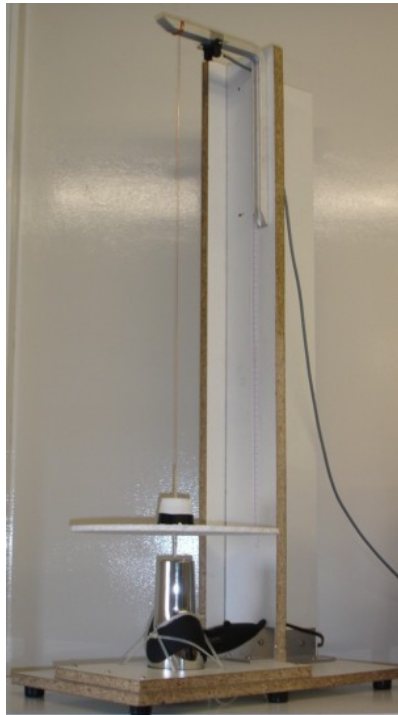


Universidade do Porto

FEUP Faculdade de Engenharia

Sistemas Baseados em Lógica Difusa
2004/2005

**“Levitação de um disco por variação de
velocidade de um Ventilador”**



Intervenientes:

Ana Luísa Martins

Fernando Manuel Gonçalves Santos

ee04255@fe.up.pt

ee01136@fe.up.pt



ÍNDICE

1. Introdução e objectivos	3
2. Dificuldades na concepção do kit	4
3. Controlo Fuzzy	6
3.1. Porquê o uso de Lógica Difusa?	6
3.2. Raciocínio tomado pela Lógica Difusa	7
3.3. Controlo do sistema em causa por lógica Difusa	8
3.3.1. Fuzzificação do sistema	8
3.3.2. Inferências	11
3.3.3. Desfuzzificação	12
4. Interface desenvolvida em Matlab	14
5. Interligação Matlab-Atmega	15
6- Evolução temporal	17
7- Controlo PD	19
8. Apreciações e Conclusões finais	20
9. Anexos	22



1. Introdução e objectivos

Este trabalho tem como objectivo a levitação de um disco por variação de velocidade de um ventilador. O nosso trabalho global encontra-se dividido em três partes, pois foi alvo em três cadeiras deste semestre. Assim temos:

- Modelação do Sistema (correspondente à cadeira de Sistemas de Accionamento e Movimentação)
- Sistema Físico de Controlo, nomeadamente interface de controlo realizada com o microcontrolador Atmega8 (cadeira de Sistemas Baseados em Microcontroladores)
- Método de Controlo por lógica difusa (cadeira de Sistemas Baseados em Lógica Difusa)

A nível de SBLD este trabalho tem particular interesse pois ajuda nos a integrar os conceitos abordados nas aulas teóricas num processo físico real. Desta forma adquirimos conhecimento de mais um método de controlo que futuramente nos poderá auxiliar em novos projectos em que tenhamos de decidir qual o melhor método de controlo a adoptar.



2. Dificuldades na concepção do kit

Em busca do kit ideal...

Dificuldades relevantes que surgiram com a concepção do kit:

Objecto a controlar:

- 1ª tentativa- cone feito em cartolina- revelou-se muito pesado e inapto para o controlo pois além de ser pesado oscilava muito.
- 2ª tentativa- copos de plástico- não resultou
- 3ª tentativa- cone feito com penas de pato- problema do peso resolvido, mas obtivemos um efeito "hélice" que dificultava o controlo.
- 4ª tentativa- voltamos ao cone em cartolina e reparamos que o que dificultava o controlo além de outros factores era o facto de não termos 2 pontos do cone a rolar sobre o eixo, o que o fazia oscilar muito (pelo 1º axioma de Euclides - "Dois pontos definem uma recta"). Verificamos também neste ponto que o eixo não era o ideal.
- 5ª tentativa- mudança de eixo- inicialmente tínhamos um tubo fino que depois de algumas experiências mostrou um forte atrito e muitas irregularidades na sua superfície, então substituímos esse tubo por fio de cobre. Mesmo assim o cone em cartolina mostrou-se pesado.
- 6ª tentativa- cone feito em papel fino- com a colocação de uma pequena base de copo de plástico circular no topo do cone obtivemos 2 pontos a rolar sobre o eixo, colocamos ainda uma carga vazia de esfera gráfica a unir estes 2 pontos. Com estas alterações parecia que tínhamos o sistema pronto a controlar.

Depois de começarmos a desenvolver o controlo, o ventilador AC avariou e o nosso kit teve de sofrer mais alterações...

Assim, o motor AC foi substituído por um secador, embora sendo um objecto utilizado principalmente em kits didácticos e não parecendo a melhor opção revelou-se mais adaptável ao nosso controlo que o ventilador inicial AC, visto que ao colocar o cone em baixo ou em cima com uma variação pequena de velocidade, não havendo posições intermédias. Com a colocação do secador este aspecto melhorou bastante, mas mesmo assim o cone mostrou uma fraca dinâmica para o nosso controlo, devido ao seu peso demasiado leve e à sua forma era difícil manter numa posição, pois balanceava. Então optámos por mudar o objecto a controlar e depois de algumas experiências verificamos que um disco leve, mas rígido seria a melhor opção. Desta forma e com estas restrições o melhor material que arranjámos foi a esfera de vidro.

O Disco que fizemos (Diâmetro=30 cm) resultou, contudo surgiu um problema com a placa que disparava o triac. Ao fim de algum tempo perdíamos o controlo na posição mais veloz no secador... inicialmente pensamos que o problema seria do ruído eléctrico à





entrada do microprocessador. Disparamos então o triac somente em meia onda e notamos que já não se verificavam com tanta intensidade esses ruídos (tivemos que fazer outro disco de diâmetro mais pequeno, pois o secador não tinha potência suficiente para levantar o disco de maior diâmetro). Porém, isto foi "solde pouca dura", pois passado algum tempo deparamo-nos com o mesmo problema. Depois de termos posto tudo em causa sem saber de onde era o problema... e sem soluções físicas a surgirem, voltamos à electrónica e depois de mais algumas "cabeçadas" e com as dicas dos nossos colegas descobrimos o problema. O triac ao fim de algum tempo saturava ficando a sua temperatura elevadíssima, motivo pelo qual e com toda a razão de ser perdíamos o controlo. Com a colocação de um dissipador no triac resolvemos definitivamente (até à data) o problema e chegamos assim à forma final do nosso kit.





3. Controlo Fuzzy

3.1. Porquê o uso de Lógica Fuzzy?

Os conhecimentos disponíveis que temos sobre um sistema qualquer são geralmente imprecisos. A lógica binária que apenas admite dois valores lógicos (verdade '1' e falso '0') revela-se incapaz de traduzir um bom funcionamento de um sistema em que reina a incerteza, pois para uma dada variável podem existir 'meias verdades'. Foi com base nesta ineficiada lógica binária que nasceu a lógica difusa. Esta lógica funciona de modo similar à capacidade do raciocínio humano, de desenvolver processos lógicos a partir de expressões pouco precisas, ou seja, difusas que variam entre os valores lógicos '0' e '1'.

Com lógica difusa não tratamos uma variável como tendo apenas um estado actual, mas sim com 'n' estados, cada um deles com determinado grau de veracidade.

Assim não afirmamos que uma pessoa é alta, mas sim que ela é 0.8 alta, 0.4 média e 0.1 pequena. Daqui surgem conjuntos em que um dado valor pode ser enquadrado. Quanto maior o nº de conjuntos maior a precisão da variável.



3.2. Raciocínio tomado pela Lógica Difusa

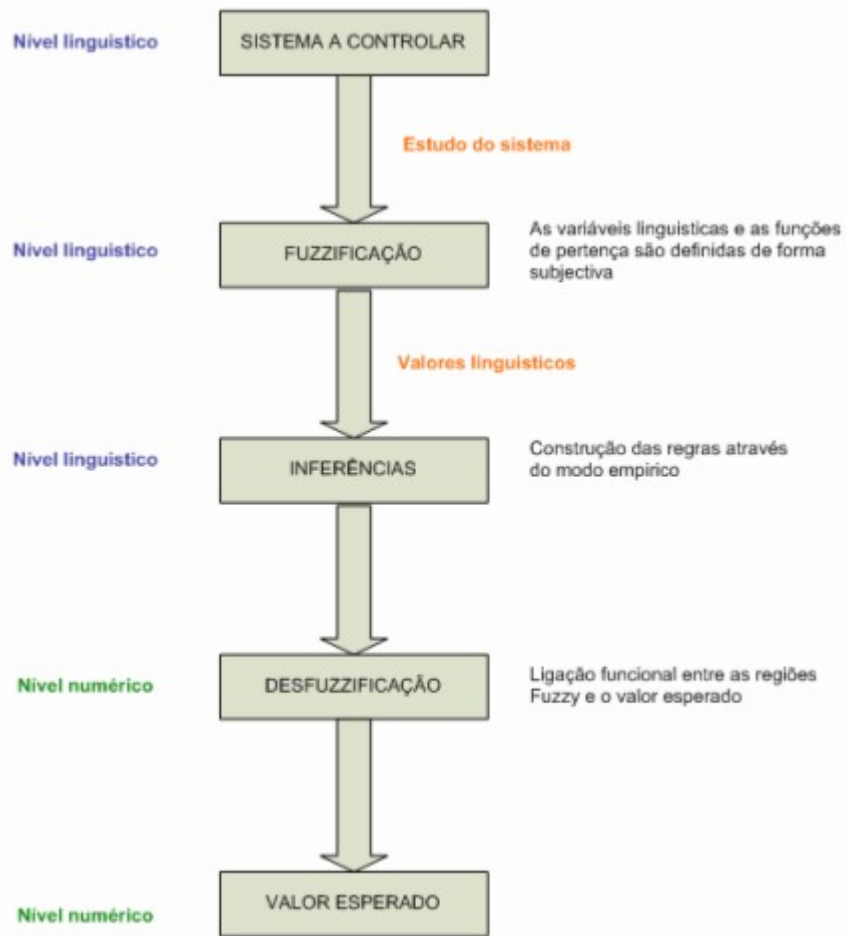


Fig.1- Raciocínio da lógica Difusa

3.3. Controlo do sistema em causa por lógica difusa

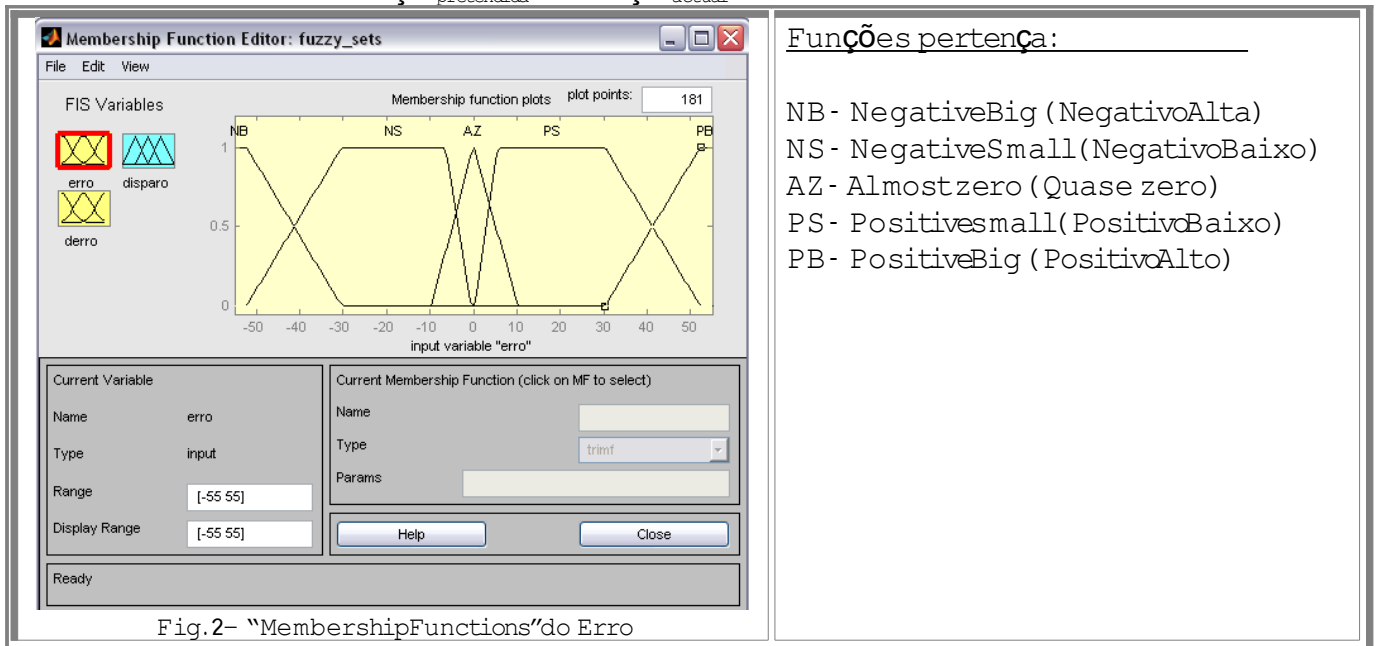
Para realizar o controlo efectivo do processo optámos por utilizar a ferramenta "Matlab", devido à nossa ambientação com a mesma e a facilidade de uso que esta nos proporciona trabalhar com a lógica difusa e construção de interface gráfica.

O Matlab é bastante versátil e dá-nos a possibilidade (tal como o "Scilab") de criar de modo quase automático conjuntos difusos ("Fuzzy Sets") de entradas/saídas e suas respectivas funções pertença ("Membership Functions"). É nos ainda possível definir regras que regem este sistema de entradas/saídas.

Depois de um estudo prévio comportamental do nosso sistema procedemos à sua validação por lógica difusa.

3.3.1. Fuzzificação do sistema:

Entrada I → Erro = Posição_{pretendida} - Posição_{actual}



Esta variável encontra-se compreendida entre [55 55] cm, sendo estes os valores extremos a que pode tomar o erro. Note-se ainda as irregularidades entre as "Membership Functions", isto deve-se de facto à nossa optimização do próprio controlo, tentando contrariar ao máximo os erros quando o disco está muito próximo da posição pretendida.

Entrada II → Derro = $\frac{\text{erro}_{\text{anterior}} - \text{erro}_{\text{actual}}}{\text{time}}$
time - tempo de uma amostra (obtido por tic, toc em matlab)

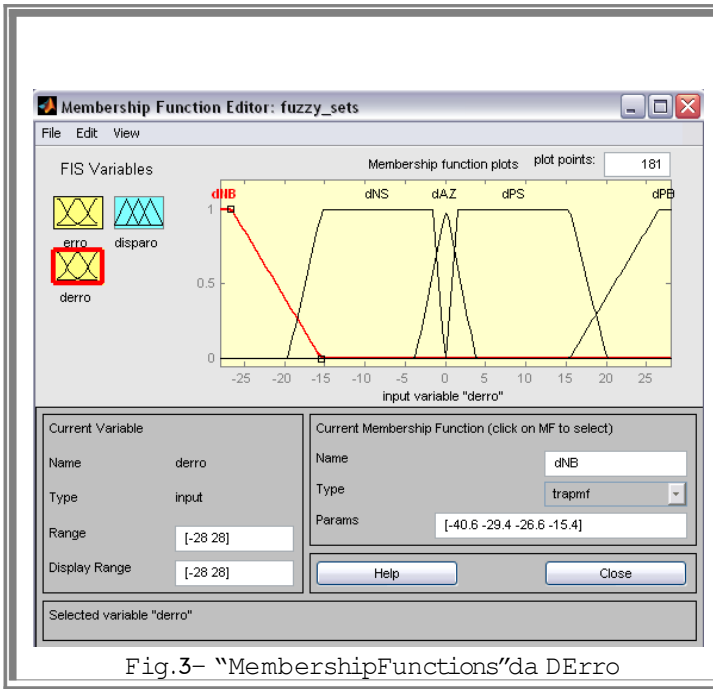


Fig.3- "MembershipFunctions"da DErro

Funções pertença:

- dNB- Derivative Negative Big (Derivada Negativa Alta)
- dNS- Derivative Negative Small (Derivada Negativa Baixa)
- dAZ- Derivative Almost Zero (Derivada Quase Zero)
- dPS- Derivative Positive Zero (Derivada Positiva Baixa)
- dPB- Derivative Positive Big (Derivada Positiva Alta)

Da mesma forma que para o erro, procedemos de novo à optimização da influência desta variável no controlo. Esta variável mostrou-se particularmente decisiva na prevenção de grandes velocidades adquiridas pelo disco.

SAÍDA I → Disparo

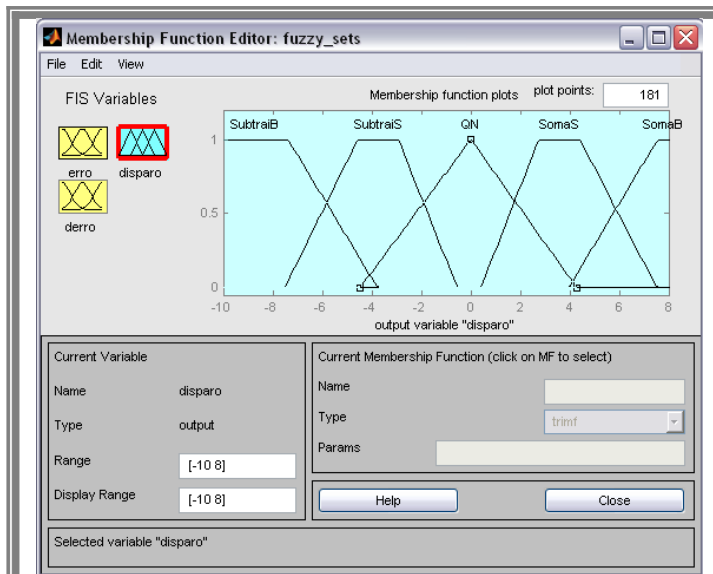


Fig.4- "MembershipFunctions"do Disparo

Funções pertença:

- Subtraib - Subtracção Grande
- Subtrais - Subtracção Pequena
- QN - Quase nada
- SomaS - Soma pequena
- SomaB - Soma grande



Nesta variável há a realçar a característica relativa da mesma, isto é, o valor transmitido ao microcontrolador não é o resultado absoluto da desfuzzificação, mas sim a soma do valor de disparo anterior e o resultado da desfuzzificação actual. Reforçando a ideia anterior, o valor de uma desfuzzificação varia entre -10 e 8, esse valor soma-se ao disparo anteriormente enviado ao "micro" formando assim um novo valor de disparo.

Note-se mais uma vez, que as "Membership Functions" da saída são específicas para o nosso sistema, não sendo assim aplicáveis a outros sistemas, salvaguardando possíveis coincidências.

3.3.2. Inferências

Agora vamos apresentar as regras que regem o controle do nosso sistema.

Formulação das regras:

Se "Erro" tem um valor positivo, isto significa que o disco se encontra abaixo da posição pretendida, ao invés se $Erro < 0$, o disco encontra-se acima da posição desejada.

Da mesma forma se $Derro > 0$ o disco está em queda, e se $Derro < 0$ o disco está a subir.

Esquemáticamente:

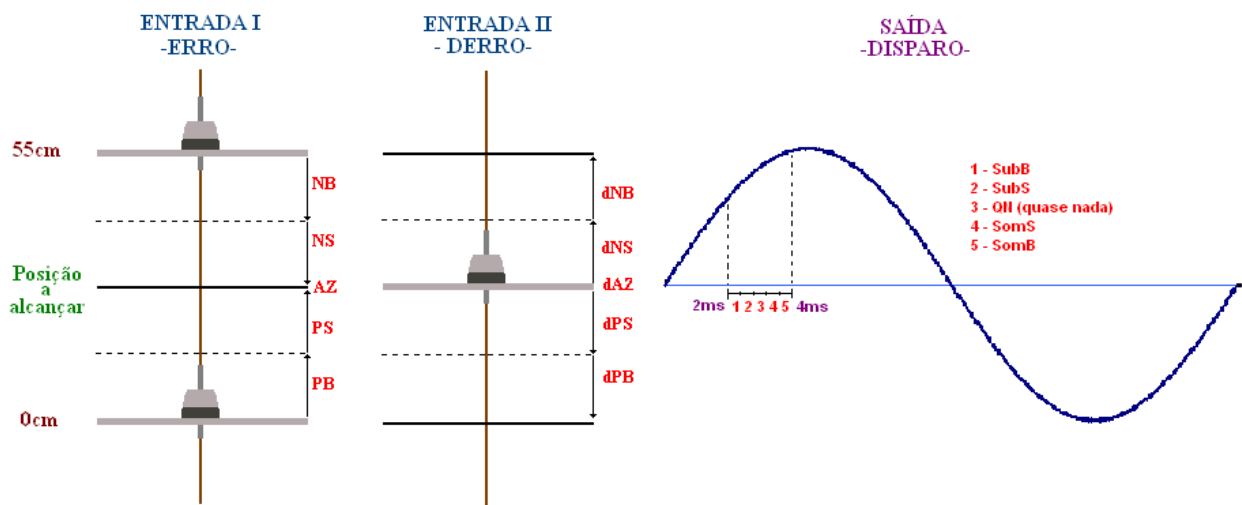


Fig. 5- Esquema de dedução das regras

Regras formuladas

Regras do tipo:

"If ERRO is x and DERRO is y then DISPARO is k"

ERRO						
		NB	NS	AZ	PS	PB
D	dNB	SubtraiS	SubtraiS	SubtraiS	SubtraiS	SubtraiB
E	dNS	QN	QN	SubtraiS	SubtraiS	SubtraiS
R	dAZ	SomaS	SomaS	QN	SubtraiS	QN
R	dPS	SomaS	SomaS	SomaS	QN	QN
O	dPB	SomaB	SomaS	SomaS	SomaS	SomaS

Tab. 1 – Regras fuzzy

3.3.3. Desfuzzificação:

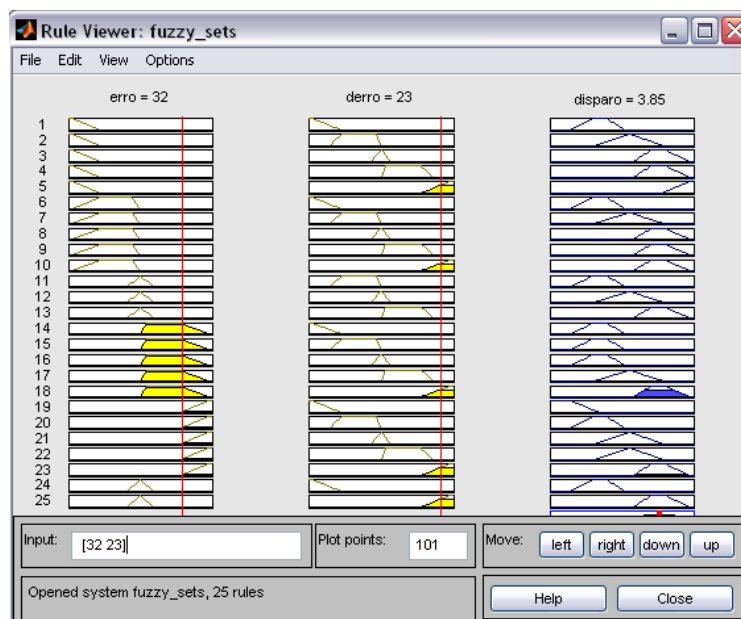


Fig.6 - Visualização das regras obtidas:

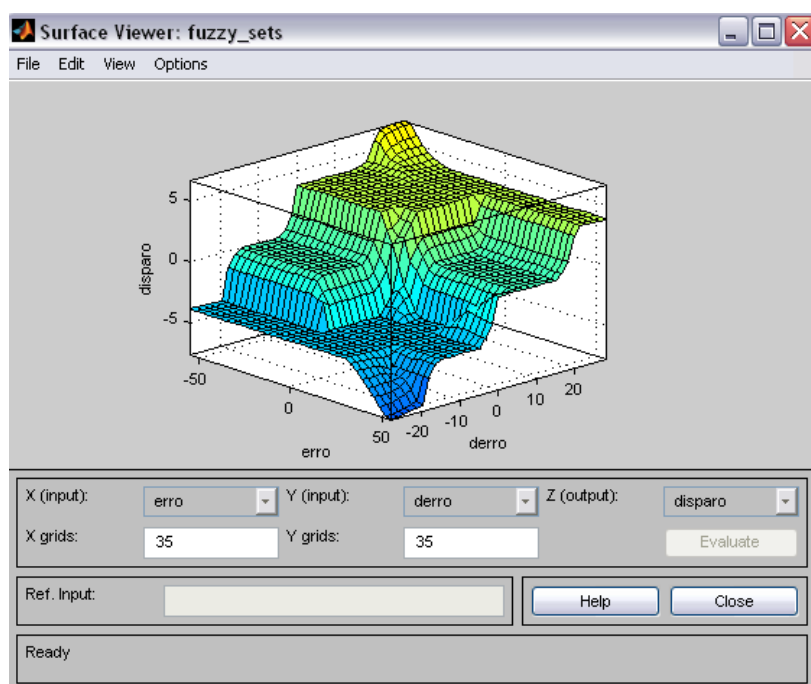


Fig.7- Superfície de controlo obtida

Devido ao facto do sistema que temos de controlar ser não linear, as regras serão forçosamente "não-simétricas", ou seja, o comportamento do disco na descida é completamente diferente do comportamento na subida. Nota-se como seria de esperar que na descida, o disco ganha uma velocidade muito superior à do sentido oposto. Nota-se ainda que o facto que a distância do disco ao ventilador é particularmente importante pois é completamente distinto o controlo a posições muito baixas e posições altas. Para melhor compreensão deste fenómeno, propomos a leitura do anexo 2 (ainda em construção!!).



4. Interface desenvolvida em Matlab

Este é o aspecto da nossa interface, é possível ao utilizador introduzir a posição pretendida a alcançar:

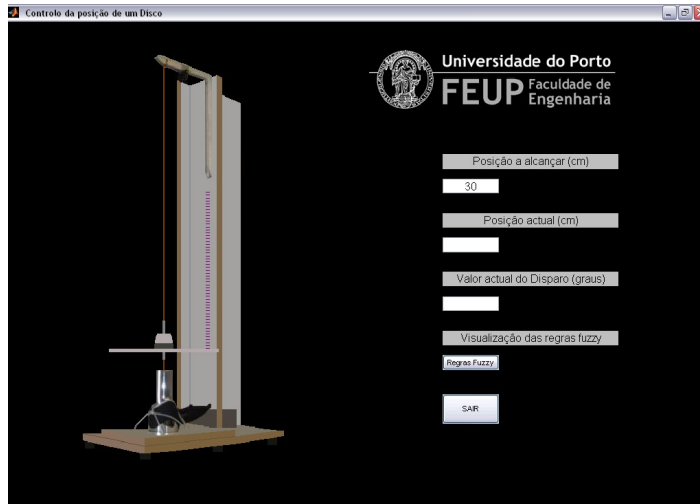


Fig.8 - Interface inicial em Matlab

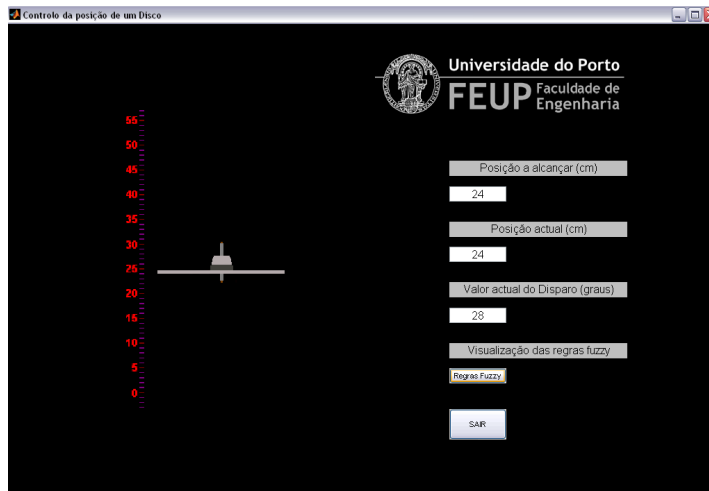


Fig.9 - Interface durante o controlo

Assim que é introduzida a posição a alcançar o processo de controlo desencadeia-se até estabilizar a posição pretendida, dando ao utilizador a cada momento, embora com um "pequeno" delay em relação ao Atmega8 a posição actual do disco e também o valor do disparo do triac.



5. Interligação Matlab - Atmega

Neste ponto vamos apresentar a ligação que estabelecemos entre o Matlab e Atmega.

Para ligarmos o PC ao Atmega, recorremos á comunicação série devido a não termos um grande volume de informação e também devido á sua facilidade de utilização.

Para interligarmos o Matlab com o Atmega tivemos que estabelecer de certa forma um "mini" protocolo de comunicação. Desta forma vamos apresentar um diagrama para explicar este "mini" protocolo:

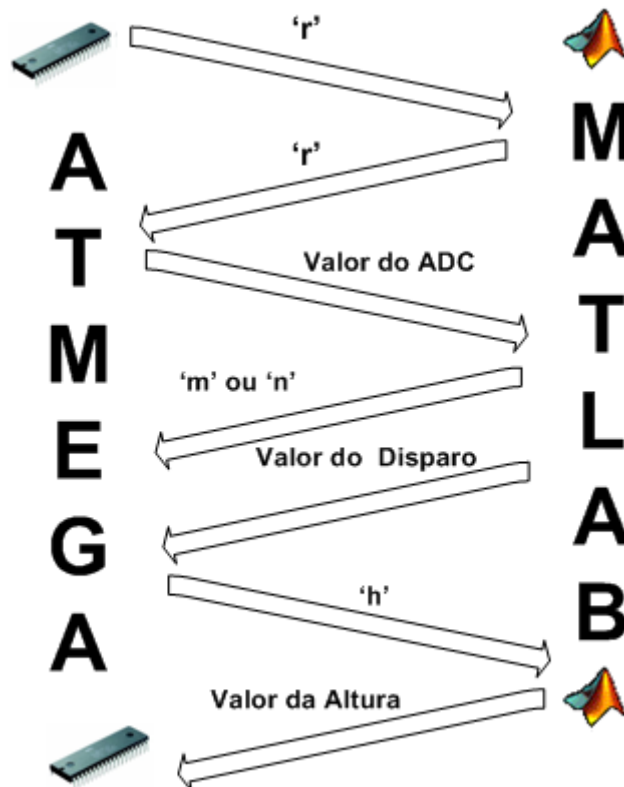


Fig.10 - Protocolo entre o Matlab e o Atmega

Sintetizando a figura, o Atmega envia um 'r' quando o Matlab responde, dá-se início á transferência de comando. De seguida o Atmega responde com o valor médio de ADC.

Neste momento o Matlab trata esta informação, calculando o erro de erro. A partir destes valores calcula o valor do disparo. Este ultimo calculo resume-se a uma pergunta á variável "fuzzy_sets" que foi criada com "Fuzzy Logic Toolbox". O resultado desta pergunta é adicionado ao valor anterior do disparo, formando-se assim o novo disparo.

Agora o Matlab verifica se tem um disparo com 2 ou três dígitos. No primeiro caso transmite um 'n' e no segundo transmite um 'm'. Este envio serve para indicar ao Atmega quantos algarismos vai enviar. Note-se que para enviar um algarismo com mais de um dígito, esse algarismo é partido em dígitos para ser enviado. Por ex. o numero 123, seria partido em 1, 2, 3, e enviados separadamente.

De seguida envia o disparo.



Após isto o Matlab espera um 'h' que significa que o Atmega está a esperar de uma altura. Neste momento o Matlab mais uma vez parte a altura em dois algarismos e envia-os separadamente.

Termina desta forma a transmissão de controlo. A partir deste momento o Matlab está preparado para uma nova transmissão.



6 - Evolução temporal

Neste ponto vamos apresentar a resposta do disco face a uma posição a controlar.

Para esta figura foi imposto um controle a altura 30, e no momento que o Matlab recebia a altura actual (enviada pelo Atmega), gravava o seu valor e o valor do tempo que demorava a fazer esta leitura. Sendo assim obtivemos o seguinte gráfico:

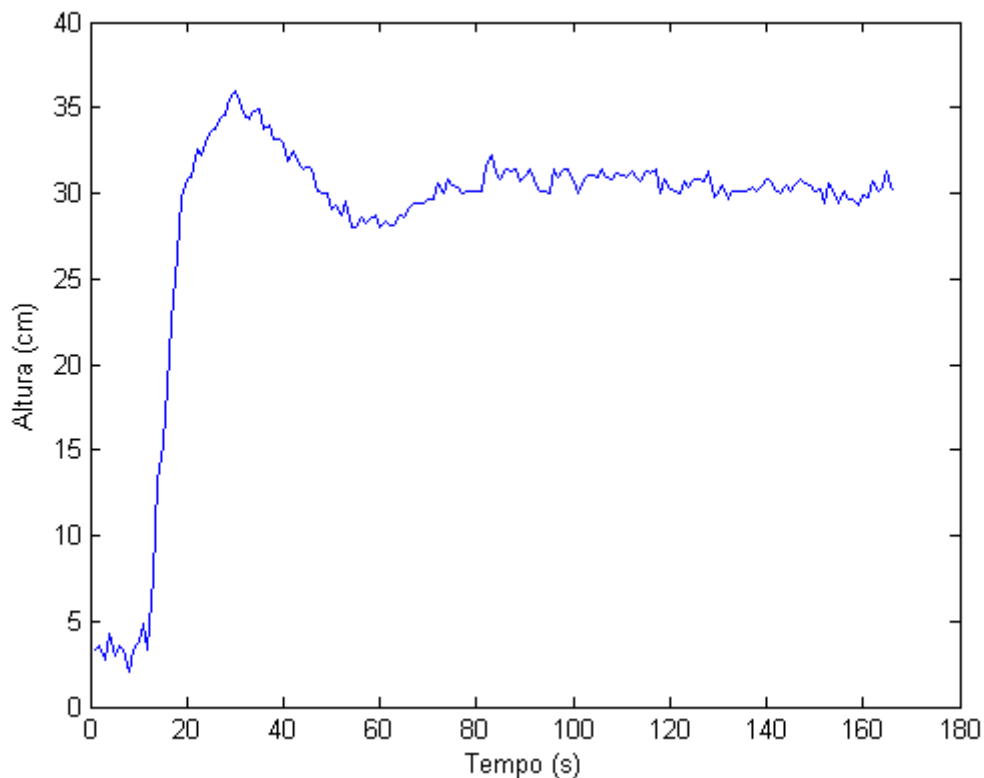


Fig. 11 – Evolução temporal do processo de controle

Note-se que de facto o controle é muito lento. Depois de fazer um rastreio verificou-se que isto se deve ao facto de o Matlab ser uma ferramenta de matemática e não uma ferramenta de controle. Tentamos ainda ir mais longe para verificar onde é que o Matlab demorava este tempo todo, e descobrimos que a comunicação séria demorava cerca de 1 segundo. Este problema é grave, pois o nosso controle fica severamente limitado, pois só de 1 em 1 segundo podemos exercer algum controle sobre o sistema. Isto origina que a nossa acção seja muito limitada, para que não fora mais certa e possa compensá-la.

Mesmo com este contratempo o controle foi possível.

Agora vamos apresentar a resposta do nosso sistema a diferentes alturas de referência.



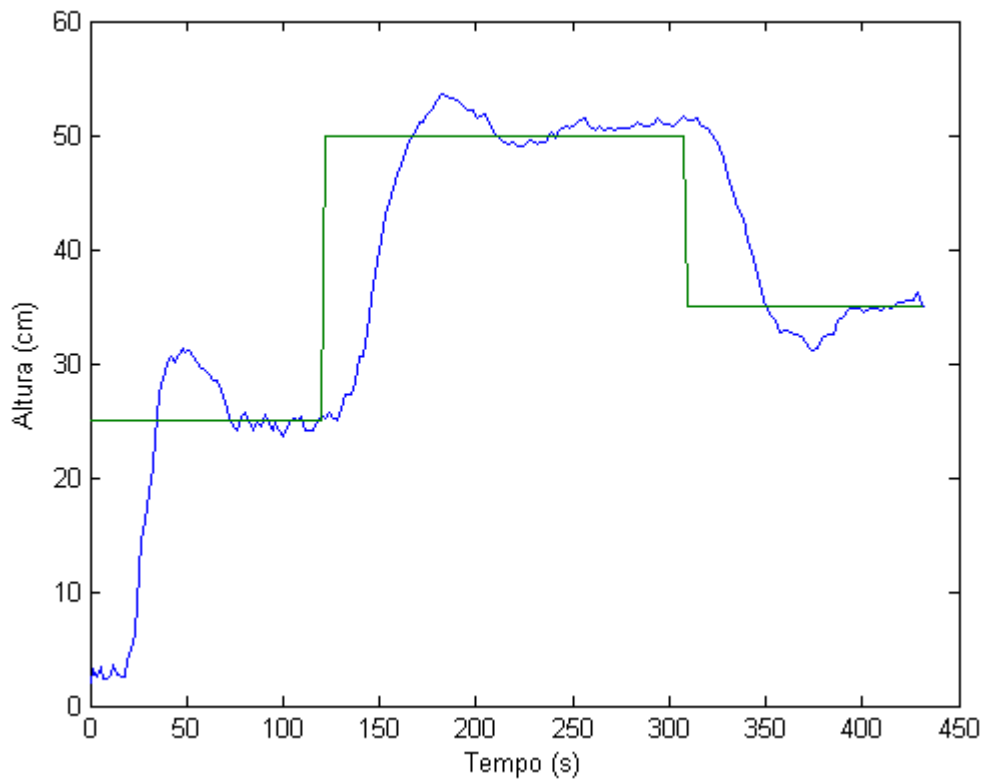


Fig.12 - Evolução temporal do processo de controle para diferentes alturas

Como se pode ver o disco conseguiu acompanhar os valores de referência ainda que com bastante tempo de atraso.

7 - Controlo PD

Como este trabalho era único para três cadeiras, foi nos proposto a nível de SAM, o controlo do mesmo processo com um controlador PD. Em primeira mão, tentamos implementar este controlador no Matlab, substituindo assim o bloco referente à Lógica Difusa. Porém rapidamente verificamos que não conseguíamos controlar o processo. No entanto, tentamos implementar o controlador a nível do Atmega, sem qualquer dependência do PC. Podemos dizer que resolvemos assim o problema que tínhamos com a comunicação série, pois não era necessária qualquer informação do PC.

O resultado foi bastante razoável em menos de 30 segundos conseguimos estabilizar o disco na posição pretendida. Por muita pena nossa, não temos nenhum gráfico para demonstrar o seu comportamento.

Teria sido interessante implementar o controlo de lógica difusa a nível do Atmega, dispensando da mesma forma o acesso ao PC. Tornaria o processo de controlo significativamente mais rápido.

Para uma melhor compreensão deste controlador propomos a leitura do relatório de SBM que já vai em anexo.

8. Apreciações e conclusões finais

Ao englobarmos este trabalho em três cadeiras permiti-nos verificar vários aspectos acerca do nosso trabalho com algum pormenor. Alguns deles mais perceptíveis do que outros.

A nível da física pudemos estudar alguns dos fenómenos envolvidos que caracterizam o nosso sistema, o que nos ajudou a perceber alguns resultados obtidos a nível de controlo, que inicialmente não percebíamos o porquê. Para melhor compreensão propomos a leitura do relatório de SAM, que vai ser entregue posteriormente como anexo.

A nível de Hardware, foi nos possível observar diversos aspectos. Um deles é que é necessária uma montagem cuidadosa, de forma a obter o melhor desempenho do kit, minimizando assim a força de atrito entre o disco e o fio que o conduz. Note-se ainda a importância de ter o sensor perfeitamente alinhado com o kit, de forma dar medições correctas. Foi nos também possível verificar que é muito mais fácil controlar a posição de um disco do que um cone, como tinha sido inicialmente proposto.

A nível de electrónica foi bastante interessante pois pudemos usar alguns



conhecimentos relativos a electrónica de potência e electrónica em geral, tanto para perceber o funcionamento de uma placa (triac) que gentilmente nos foi cedida pelo docente da cadeira "Sistemas Baseados em Tempo Real", como também para solucionar "pequenos" percalços que tivemos no desenvolvimento deste trabalho.

Foi também particularmente interessante o uso de um microcontrolador, pois pudemos solidificar os conhecimentos obtidos durante o semestre na cadeira de "Sistemas Baseados em Microprocessadores", tanto a nível da programação do microcontrolador como no projecto do circuito de acondicionamento do mesmo, na alimentação do sensor envolvido neste trabalho, assim como a ligação com a placa cedida, e a ligação ao PC. Desta forma podemos dizer que está garantido assim a interface ControladorKit. Verificamos ainda que o Matlab embora sendo uma ferramenta poderosíssima não é o software apropriado para o controlo viaportas érie de um sistema, pois é muito lento neste aspecto. O Delphi seria uma das ferramentas apropriadas, pois é mais virado no sentido da comunicação com o PC.

Podemos ainda dizer que este trabalho foi bastante importante também para consolidar os conhecimentos adquiridos na Cadeira de "Sistemas Baseados em Lógica Difusa" e verificar as potencialidades deste controlo.

As principais dificuldades sentidas foram a nível de Hardware, pois o controlo fuzzy em si não foi muito complicado implementar.

Resta ainda salientar que para trabalhos desta envergadura convém ter uma certa calma quando escolhemos ir por um caminho e ao depararmo-nos com dificuldades, em vez de insistimos em resolvê-las mudamos de caminho. Isto muitas das vezes (como ficou provado connosco) não é a melhor opção, pois existe uma grande diferença entre saber o caminho e percorrê-lo. Este trabalho ajudou-nos a amadurecer em muitos aspectos, entre os quais estes últimos.

Por fim resta convidar a ler os outros relatórios para se obter uma visão mais ampla de como todo este trabalho foi elaborado convidamos mais uma vez a ler os relatórios de SBM (já pronto) e o relatório de SAM que dentro em breve será também entregue como anexo.



9. Anexos

ANEXO (código em Matlab)

```
close;clc;
g75=[.95.75.75];g8=[.8.8.8];i=0;op=1;

fig=figure('menubar','none','units','normalized','position',[0 0 1 .95],'name','Controlo da posição de um
Disco',...
    'numbertitle','off','color','black');
set(fig,'DoubleBuffer','ON');

axes('drawmode','fast','userdata','membrane','units','normalized','Position',[0.40.41 1],'box','on','visible','off');

%PUSH's-----
p1=uicontrol('Style','push','string','Regras Fuzzy','Units','normalized','Position',[0.62 0.29 0.08 0.03],
    'callback','[op]=1',...
    'backgroundcolor',[.75.75.75]);

p2=uicontrol('Style','push','string','SAIR','Units','normalized','Position',[0.62 0.18 0.08 0.06],
    'callback','[op]=5,fclose(s1),closeall',...
    'backgroundcolor',[.75.75.75]);

%TEXT's-----
t1=uicontrol('Style','text','string','Posição a alcançar (cm)','FontSize',12,'Units','normalized','Position',[0.62
0.70.250.03],...
    'backgroundcolor',[.75.75.75]);
t2=uicontrol('Style','text','string','Valor actual do Disparo (graus)','FontSize',12,'Units','normalized',
'Position',[0.620.460.250.03],...
    'backgroundcolor',[.75.75.75]);
t3=uicontrol('Style','text','string','Posição actual (cm)','FontSize',12,'Units','normalized','Position',[0.620.58
0.250.03],...
    'backgroundcolor',[0.75.75.75]);
t4=uicontrol('Style','text','string','Visualização das regras fuzzy','FontSize',12,'Units','normalized',
'Position',[0.620.340.250.03],...
    'backgroundcolor',[0.75.75.75]);

%EDIT's-----
e1=uicontrol('Style','edit','FontSize',12,'Units','normalized','Position',[0.620.650.080.03],...
    'backgroundcolor',[.15.61.95]);

e2=uicontrol('Style','edit','FontSize',12,'Units','normalized','Position',[0.620.410.080.03],...
    'backgroundcolor',[.15.61.95]);

e3=uicontrol('Style','edit','FontSize',12,'Units','normalized','Position',[0.620.530.080.03],...
    'backgroundcolor',[.15.61.95]);

%%- -----
%%- -----
holdon;
subplot('position',[0.1,0.1,0.3,0.88]);
B0=IMREAD('kit.bmp');
fig0=image(B0);
axisoff;
```

```

drawnow;

%- -----Abertura da Porta Série-----
s1=serial('COM3','Baudrate',38400)%define porta
fopen(s1)%abre porta

%- -----Inicialização das variáveis-----
erro1=0
pos=25
disparo3=180
%- -----Controlo-----
while op~=5

switch op;

case 1

hold on;
subplot('position',[0.51,0.8,0.37,0.17]);
B1=IMREAD('feup.bmp');
fig1=image(B1);
axis off;
drawnow;

if pos~=str2num(get(e1,'String'))
    pos=str2num(get(e1,'String'))
    if pos<1 | pos>55

        error('Posições Válidas de 1 a 55!!!','Erro');
        set(e1,'String','');
        op=1;
    end
else
end
pos
w=s1.BytesAvailable

if w~=0
op=3
end

case 2
ruleview(fuzzy_sets)
op=1

case 3

hold on;
subplot('position',[0.14,0.22,0.06,0.62]);%&x,ytamx,tamy
B2=IMREAD('regua.bmp');

fig2=image(B2);
axis off;
drawnow;

```



```
%calculoda saidado sensorem funçao da alturapretendida

rec=fscanf(s1,'%c',w)%le o valoractualdo ADC
k=s1.BytesAvailable

ifrec=='r'|rec=='>r'

    fprintf(s1,'%c','r')pede ao avro valoractualdo ADC
    pause(1);
    q=s1.BytesAvailable
    while(q==0)
        q=s1.BytesAvailable
    end
    ifq~=0

        ADC2=fscanf(s1,'%c',q)%le o valoractualdo ADC
        k=s1.BytesAvailable
        end
        ifk~=0
            karamelo=fscanf(s1,'%c',k)%limpa eventuallixoda portasérie
            end
        ADC=str2num(ADC2)
        Altura = -2.008377743419*((ADC/380)^6) + 25.856920825050*((ADC/380)^5) -
135.300139807327*((ADC/380)^4) + 376.958350602915*(ADC/380)^3 - 606.010587307718*((ADC/380)^2)
+ 564.866278844587*(ADC/380) - 210.060393336037
        ifAltura<0
            Altura=0
        end
        set(e3,'String',int16(Altura))

holdon;

subplot('position',[0.2,(Altura*0.01),0.2,0.6]);%x,ytamx,tamy
B3= IMREAD('disco.bmp');

fig3=image(B3);
axisoff;
drawnow

erro0=posAltura%calculodo erro

derro=erro1 erro0%calculoda derivada do erro

disparo1=evalfis([erro0derro],fuzzy_sets)+disparo3%consultada tabelafuzzy
disparo3=disparo1
disparo2=int16(disparo1)%convertep into valordo disparo

set(e2,'String',disparo2*0.16)
disparo=num2str(disparo2)%convertep stringo valordo disparo
if(disparo2<100)
    fprintf(s1,'%c','n')
```



```

        fprintf(s1,'%c',disparo(1))%enviudo valordo disparo
        fprintf(s1,'%c',disparo(2))
    else
        fprintf(s1,'%c','m')
        fprintf(s1,'%c',disparo(1))%enviudo valordo disparo
        fprintf(s1,'%c',disparo(2))
        fprintf(s1,'%c',disparo(3))
    end

    alt=num2str(int16(Altura))
    t=s1.BytesAvailable
    while(t==0)
        t=s1.BytesAvailable
    end
    var=fscanf(s1,'%c',t)
    if(var=='h')
        if(int16(Altura)>9)
            fprintf(s1,'%c',alt(1))
            fprintf(s1,'%c',alt(2))
        else
            fprintf(s1,'%c','0')
            fprintf(s1,'%c',alt(1))
        end
    else
        end
    erro1=erro0%atualizaçãodo erro

end%fecha ifrec='r'

ifk~=0
    karamelo=fscanf(s1,'%c',k)%limpa eventuallixoda portasérie
end

op=1

otherwise
end
end

```