

1 Descrição do sistema

O sistema será composto por uma planta, a ser simulada, e um controlador. A planta é composta por apenas um tanque, com uma válvula de entrada e uma válvula de saída e um sensor de nível. Todos os 'atuadores' e 'sensores' são normalizados.

2 Protocolo a ser implementado

O formato das mensagens seguirá o seguinte padrão: **<keyword>#<seq>#<value>!** ou **<keyword>!** e as respostas terão o seguinte formato **<return string>#OK!**, **<return string>#<value>!** ou **<return string>#<seq>!** sendo **<seq>** um número de sequência (pseudo-aleatório) qualquer. **<value>** representa um valor, inteiro, no intervalo 0 à 100. O conjunto de **<keywords>** é:

OpenValve sintaxe: **OpenValve#<seq>#<value>!** **<value>** representa o valor de pontos percentuais que a válvula será aberta.
Retorno: **Open#<seq>!** confirmando o comando **<seq>** recebido e executado.

CloseValve sintaxe: **CloseValve#<seq>#<value>!** **<value>** representa o valor de pontos percentuais que a válvula será fechada.
Retorno: **Close#<seq>!** confirmando o comando **<seq>** recebido e executado.

GetLevel sintaxe: **GetLevel!** retorna a nível atual do tanque.
Retorno: **Level#<value>!** nível atual.

CommTest sintaxe: **CommTest!** retorna um OK.
Retorno: **Comm#OK!**

SetMax sintaxe: **SetMax#<value>!** **value** representa o fluxo máximo de saída.
Retorno: **Max#<value>!** confirmando o valor enviado.

Start sintaxe: **Start!** (Re-)Inicia o simulador da Planta.
Retorno: **Start#OK!**

— Em caso contrário: O servidor não reconheceu o comando
Retorno: **Err!**

3 Trabalho a ser entregue

O trabalho a ser entregue é composto por dois módulos:

1. *Client* : Responsável pelo controle remoto do 'simulador da planta'. O qual deve ter pelo menos duas threads ativas:
 - (a) Uma Thread de controle, propriamente dita, a qual acessará o 'Nível Atual' do tanque e controlará a abertura da 'Válvula de entrada' via protocolo IP.
A periodicidade desta thread é a critério do grupo.
 - (b) Uma Thread de exibição gráfica, no qual devem ser protadas as variáveis de nível atual e a suposta abertura atual da válvula de entrada.
A periodicidade desta thread deve ser de 50ms.
2. *Server* : Responsável pela simulação da planta. O qual deve ter pelo menos três threads ativas:
 - (a) Uma Thread para simulação da planta. com periodicidade de 10ms.

- (b) Uma Thread de exibição gráfica, no qual devem ser protadas as variáveis de nível atual, abertura das válvulas de entrada e saída.
A periodicidade desta thread deve ser de 50ms.
 - (c) Uma Thread para o servidor IP. Esta thread será responsável por receber os comandos do 'client' e ajustar os parâmetros do simulador.
3. *Control Target* : Manter o nível do tanque em 80%, com um overshoot de no máximo 2%.

4 Modelo simplificado da planta

```

;
if OpenValve then
    | delta += value
end
if CloseValve then
    | delta -= value
end
if SetMax then
    | Max := value
end
if delta > 0 then
    if delta < 0.01*dT then
        | in.angle(T+dT) := in.angle(T)+delta;
        | delta := 0
    else
        | in.angle(T+dT) := in.angle(T)+0.01*dT;
        | delta -= 0.01*dT
    end
end
else
    if delta < 0 then
        if delta > -0.01*dT then
            | in.angle(T+dT) := in.angle(T)+delta;
            | delta := 0
        else
            | in.angle(T+dT) := in.angle(T)-0.01*dT;
            | delta += 0.01*dT
        end
    end
end
end
in.angle(0) := 50;
influx(T) := 1*sin(pi/2*in.angle(T)/100);
outflux(T) := (MAX/100)*(level(T)/1.25+0.2)*sin(pi/2*out.angle(T)/100);
level(0) := 0.4;
level(T+dT) := level(T)+0.00002*dT*(influx(T)-outflux(T));

```

Procedure plant

```

;
if (T <= 0) then
    | return 50;
end
if (T < 20000) then
    | return (50+T/400);
end
if (T < 30000) then
    | return 100;
end
if (T < 50000) then
    | return (100-(T-30000)/250);
end
if (T < 70000) then
    | return (20 + (T-50000)/1000);
end
if (T < 100000) then
    | return(40+20*cos((T-70000)*2*pi/10000));
end
return 100;

```

```

/* T em miliseconds */

```

Function out.angle(T)