



Controle I

Projeto – Controle do nível de água

LUCAS CARVALHO

PEDRO HENRIQUE GROSSI DA SILVA

CURITIBA

2023

Controle I – Controle de nível de água

INTRODUÇÃO

O objetivo deste projeto é construir um controlador PID a fim de controlar o nível de água de um recipiente utilizando uma minibomba de água, uma célula de carga e um microcontrolador Arduino UNO para a disciplina de Controle 1.

IDEIA DE PROJETO

Durante o semestre foram cogitadas várias ideias de projeto, como controladores de posição por temperatura, controladores de velocidade de motor, entre outros com graus de dificuldade que não tínhamos ideia.

Porém com o desenvolvimento do semestre fomos maturando a ideia de construirmos um controlador de posição. Desde o princípio já pensamos em controlar a posição de um recipiente com água, não teve um motivo específico, mas inicialmente a ideia era realizar o controle de nível de água esvaziando o recipiente usando uma válvula controlada. Quando fomos buscar os componentes para construir esse sistema nos deparamos com o fato de que a válvula precisaria de pressão mínima para funcionar e no nosso projeto isso não existia, a ideia era tudo funcionar por gravidade.

Com essa ideia frustrada, percebemos que precisaríamos de uma bomba de água e, enquanto pesquisávamos, encontramos um projeto que fazia um controle de nível de água, porém no modelo start-stop. Vendo aquele vídeo tivemos a ideia de adaptar aquele sistema para utilizar um controlador.

ADAPTAÇÃO DO SISTEMA

O sistema que vimos funcionava da seguinte maneira, quando o nível de água atingia um limite inferior (nível inferior detectado por um sensor de nível de líquidos parecido com o XKC-Y25-V) era ligada uma bomba de água que enchia um recipiente até um nível superior (detectado por um sensor similar ao XKC-Y25-V) que desligava a bomba.

Claramente por serem sensores detectores de nível, eles não serviriam para o nosso sistema de controle. Conversando com o professor da disciplina de Controle

ele, dentre várias sugestões, deu a ideia de realizar a leitura do nível de água pelo peso do recipiente. E nós decidimos utilizar essa forma de leitura no nosso sistema.

E para encher o nosso recipiente, uma bomba de água controlada pelo nosso microcontrolador via PWM e uma ponte h.

ESCOLHA DOS MATERIAIS UTILIZADOS

- Célula de carga

Quando fomos buscar uma maneira de lermos o peso do recipiente, a maneira mais simples que encontramos foi por meio de uma célula de carga. Inicialmente, a célula de carga escolhida foi a de 50Kg, pois foi a mais barata encontrada. Porém ao realizar os primeiros testes percebeu-se que na zona na qual buscávamos operar o sensor trazia leituras muito imprecisas, portanto, foi necessário trocá-lo.



Célula de carga 50kg

Sabendo agora a região de carga que gostaríamos de operar optou-se então por uma célula de carga com um *range* menor, de até 5kg, que apresentou resultados mais satisfatórios.



Célula de carga 5kg

A maneira com que este sensor realiza suas medições é por meio da contorção realizada pelo peso sobre o material. Portanto para a utilização correta deste sensor foi necessário a construção de um suporte, formando assim uma balança.



Balança

- Módulo HX711

Devido os sinais que saem da célula de carga serem muito pequenos, se fez necessário a aquisição de um amplificador de sinal. Foi optado pelo módulo HX711 pois além de realizar a amplificação ele converte o sinal para digital.



Módulo HX711

- Minibomba da água

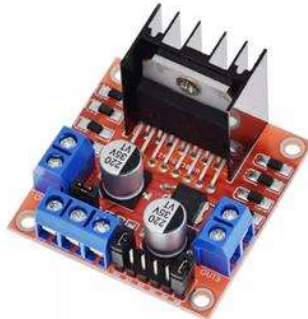
Como bomba de água foi escolhida uma minibomba de água 6V.



Minibomba da água

- Módulo Ponte H

Como ponte H para controlar a bomba de água foi escolhido o Módulo Ponte H L298n.



Ponte H

- Microcontrolador

O microcontrolador escolhido para este projeto foi o Arduino Uno, pois era o microcontrolador que já tínhamos.



Arduino UNO

- Fonte 12V

Para alimentar a Ponte H e a bomba da água foi utilizada uma fonte de 12V.



Fonte 12v

CALIBRAÇÃO DA CELULA DE CARGA

Para a correta utilização da célula de carga é necessário efetuar sua calibração, para isso foi utilizada a biblioteca HX711.h para o Arduino, nela existem diversas funções voltadas para o uso de uma célula de carga no Arduino.

Para calibrar a balança utilizamos uma outra balança, onde aferíamos o peso de uma amostra e colocávamos ele sobre a nossa balança. E conforme o peso era impresso na serial, íamos ajustando o fator de correção até que chegasse no peso correto do objeto.

```
void loop ()
{
    escala.set_scale(fator_calibracao); //ajusta a escala para o fator de calibracao

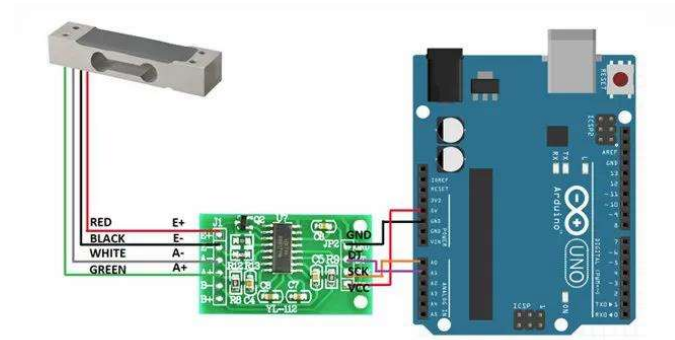
    //verifica se o modulo esta pronto para realizar leituras
    if (escala.is_ready())
    {
        //mensagens de leitura no monitor serial
        Serial.print("Leitura: ");
        Serial.print(escala.get_units(), 1); //retorna a leitura da variavel escala com a unidade quilogramas
        Serial.print(" kg");
        Serial.print(" \t Fator de Calibracao = ");
        Serial.print(fator_calibracao);
        Serial.println();

        //alteracao do fator de calibracao
        if(Serial.available())
        {
            comando = Serial.read();
            switch (comando)
            {
                case 'x':
                    fator_calibracao = fator_calibracao - 100;
                    break;
                case 'c':
                    fator_calibracao = fator_calibracao + 100;
                    break;
                case 'v':
                    fator_calibracao = fator_calibracao - 10;
                    break;
                case 'b':
                    fator_calibracao = fator_calibracao + 10;
                    break;
                case 'n':
                    fator_calibracao = fator_calibracao - 1;
                    break;
                case 'm':
                    fator_calibracao = fator_calibracao + 1;
                    break;
            }
        }
        else
        {
            Serial.print("HX-711 ocupado");
        }
        delay(TEMPO_ESPERA);
    }
}
```

RESPOSTA EM MALHA ABERTA

Com a célula de carga devidamente calibrada, tentamos levantar a curva em malha aberta do sistema utilizando um pote de 2L

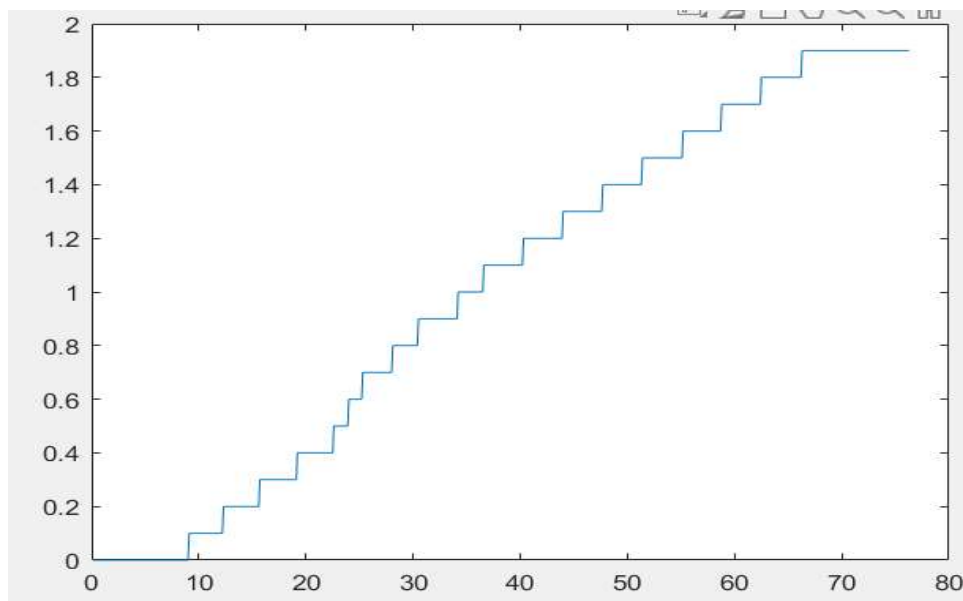
A balança foi ligada seguindo o esquemático abaixo, e a bomba da água foi ligada direto na saída 5V do Arduino.



Esquemático - Balança

O primeiro problema encontrado foi que o Arduino não estava sendo capaz de alimentara bomba de água, a solução inicial foi utilizar uma bateria de 9V em conjunto com o regulador de tensão LM7805, desse modo conseguindo 5V. Porém a bateria começou a falhar após alguns testes. Sendo assim optou-se por usar uma fonte de celular de 5V.

Com o sensor de peso trabalhando de 0kg até 2kg, algumas inconsistências ocorreram. Conforme demostra o gráfico abaixo.



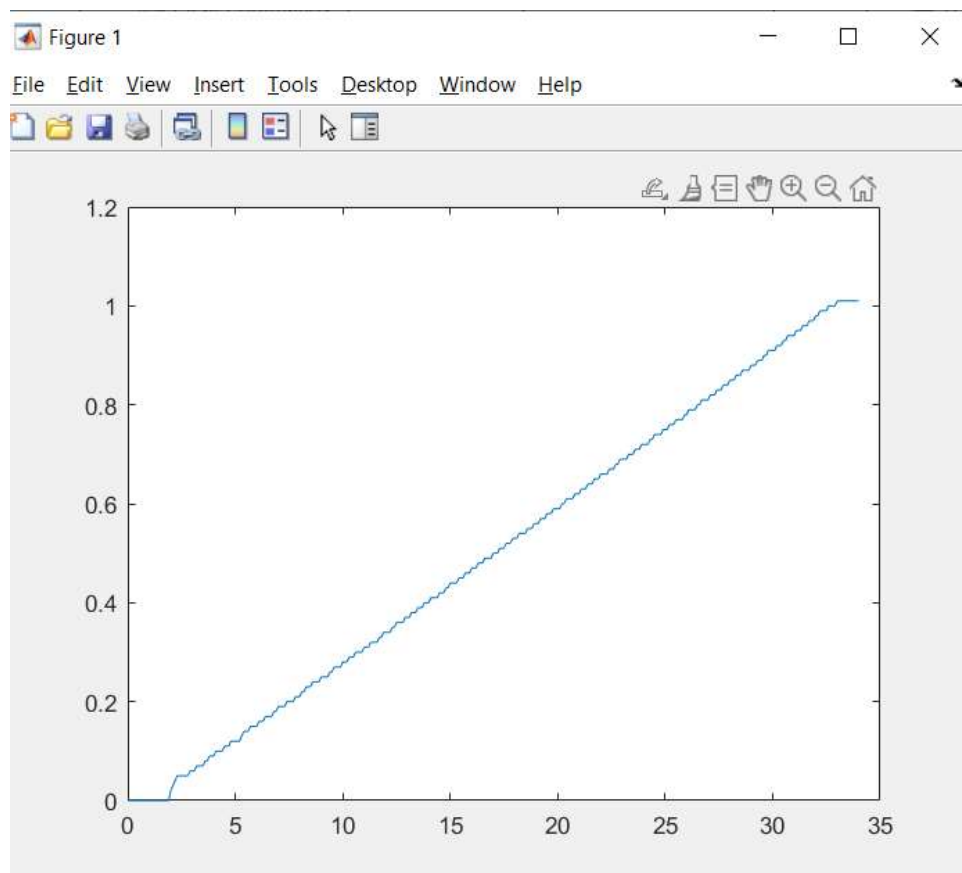
Sensor – 0 até 2kg

Logo, optou-se por trabalhar com uma região de maior peso, para isso o pote de 2L foi substituído por um galão de 5L. Dessa forma conseguiu-se utilizar todo o *range* do sensor e não somente 40%.



Montagem do Projeto

Nesse arranjo, a resposta do *step* de 5V foi o seguinte:



Resposta ao Step - 5V

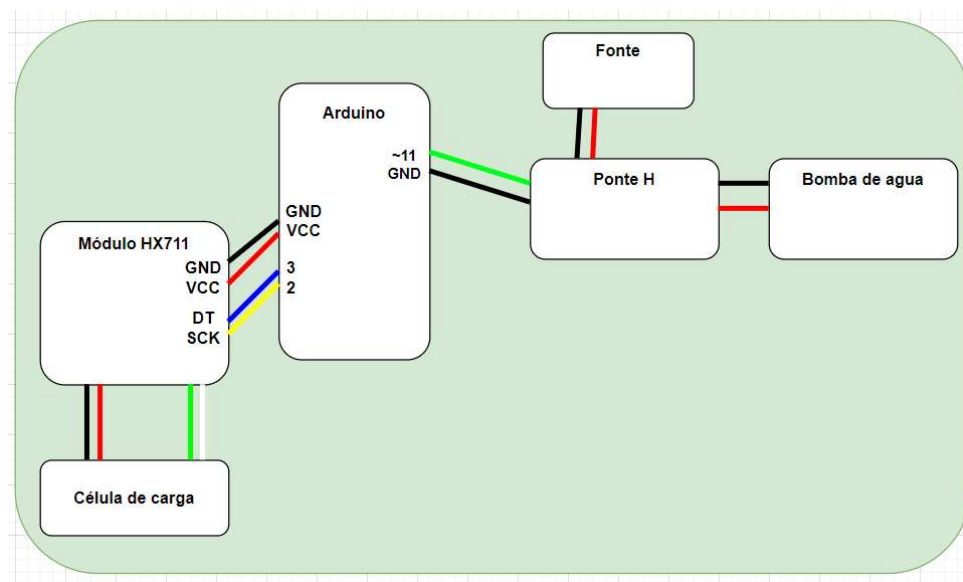
Através desse gráfico, pode-se chegar as seguintes conclusões:

- O sensor está devidamente calibrado e respondeu linearmente ao *step*, conforme o esperado;
- A bomba foi capaz de fornecer uma corrente linear de água, cerca de 33 gramas por segundo.

REPOSTA EM MALHA FECHADA

Para levantar a resposta em malha fechada do sistema, a balança foi ligada da mesma forma, porém foi necessário ligar a bomba da água em conjunto com a Ponte H, pois dessa forma é possível controlar a tensão na bomba.

O esquemático ficou da seguinte forma:



Esquemático - Malha Fechada

Para o código então foi feita então a seguinte implementação para fechar a malha:

- Por se tratar de um sistema de nível, foi escolhida uma posição próxima ao centro do galão de água para ser o nível superior. Quando fomos medir quanto de água se encontrava naquele ponto chegamos a um valor próximo de 3.5Kg, portanto esse valor foi escolhido como nosso *setpoint*.
- Como nós queríamos fazer um controle de nível, mas também não gostaríamos que demorasse muito para chegar ao ponto, resolvemos escolher

uma posição próxima ao nosso *setpoint*, foi escolhida uma posição próxima a região de 2.5Kg do galão.

- Como o recipiente já inicia sobre a balança é necessário fazer um ajuste no peso para o peso real. (“*weight* +=2.5”)
- O erro é calculado pela diferença entre o *setpoint* e o peso efetivo
- E para que a bomba seja controlada esse erro é enviado para o PWM

```
8
9  HX711 scale;
10
11 float reading_average = 0;
12 float weight = 0;
13 float setpoint= 3.50;
14 float error = 0;
15 int controlPWM = 0;
16
```

```
26 void loop()
27 {
28     scale.set_scale(calibration_factor);
29     if (scale.is_ready())
30     {
31         weight = scale.get_units(); //receives weight from scale
32         weight += 2.50; //adjust the weight of the container
33         error = setpoint - weight; //calculate the error
34         Serial.print(error); //print of error
35         Serial.print(" ");
36         Serial.print(weight); //print of weight
37         Serial.println();
38         controlPWM = error*99+42; //calculate the PWM value based on our scale
39         analogWrite(pumpPin, controlPWM); //send PWM value
40     }
41     delay(wait);
42 }
```

Inicialmente, quando enviávamos o erro para o PWM, era da seguinte maneira:

*ControlPWM = error*255;*

Porém quando fomos testar desta forma percebemos um problema mecânico do nosso sistema, como a bomba precisa bombear a água para cima a fim de entrar no galão, quando o controlador está regulando os instantes finais, a tensão sobre a bomba é muito baixa e ela não possui tensão suficiente para fazer a água subir.

Tendo percebido isso, nos testamos para descobrir qual era o valor de PWM em que a bomba já não conseguia mais fazer com que a água subisse e chegamos

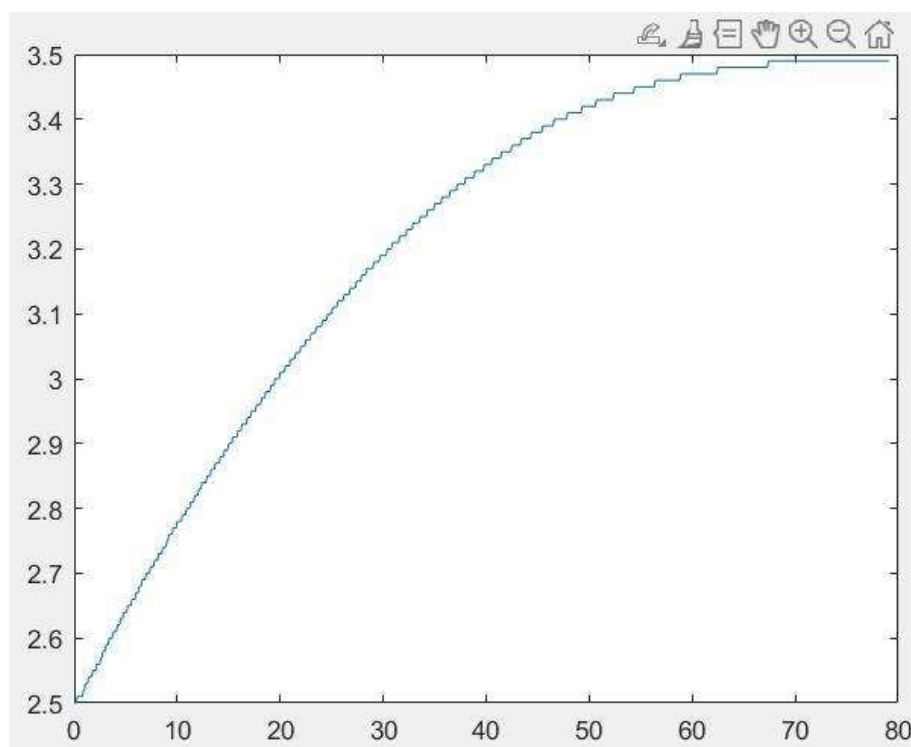
à média de 42. Por esse motivo, como o valor de PWM também possui um limite (0 a 255) foi necessário estabelecer uma forma de conversão.

Com isso foi utilizado a seguinte equação:

$$\frac{(erro - 0)}{1 - 0} = \frac{(PWM - 42)}{255 - 42}$$

E com isso chegamos à conversão de erro em PWM: $PWM = error \cdot 99 + 42$

Com isso então a resposta do sistema, com uma realimentação de ganho unitário foi a seguinte:



Resposta do Sistema – Ganho Unitário

RESPOSTA EM MALHA FECHADA COM GANHO

Com isso feito, partimos então para encontrarmos a função de transferência do nosso sistema, para isso seria necessário realizar a implementação de ganho ao nosso sistema de malha fechada. Resolvemos seguir o tutorial disponível no Drive da disciplina, neste tutorial era necessário que o sistema apresentasse *overshoot* para que se consiga formar a função de transferência.

Desse modo, utilizando o mesmo código da malha fechada foi adicionado a parte de ganho ao nosso sistema.

```

16
17 //Proportional
18 double kP = 2.15; //gain necessary to make the overshoot
19 double P = 0; //variable to proportional value
20

```

```

41 Serial.println();
42 P = error*kP; //calculate the gain
43 controlPWM = P*99+42; //calculate the PWM value based on our scale
44 analogWrite(pumpPin, controlPWM); //send PWM value

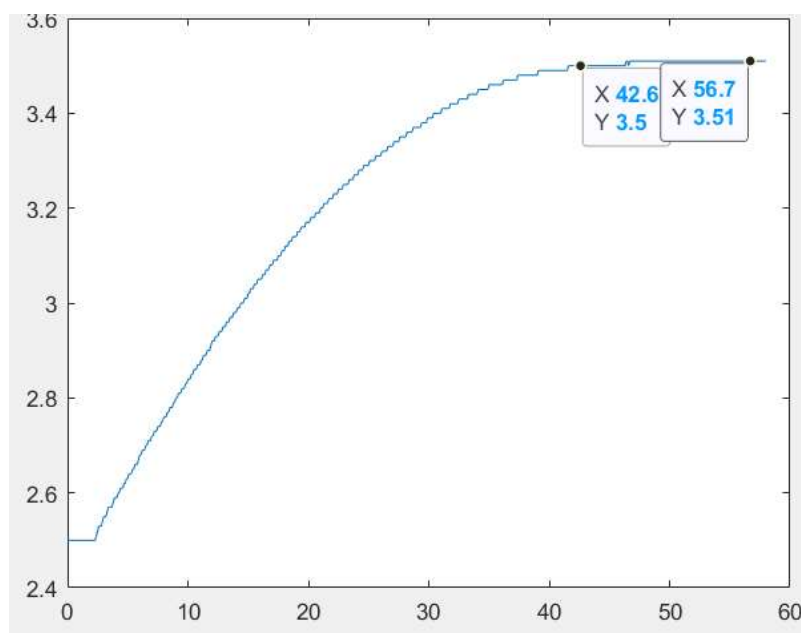
```

Entretanto nesta etapa nos encontramos algumas limitações do hardware do nosso sistema e elas foram:

- A nossa fonte de alimentação, por mais que de 12V, a maior tensão que chegava aos terminais da ponte H era de 9V.
- Por mais que se chegasse ao valor de 9V nos terminais a nossa bomba de água funciona na margem de 6V portanto nos já estaríamos forçando-a.

Porém mesmo sabendo das limitações, resolvemos forçar a bomba e fomos subindo o ganho do nosso sistema até que fosse apresentado um *overshoot*. Entretanto, conforme fomos subindo o nosso sistema não ia apresentando nenhum *overshoot*, até chegarmos ao ponto de calcularmos qual era o maior valor de ganho que o nosso sistema conseguia suportar, visto as limitações de valores do PWM.

Com isso chegamos ao ganho de 2.1515. Colocando este ganho no nosso sistema, foi apresentado um *overshoot* mínimo de 10g como mostra o gráfico a seguir.



Overshoot

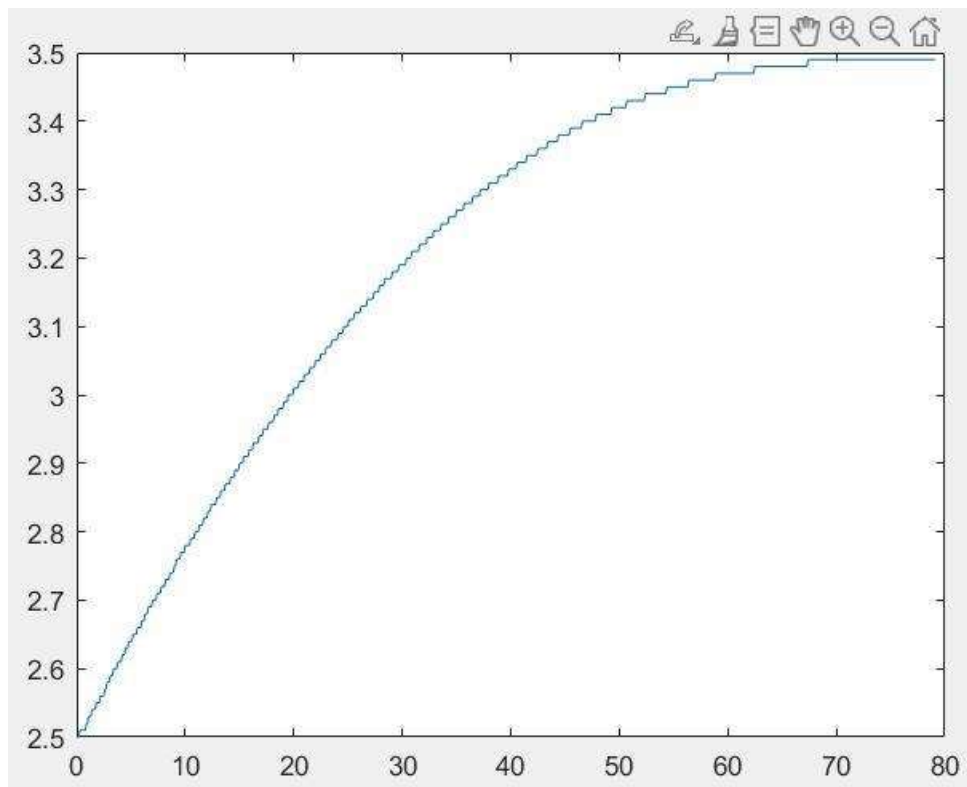
Visto esta informação, nós percebemos um problema de projeto. Como o nosso sistema só possui uma bomba de água, para encher o recipiente, nós não tínhamos como corrigir o *overshoot* caso fosse causado pelo nosso controlador. Como faltavam poucos dias para a apresentação final, nós resolvemos continuar assim o projeto e, tendo isso em mente, resolvemos então fazer um controlador que apresente o menor *overshoot* possível.

E quanto a função de transferência, como foi visto, o nosso sistema quase não apresenta *overshoot*, em algumas situações ele não apresenta mesmo (dependendo de como o galão foi posicionado). Dessa maneira não conseguimos levantar a função de transferência para o nosso sistema, porém, como será visto posteriormente, foi feito um controlador satisfatório sem o seu uso.

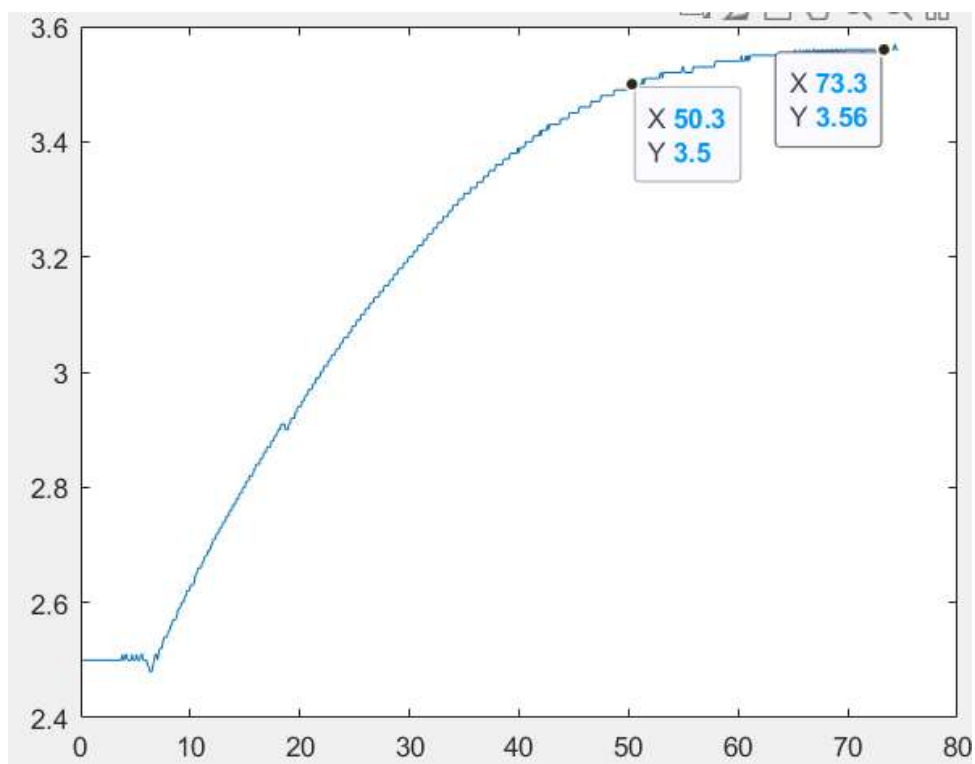
CONTROLADOR

Para atingir o nosso objetivo então, seria necessário um controlador que não apresentasse *overshoot*. Tendo em visto que no ganho máximo possível o nosso sistema quase não apresenta *overshoot*, tentamos reduzir um pouco o ganho e verificar se ele ainda apresentaria esse *overshoot*.

O ganho escolhido foi de 2.1, ao testá-lo no nosso sistema foi visto então que ele apresenta muito pouco *overshoot* servindo para o nosso propósito, visto que é natural que um controlador P apresente um *overshoot*, mesmo que pequeno. Foi constatado também que o tempo de estabilização do controlador diminuiu cerca de 20 segundos, saindo de aproximadamente 70 segundos para próximo de 50 segundos. Como mostram os gráficos a seguir:



Malha fechada sem o controlador



Malha fechada com o controlador

Dessa forma, visto as limitações de hardware encontradas e a necessidade de o sistema não possuir/possuir pouco *overshoot*, acredito que o resultado do controlador foi satisfatório.

CONTROLADOR

Acreditamos que este projeto de controle foi muito útil para a melhor compreensão dos conteúdos aprendidos durante o semestre. Entendemos os problemas/limitações do sistema que criamos, porém acreditamos que isto se deve à falta de experiência com projetos desta maneira. Acreditamos também que a experiência obtida durante esse desenvolvimento será de grande ajuda para que continuemos a projetar sistemas cada vez melhores.

Quanto ao sistema de controle em si, obtivemos resultados satisfatórios naquilo que nos propomos a fazer, por mais que bem restritos o sistema de comporta da maneira com que deveria.