')
ylabel('Ordem do filtro')
title('Grafico de ganhos em função da ordem do filtro e a função de peso')

```

## **Anexo 2**

### **Cálculo do resultado real**

```
%Importar vetor de base historica na variavel base_total
%Inputar valores da ordem do filtro em n e vetores de peso em q para a
%iteração

% Colocar em t o inicio do teste de performance
t = 100;

tamanho = length(base_total);

while t < tamanho

    base = base_total(1:t);

    principal

    vetor_imax(t) = imax;
    vetor_jmax(t) = jmax;

    if resposta(t) == base(t)
        result_real(t) = 0;
    elseif resposta(t) < base(t)
        result_real(t) = base_total(t+1) - base(t);
    else
        result_real(t) = base(t) - base_total(t+1);
    end

    t = t + 1;

end

result_real = result_real';
```

```
vetor_imax = vetor_imax';  
vetor_jmax = vetor_jmax';
```



### Anexo 3

#### Função que Retorna Posição dada a superfície

```
function [posicao] = retorna_posicao(spread,ppgi,fis)
% dada os pontos de spread e ppgi, retorna a posição resultante da
% superfície fuzzy
% carregar superfície em fis com comando fis = readfis
% colocar valores procurados em spread e ppgi

% carrega superfície fuzzy
[a,b,c] = gensurf(fis);

n = 1;

while n <= length(spread)

    % encontra coordenadas do ponto mais próximo do valor de spread
    [i,j] = find(abs(a-spread(n))==(min(min(abs(a-spread(n))))));

    % encontra coordenadas do ponto mais próximo do valor de ppgi
    [k,l] = find(abs(b-ppgi(n))==(min(min(abs(b-ppgi(n))))));

    % captura apenas 1 elemento do vetor
    j = j(1,1);
    k = k(1,1);

    % retorna posição
    posicao(n) = c(k,j);

    n = n + 1;

end

posicao = posicao';
```

### **Função do Cálculo com implementação Fuzzy**

```
function [total] = calcula_fuzzy(base,posicao)

% calcula(base historica, vetor repostas do filtro)

% dadas a base historica e a resposta do filtro,
% esta funcao calcula o resultado que este filtro geraria ao investidor,
% seguindo a logica de que se a repostas estiver abaixo da amostra, o ativo
% esta com tendencia de alta e o investidor deve comprar o ativo e vice
% versa

tamanho = length(base);
n = 1;

while n < tamanho
    result(n) = posicao(n) * (base(n+1) - base(n));
    n = n + 1;
end

total = sum(result);
```