

5 Montagem e teste do modelo real

Esse capítulo irá apresentar os processos de montagem do sistema (Figura 5.1), tanto as montagens que deram certo quanto as que não funcionaram a contento, como registro para futuras análises. O local escolhido, foi o Laboratório de Dinâmica e Vibrações da PUC-Rio. Para um melhor entendimento do que foi realizado, é necessário explicar alguns conceitos que serão muito utilizados neste capítulo.

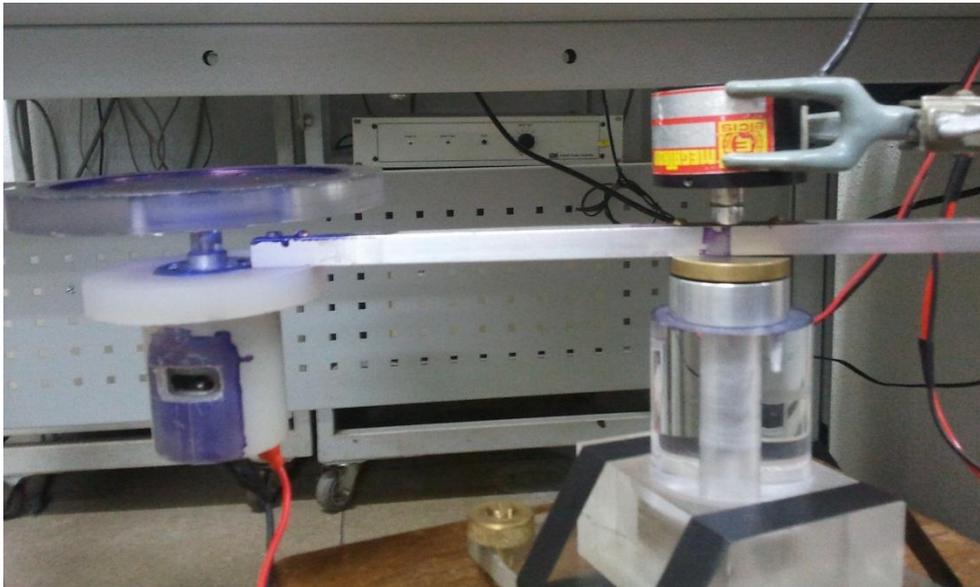


Figura 5.1 – Modelo construído no laboratório

5.1. Conceitos

5.1.1. Encoder

O encoder (Figura 5.2) é usado para medir ângulos e velocidades de rotação. Geralmente possui 4 fios: VCC, Ground e 2 fases. Essas fases tem um "delay" de um quarto de ciclo de clock (como mostrado na Figura 5.3), para obter a leitura

em ambos os sentidos. Quando a fase A está adiantada em relação à fase B, o encoder está lendo o giro em um sentido, quando o caso oposto, da fase B está adiantada em relação à fase A, tem-se o giro no outro sentido.



Figura 5.2 – Detalhe dos pinos do encoder

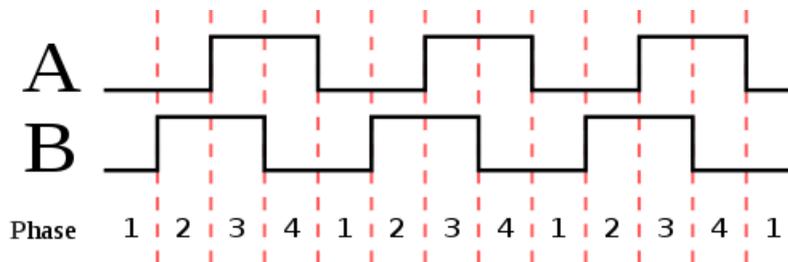


Figura 5.3 – Exemplo de leitura do encoder

5.1.2. Micro controlador Arduino

Assim como um PIC, esse "hardware" livre (Figura 5.4) dispõe de um chip micro controlador da Atmel, com entradas e saídas embutidas e uma linguagem chamada Wiring (similar ao C/C++), cujo objetivo é promover criações a baixo custo e é uma ferramenta muito simples de usar. Além de possuir várias entradas e saídas digitais e analógicas, a placa conta com um software aberto para programação da placa.

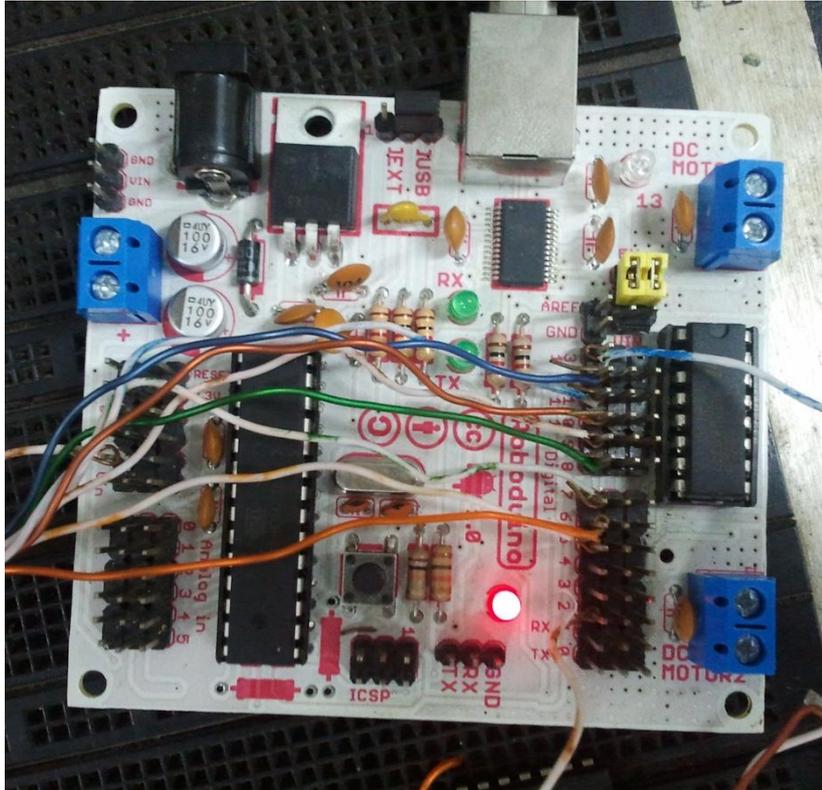


Figura 5.4 – Placa do arduino utilizado, modelo Roboduino

Este micro controlador se conecta ao PC por meio da interface serial (RS232), e pode ser usado em conjunto com o computador ou só (“stand alone”), dependendo da finalidade da placa.

5.1.3. DAQ

Placa de aquisição de dados da National Instruments (também chamada de NI-DAQ), é responsável por toda comunicação do sistema com o LabView. Internamente possui entradas e saídas analógicas, contadores, leitor de encoders e outros artifícios que facilitam a manipulação do sistema real com o digital.

Para o projeto foram utilizadas 2 tipos de placas: inicialmente a USB-6008 (Figura 5.5), e em seguida a USB-6229 (Figura 5.6).



Figura 5.5 – Placa NI-DAQ, modelo USB-6008

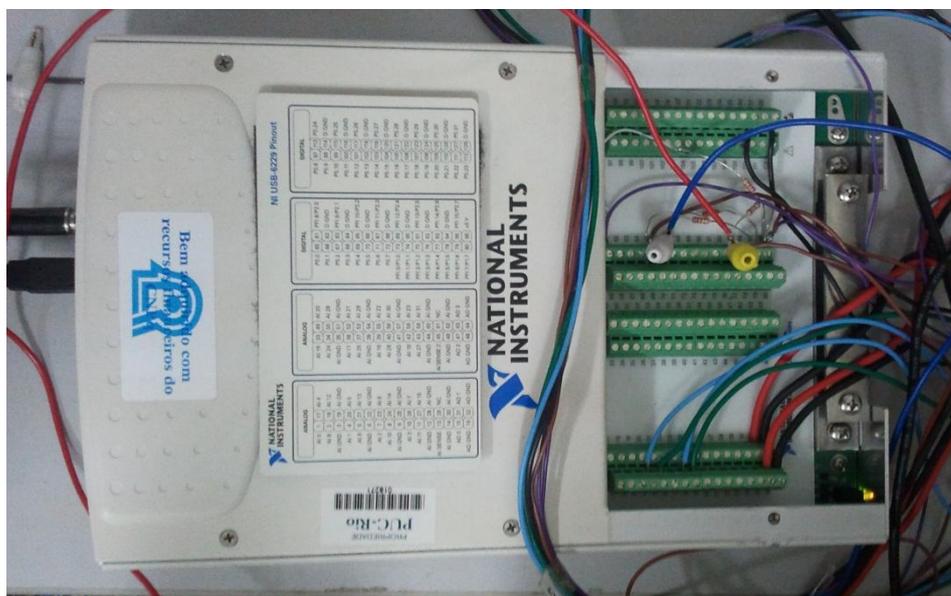


Figura 5.6 – Placa NI-DAQ, modelo USB-6229

Esta última, com bem mais recursos, já possui leitor de encoder rotacional, dispensando o uso do arduino e diminuindo assim o número de itens utilizados na confecção do projeto.

5.1.4. Ponte H

É um circuito eletrônico que permite o controle de um motor DC, que consiste de 4 chaves que são acionadas alternadamente para girar o motor em uma sentido ou no outro. Para acionar essas 4 chaves, foram utilizados transistores do tipo TIP41 e TIP42, com diodos do tipo 1N4004 para segurança do sistema, conforme mostra a Figuras 5.7 e 5.8.

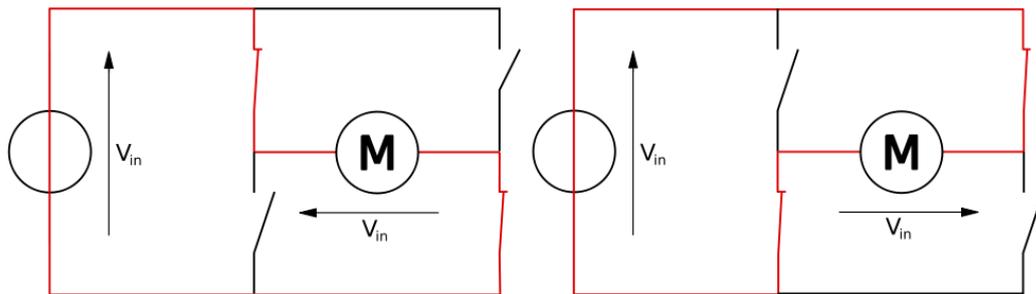


Figura 5.7 – Esquema de funcionamento da ponte H

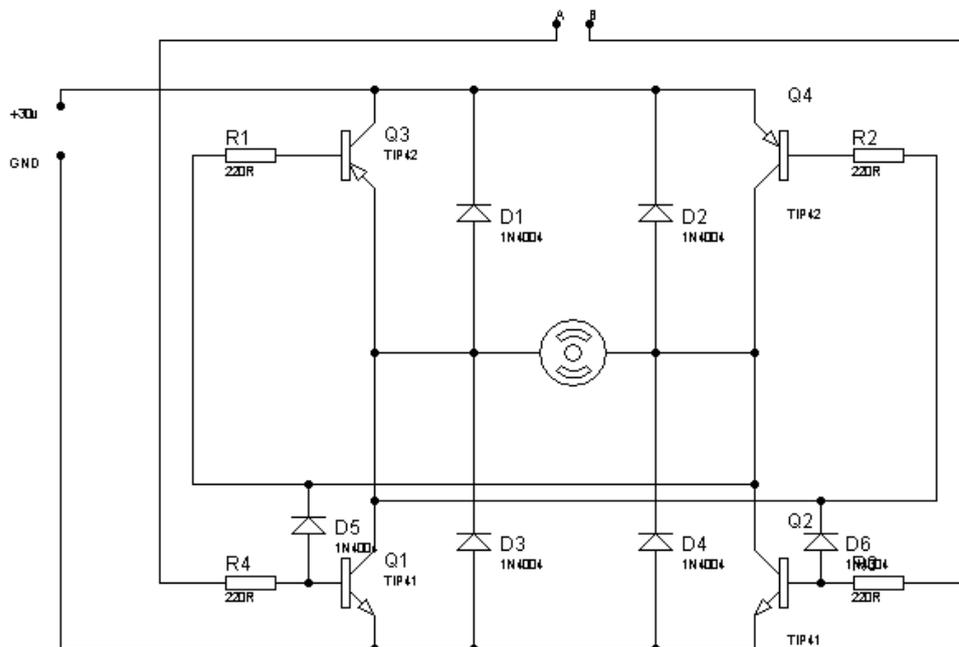


Figura 5.8 – Diagrama do circuito da ponte H

Para montar esse circuito (Figura 5.9), utilizou-se um procedimento artesanal de montagem de placas de circuito impresso, utilizando:

- Placa de cobre
- Folha de revista
- Ferro de passar roupa
- Solução de percloroeto de ferro
- Impressora a jato de tinta
- Software de criação de trilhas para placas de circuito (foi utilizado o Eagle Layout Editor, da CadSoft)

O processo de confecção é bem simples, praticamente imprimir o desenho espelhado do circuito feito no Eagle em uma folha de revista e em seguida, fixar a folha na placa de cobre para passar o ferro numa temperatura superior a 300C sobre o papel e com isso a tinta vai se fixar no cobre formando um espelho do espelho (ou seja, o original). Assim, mergulha-se a placa na solução de percloroeto de ferro, até que todo o cobre descoberto esteja corroído. E, por fim, após secar a placa e cuidando para não sobrar nenhum ponto brilhante de cobre, passa-se uma esponja de aço para tirar a tinta, formando assim as trilhas de cobre que são vistas em placas desse tipo.

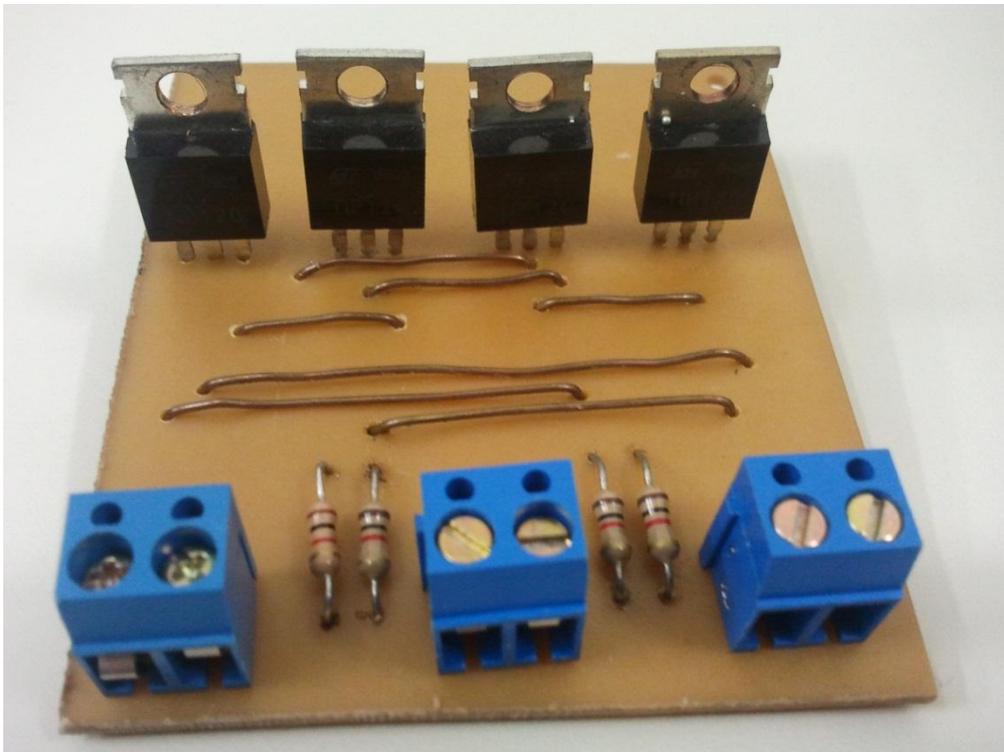


Figura 5.9 – Placa da ponte H confeccionada pelo aluno

5.2. Montagem

Essa seção irá detalhar os dois esquemas de montagem que foram utilizados para o experimento. O primeiro funcionou, mas não perfeitamente, e o segundo funcionou melhor, e deste foram tirados os resultados.

5.2.1. Esquema 1

Primeiramente, foi montado o esquema 1 da Figura 5.10 utilizando o micro controlador Arduino em conjunto com uma placa de aquisição de dados.

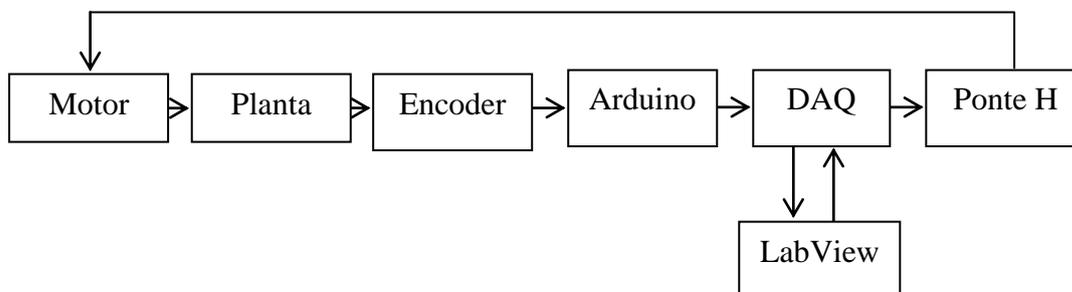


Figura 5.10 – Diagrama de blocos da primeira montagem

Para esta montagem inicial, foi utilizado um encoder de poucos pontos por revolução (200p/r) meramente para fins de teste. O DAQ utilizado foi muito simples, sequer possuía uma leitura de encoder. Para isso, o Arduino foi utilizado para realizar a leitura do encoder, e seus dados passavam por 8 bits para o DAQ, este fazendo a comunicação com LabView, com o ajuste do controle e em seguida passaria a resposta para o DAQ aplicar uma tensão na ponte H.

Para o projeto, o Arduino foi utilizado para ler o encoder, processar a leitura, e passar valor do encoder para 8 portas digitais conforme a figura abaixo. Como o encoder tinha 200 pontos por volta, 256 estados (com 8 bits tem-se 2^8 combinações diferentes binárias) eram mais que o suficiente para fazer a medição. O código a seguir foi escrito para essa finalidade.

```
/* Marcelo da Cruz Pereira - Mat: 0921505 */
```

```
/* Projeto de Mestrado - Leitura de encoder de quadratura */
```

```

/* Placa Roboduino com Arduino 0021 */
#define ENC_A 0
#define ENC_B 1
/* Pinos das fases A e B, respectivamente 0 e 1 */

int pinArray [] = { 6,7,8,9,10,11,12,13 };
int Size = 8;
/* Pinos da saida 8bits, pino 6 a pino 13 */

void setup()
{
    /* Inicializa os pinos do encoder */
    int i;
    for (i=0; i< Size; i++)
    {
        pinMode(pinArray[i], OUTPUT);
    }
    pinMode(ENC_A, INPUT);
    digitalWrite(ENC_A, HIGH);
    pinMode(ENC_B, INPUT);
    digitalWrite(ENC_B, HIGH);
    Serial.begin (115200);
    Serial.println("Start");
}

void loop()
{
    int num = 0;
    static uint8_t counter = 0;
    int8_t tmpdata;
    tmpdata = read_encoder();
    if( tmpdata ) {
        Serial.println(counter, DEC);
        counter += tmpdata;
    }
    num = counter;
    /* Escreve em 8 bits o valor lido no encoder */
    if((num & 0x01) == 1) {
        digitalWrite(pinArray[0], HIGH);
    } else {
        digitalWrite(pinArray[0], LOW);
    }
}

```

```

}
if((num & 0x02) == 0x02) {
    digitalWrite(pinArray[1], HIGH);
} else {
    digitalWrite(pinArray[1], LOW);
}
if((num & 0x04) == 0x04) {
    digitalWrite(pinArray[2], HIGH);
} else {
    digitalWrite(pinArray[2], LOW);
}
if((num & 0x08) == 0x08) {
    digitalWrite(pinArray[3], HIGH);
} else {
    digitalWrite(pinArray[3], LOW);
}
if((num & 0x10) == 0x10) {
    digitalWrite(pinArray[4], HIGH);
} else {
    digitalWrite(pinArray[4], LOW);
}
if((num & 0x20) == 0x20) {
    digitalWrite(pinArray[5], HIGH);
} else {
    digitalWrite(pinArray[5], LOW);
}
if((num & 0x40) == 0x40) {
    digitalWrite(pinArray[6], HIGH);
} else {
    digitalWrite(pinArray[6], LOW);
}
if((num & 0x80) == 0x80) {
    digitalWrite(pinArray[7], HIGH);
} else {
    digitalWrite(pinArray[7], LOW);
}
}

/* Retorna uma mudança no encoder (-1,0,1) */
int8_t read_encoder()
{

```

```

int8_t enc_states[] = {0,-1,1,0,1,0,0,-1,-1,0,0,1,0,1,-
1,0};

static uint8_t old_AB = 0;
/**/
old_AB <=<= 2;
old_AB |= ( ENC_PORT & 0x03 );
return ( enc_states[( old_AB & 0x0f )]);
}

```

Nesse primeiro modelos do projeto foi utilizada uma ponte H devido a falta de uma maneira de obter uma tensão negativa na placa NI-DAQ 6008.

Entretanto, esse modelo não teve muito sucesso nos testes por dois motivos:

- A leitura do encoder era muito imprecisa, para 200 pontos por revolução tem-se um erro de 1,8 graus.

- A corrente drenada pelo motor não era grande o suficiente para executar um torque na roda de reação capaz de mover a haste rapidamente.

Por esses motivos esse esquema foi deixado de lado, mas algumas peças puderam ser aproveitadas para outras montagens.

5.2.2. Esquema 2

Nessa montagem da Figura 5.11, a placa NI-DAQ USB-6008 foi substituída pela USB-6229, evitando o uso de uma ponte H externa. O micro controlador Arduino foi eliminado da montagem pois o novo NI-DAQ já possuía leitor de encoder de quadratura, facilitando assim a montagem.

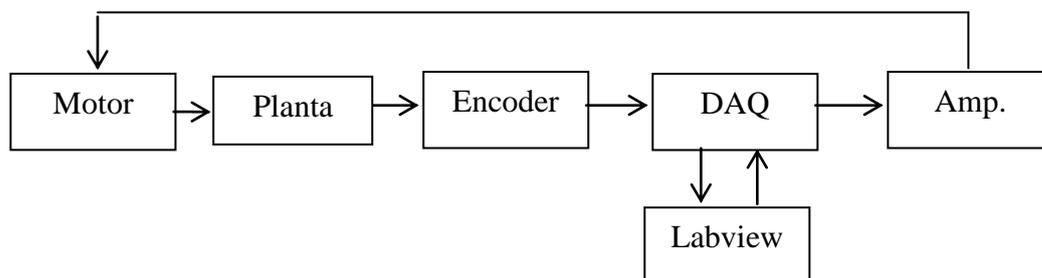


Figura 5.11 – Diagrama de blocos da segunda montagem

O encoder (Figura 5.12) também foi mudado por outro com precisão dez vezes superior, 2000 pontos por revolução. Com isso, ao invés de um erro de 1.8

graus como no esquema anterior, ter-se-ia um erro de apenas 0.18 graus na angulação da haste, permitindo assim um controle mais apurado do sistema.



Figura 5.12 – Encoder de 2000p/r

Há também um amplificador de potência (Figura 5.13) para poder gerar a corrente próxima daquela que foi utilizada no modelo experimental (4,6 Amperes). Essa placa já existia no Laboratório, por isso não será detalhada nesse trabalho.



Figura 5.13 – Amplificador de potência

Nesse esquema também foi testado um motor DC com redução (Figura 5.14) e um disco mais pesado (de ferro) (Figura 5.15).



Figura 5.14 – Motor de redução com disco de ferro



Figura 5.15 – Disco de ferro

Infelizmente para os testes com o disco mais leve e o motor DC convencional, e o disco mais pesado e o motor DC convencional, o resultado não

foi o esperado. O mesmo caso para o motor com redução. Nenhum disco acoplado neste motor teve um resultado bom, pois o pendulo não se moveu rápido o suficiente.

Com isso, dos três discos e dois motores testados, o disco de acrílico (peso intermediário dos três) e o motor sem redução foram os que casaram melhor em termos de movimentação da haste (Figura 5.16).

Uma simulação com o modelo matemático que poderia ser feita para medir a influência da inércia nos diversos testes acabou não sendo feita pelo alto número de testes já realizados, e em também que não era esta a proposta principal do trabalho.

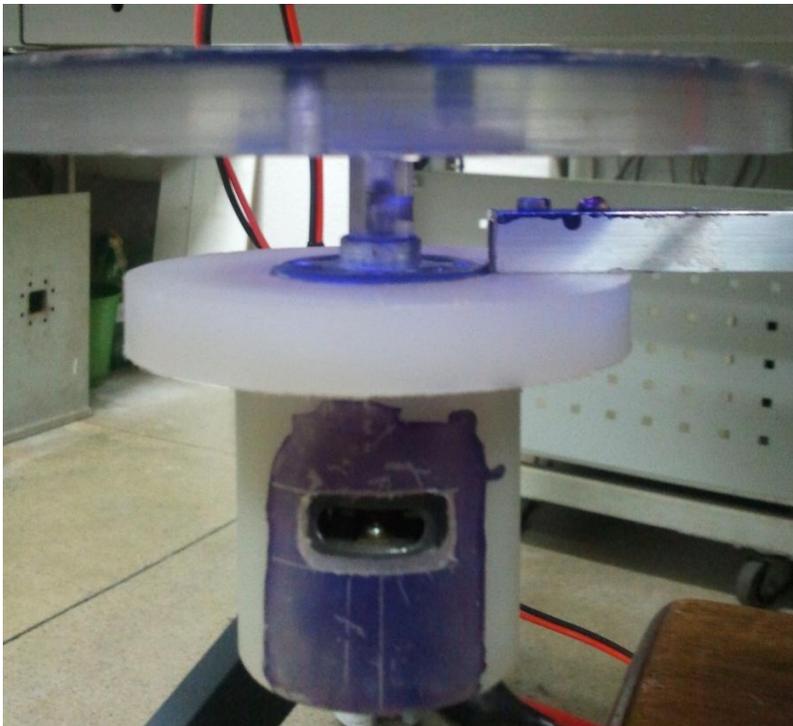


Figura 5.16 – Esquema do motor usado nos testes

Esse disco também foi alterado de forma a aumentar seu momento polar de inércia para uma mesma massa. Desta forma aumenta-se a massa na periferia do disco, como mostra a Figura 5.17. Foram feitos 4 furos de 20mm de diâmetro, e neles foram montados 4 pesos (porcas de metal) para aumentar a inércia um pouco, mas não a ponto de impedir o bom funcionamento do sistema.

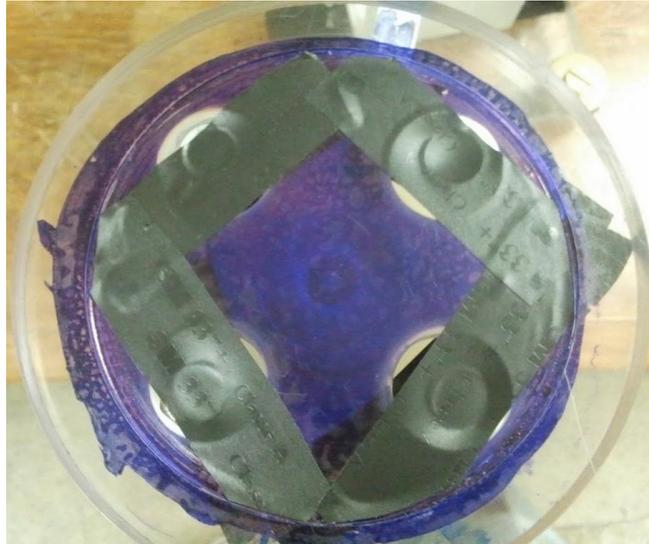


Figura 5.17 – Detalhe do disco

Também foi implementada uma base regulável (Figura 5.18) para minimizar a possibilidade de uma componente residual da gravidade que resulte em uma posição de equilíbrio para o pêndulo que deveria ser horizontal.



Figura 5.18 – Detalhe da base ajustável do sistema

5.3 Testes experimentais

Depois de escolhido qual esquema iria ser testado, estabeleceu-se a sequência de realização dos testes. Por não possuir uma maneira de controlar a corrente eficientemente, foram realizados 5 tipos de testes, todos com controle por tensão:

- Controle P
- Controle PI
- Controle PID
- Controle Fuzzy medindo apenas o ângulo da haste (SISO)
- Controle Fuzzy medindo ângulo e velocidade da haste (DISO)

Todos os testes foram feitos de duas maneiras: aplicando uma variação no Set Point (ângulo desejado) e observando o comportamento do ângulo em todos os casos, e aplicando uma força de impacto na haste para tirá-la do ponto de equilíbrio. Assim tem-se 10 testes práticos.

As janelas do LabView criadas para estes testes são mostradas nas Figuras 5.19 a 5.21:

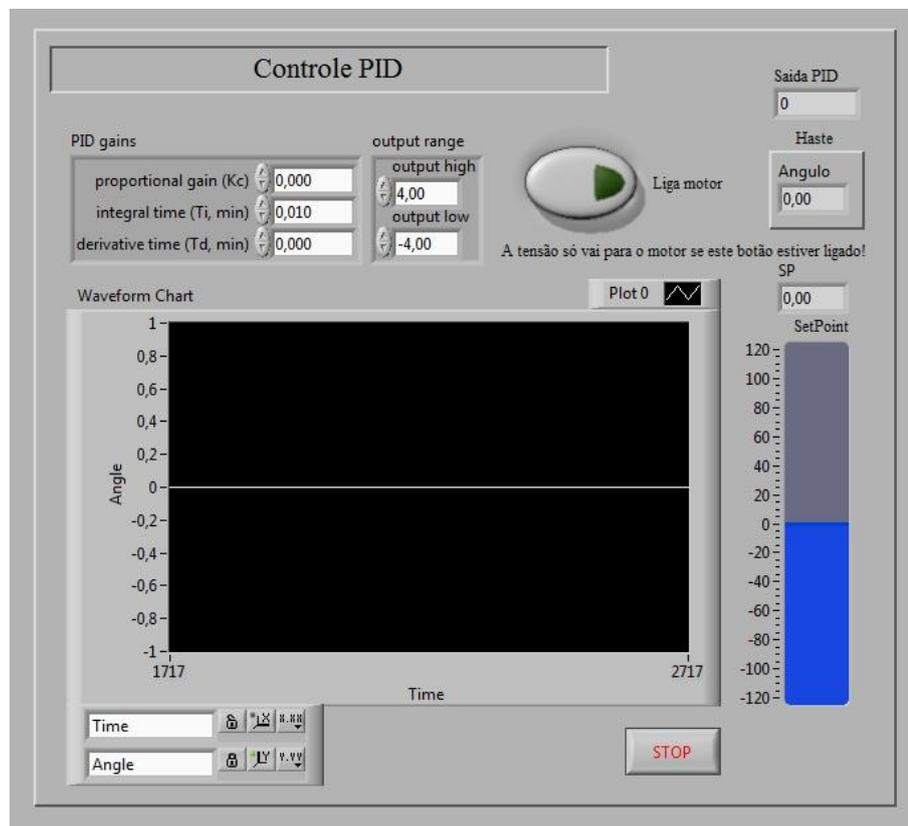


Figura 5.19 – Janela de testes no LabView para controle PID

O campo PID Gains permite o ajuste das constantes do controle PID, mas é interessante notar que ao invés dos clássicos k_p , k_i e k_d , esse bloco do Labview trabalha com k_c (o mesmo k_{crit} do método de Ziegler Nichols), T_i e T_d como

constantes de tempo (em minutos, não em segundos). A conversão dos valores se faz de acordo com as equações abaixo:

$$k_p = k_c \quad (5.1)$$

$$k_i = \frac{k_p}{T_i} \quad (5.2)$$

$$k_d = k_p * T_d \quad (5.3)$$

Considerando que no LabView, os valores T_i e T_d são mostrados em minutos, as equações 2 e 3 ficam assim:

$$k_i = \frac{k_p}{60 * T_i} \quad (5.4)$$

$$k_d = 60 * k_p * T_d \quad (5.5)$$

O que explica os valores bem pequenos usados no teste prático.

Output Range é um valor de máxima tensão que irá entrar no amplificador de potência que naturalmente possui um ganho 3, fazendo assim com que a máxima tensão seja 12V e a mínima seja -12V.

O gráfico da Figura 5.19 mostra a comparação do valor do Set Point (ângulo desejado) e o angulo lido pelo encoder (ângulo medido); seus respectivos valores numéricos estão nos campos SP e Angulo, e o ajuste do Set Point em formato de barra. Todos os testes foram feitos numa janela de 1000 aquisições de dados, que é cerca de 10 seg.

Em Saída PID, tem-se o valor da tensão da saída do bloco PID, que irá para o amplificador de potência a fim de controlar o motor. Por fim, o botão Liga Motor é uma alternativa de segurança para desligar o sistema rapidamente caso ocorra algum problema.

Analogamente a janela do controle Fuzzy possui quase as mesmas opções exceto por algumas adições, conforme as Figuras 5.20 e 5.21:

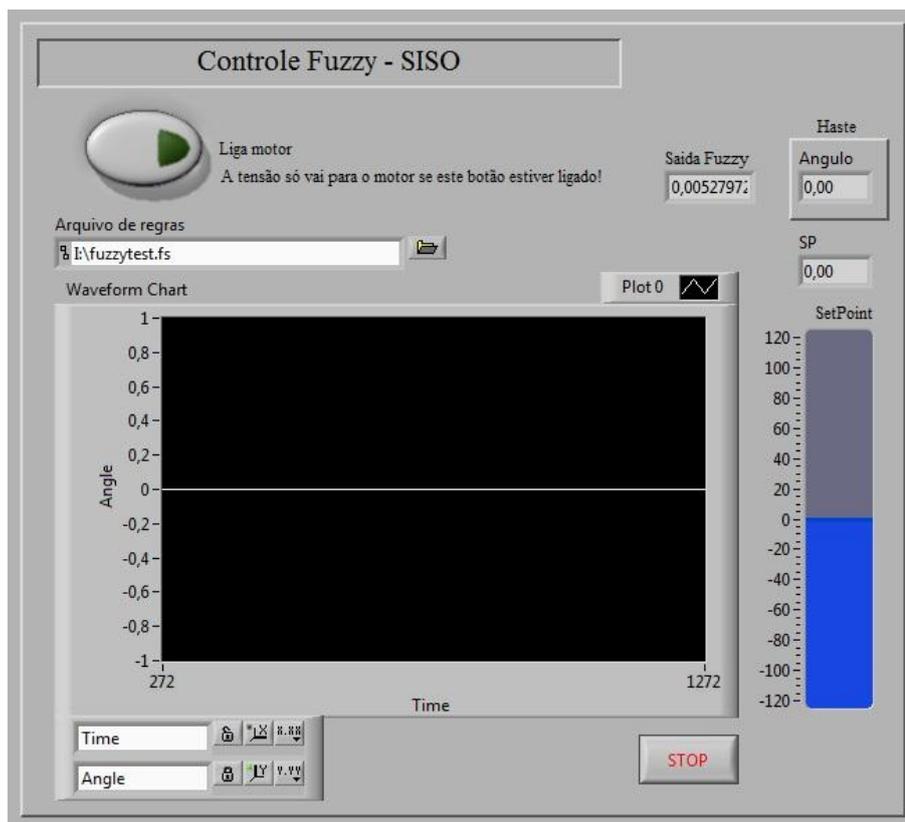


Figura 5.20 - Janela de testes no LabView para controle Fuzzy SISO

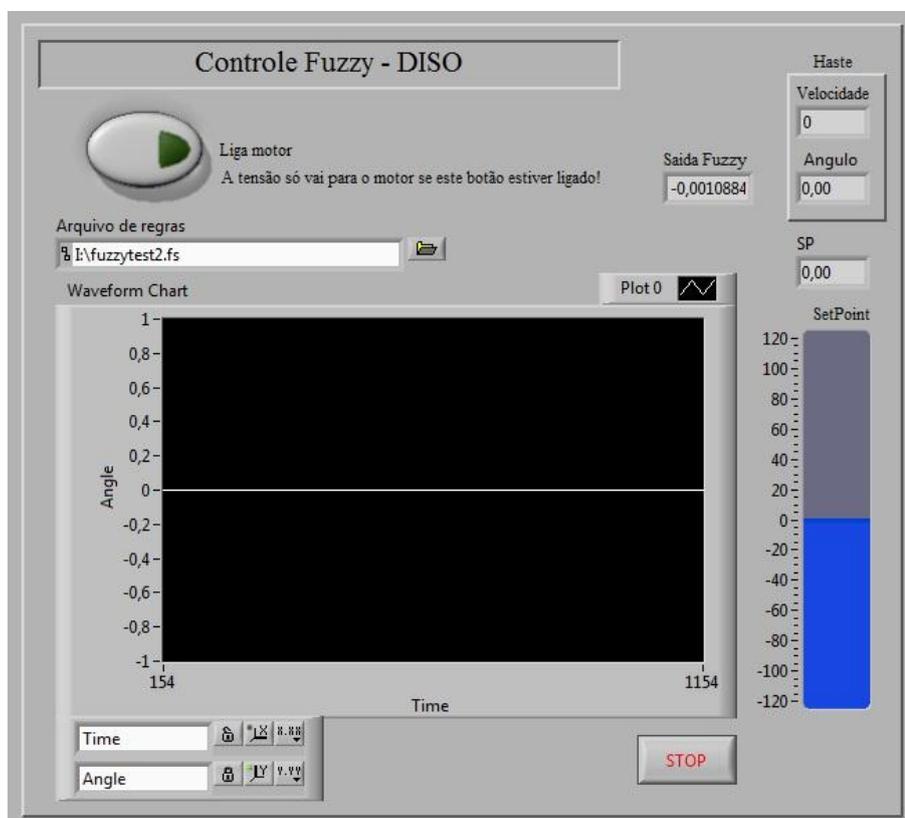


Figura 5.21 - Janela de testes no LabView para controle Fuzzy DISO

As únicas diferenças entre as janelas são no controle Fuzzy a adição da caixa Arquivo de Regras que contém o arquivo com todos os dados do sistema, e no caso do Controle Fuzzy – DISO a adição do indicador de velocidade

5.3.1. Teste com controle P

O primeiro teste foi feito com o controle puramente Proporcional, zerando os outros valores e aplicando o valor 1 em K_c . O fato de essa constante ser negativa é explicado no capítulo 3. A primeira parte do teste foi uma mudança no valor do Set Point, que resultou numa lenta estabilização porém não precisa, conforme a Figura 5.22.

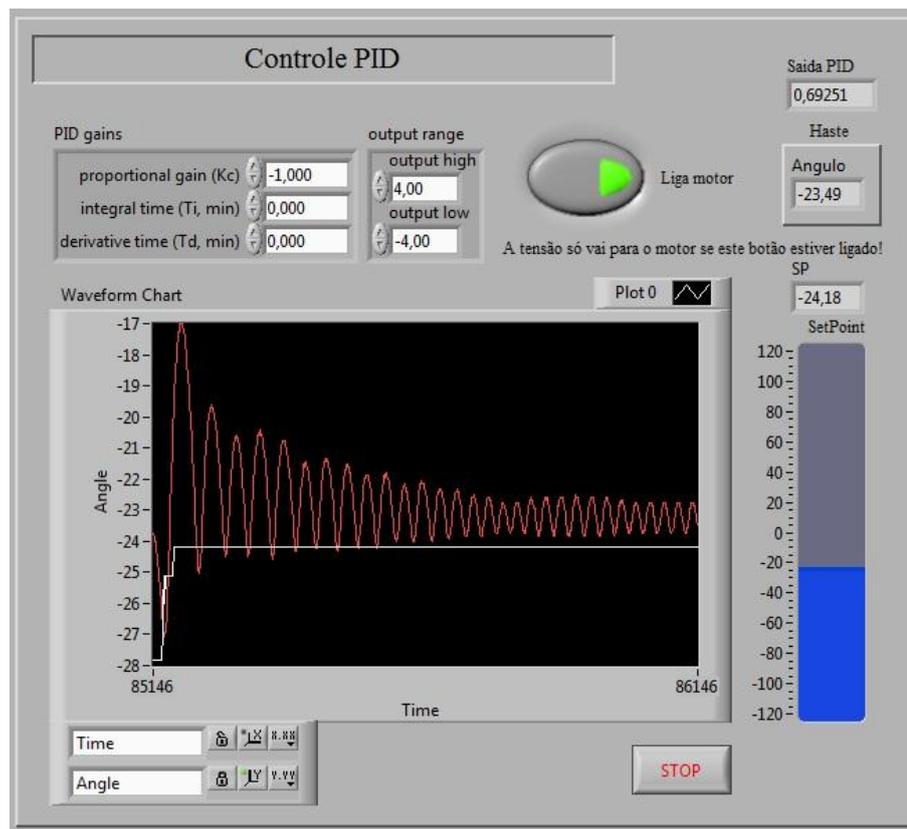


Figura 5.22 – Teste 1 do controle P

A seguir, após o sistema estabilizado aplicou-se uma força na haste que a tirou do ponto de equilíbrio. Neste teste (Figura 5.23) pôde-se notar uma estabilização mais rápida que no anterior, e um menor erro.

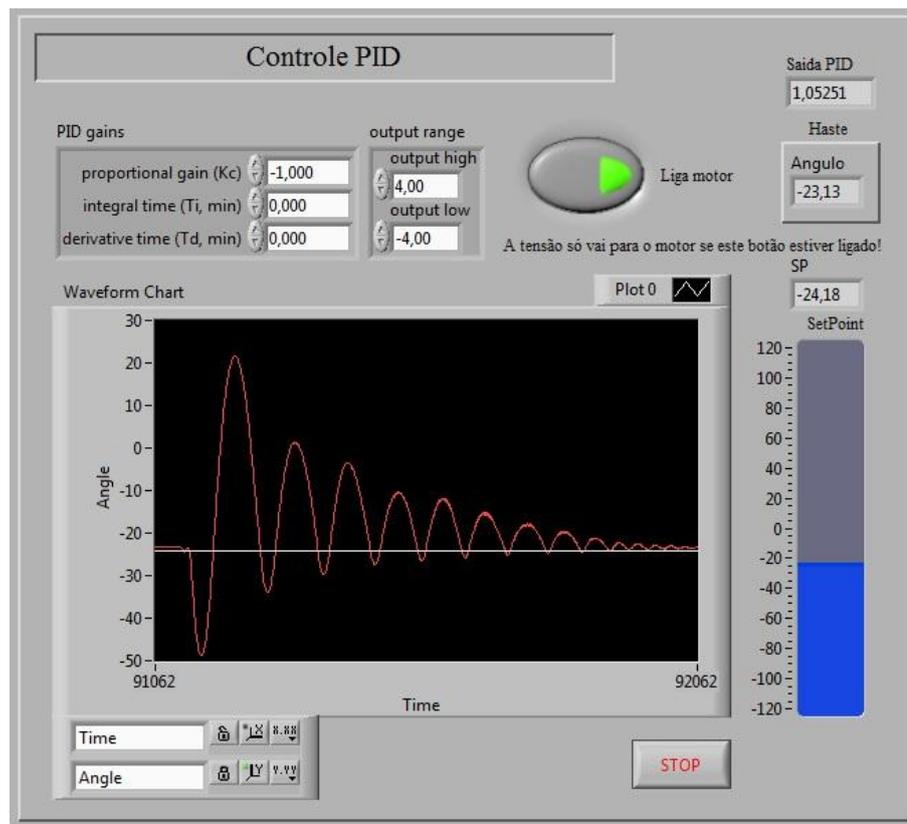


Figura 5.23 – Teste 2 do controle P

5.3.2. Teste com controle PI

Para este teste, manteve-se o ganho proporcional em seu valor igual ao teste anterior e acrescentou-se o valor 0,1 na constante de tempo T_i . Pode-se notar que o efeito senoidal para estabilização que se tinha no primeiro teste (Figura 5.24) começa a ficar menor. Isso se deve ao fato da constante de tempo integrativa atenuar essa oscilação.

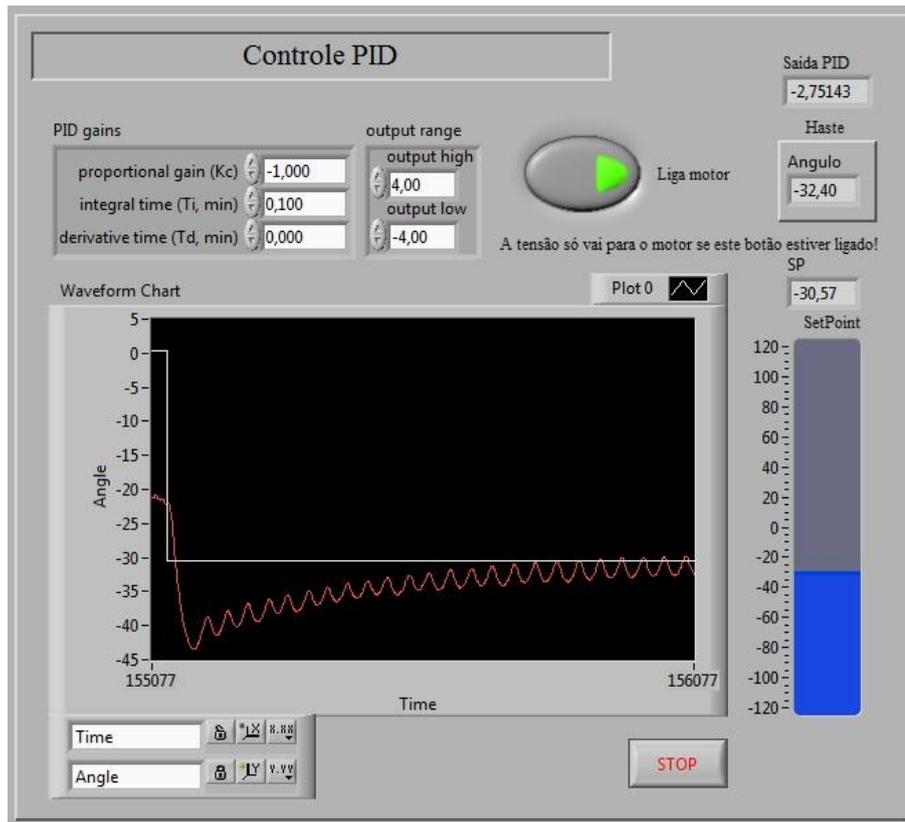


Figura 5.24 – Teste 1 do controle PI

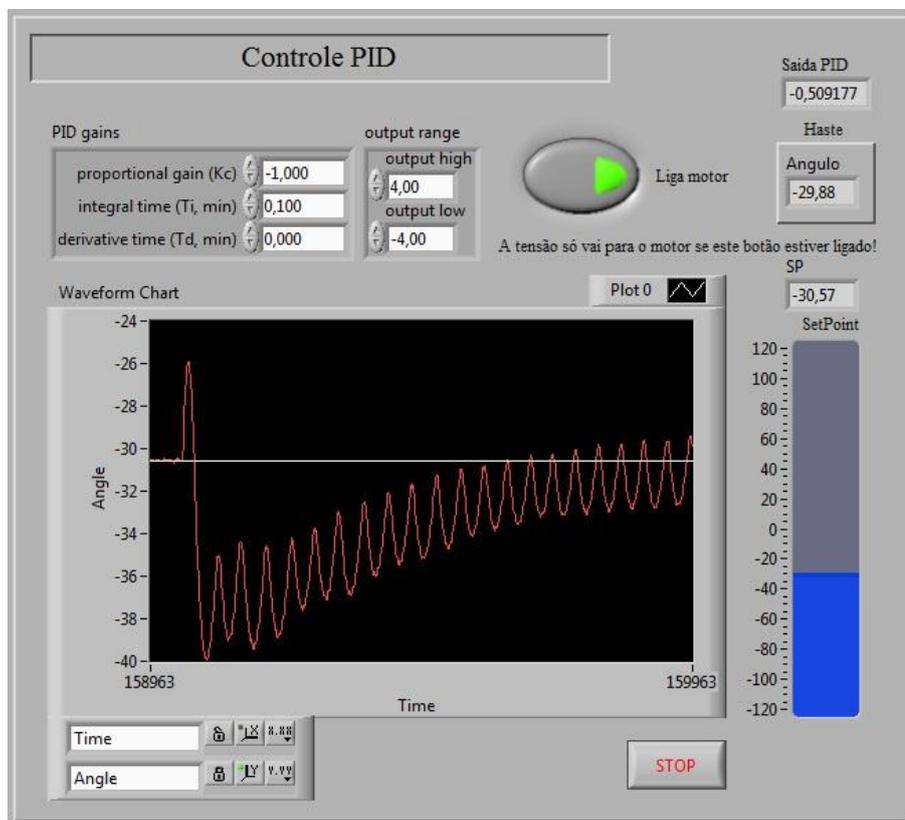


Figura 5.25 – Teste 2 do controle PI

O segundo teste (Figura 5.25) seguiu basicamente o resultado obtido no primeiro teste, com a atenuação da oscilação e uma estabilização mais precisa que o teste com o controle puramente Proporcional.

5.3.3. Teste com controle PID

Finalmente para o controle PID, as constantes de tempo T_i e T_d foram mudadas ambas para 0,01 (que é o menor valor que o LabView aceitou, já que os valores calculados com o método de ZN deu valores menores), que foi o melhor resultado obtido nesse teste prático. Com isso, tem-se o melhor resultado dos controles clássicos, com uma oscilação quase nula e um erro final muito pequeno.

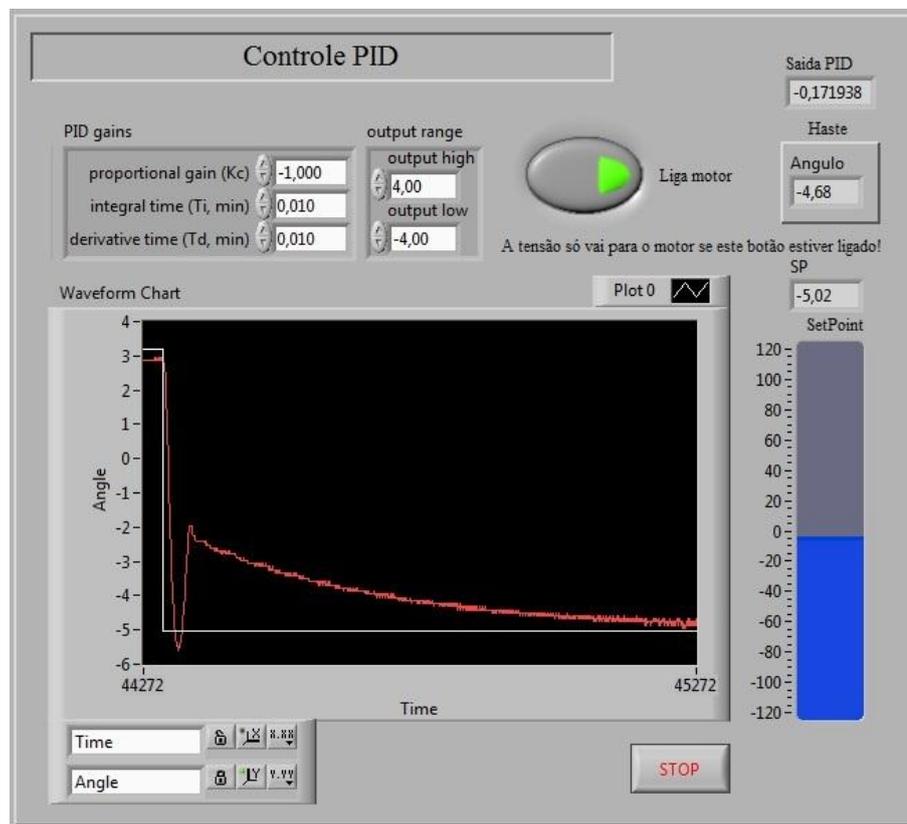


Figura 5.26 – Teste 1 do controle PID

A vibração notada no início até a estabilização no gráfico do primeiro teste (Figura 5.26) pode ser explicada pela alta velocidade do rotor, o que causa um efeito maior perto do ponto de estabilidade. Essa vibração não será analisada em maiores detalhes visto que não prejudica a operação em condições de operação

efetiva, mas deve ser levada em consideração em um estudo de segurança, se por acaso a velocidade do disco escapar da região de projeto.

Para o teste 2, obteve-se o melhor resultado até este ponto do projeto, com uma estabilização rápida. Para testar esse controle, foi aplicado um impacto maior que os anteriores para comprovar a funcionalidade do controle. O erro residual foi bem próximo de zero, como mostra a Figura 5.27

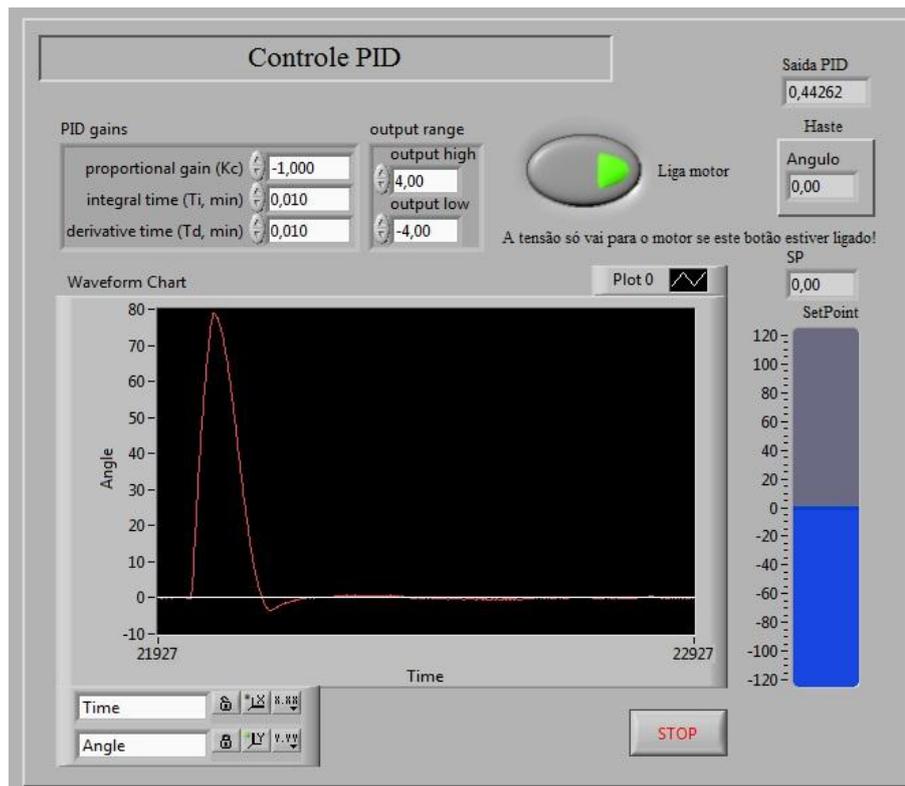


Figura 5.27 – Teste 2 do controle PID

5.3.4. Teste com controle Fuzzy (SISO)

Para o teste com o controle Fuzzy com uma entrada apenas, foi usado o Banco de Regras com suas inferências mostradas no capítulo anterior e mostradas na Figura 5.28.

Pela simplicidade desse teste (Figura 5.29), não se esperava um resultado muito bom, e de fato o que aconteceu foi uma estabilização aproximada para o valor desejado.

Neste teste ocorreu um diferencial em relação aos testes com o controle PID, pois quando o sistema se aproximou da estabilidade, a velocidade do rotor se

reduziu aos poucos até parar. Isso pode ser explicado pela falta do somatório do erro integrativo que existia no controle PI e PID, que fazia, nestes casos, com que a tensão no motor sempre subisse.

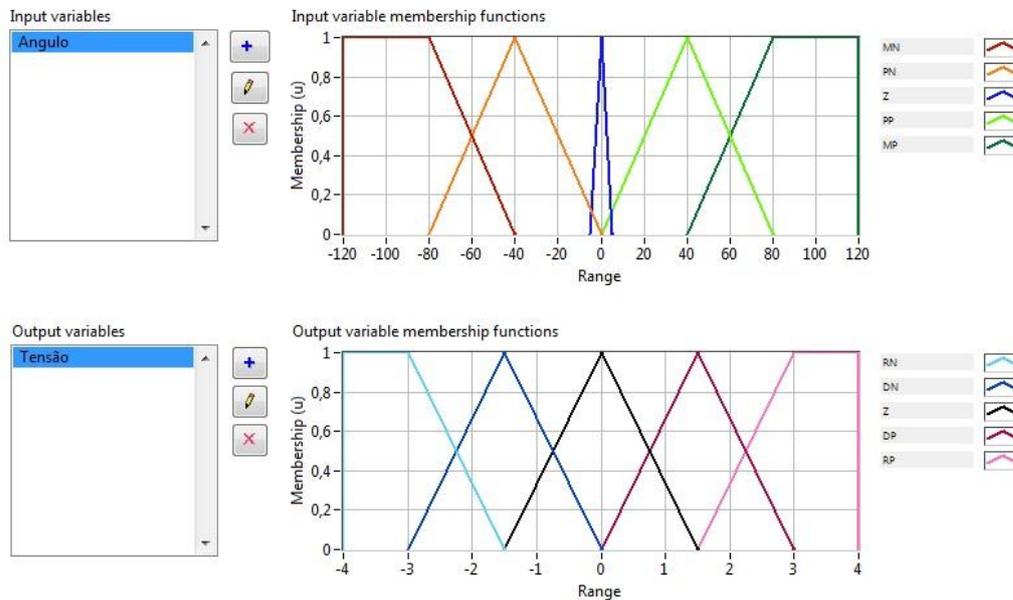


Figura 5.28 – Funções de pertinência da entrada e da saída

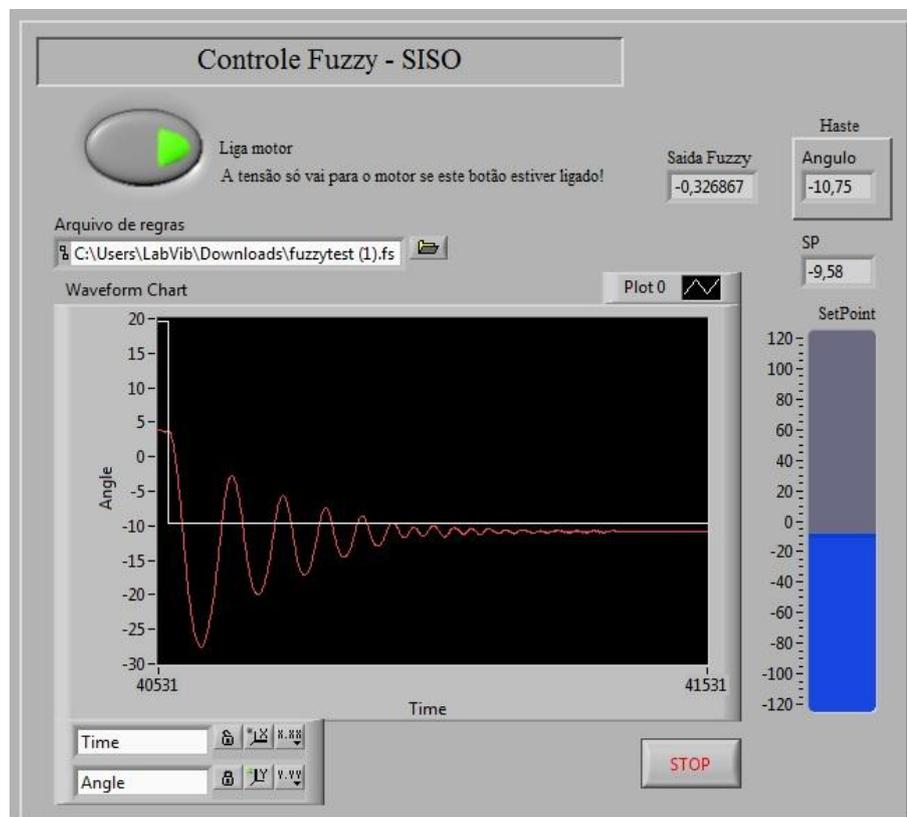


Figura 5.29 – Teste 1 do controle Fuzzy SISO

O segundo teste (Figura 5.30) do controle Fuzzy com uma entrada repetiu o ocorrido no primeiro teste, estabilizando sistema para um valor próximo do desejado. A oscilação também foi bem notada em ambos os testes.

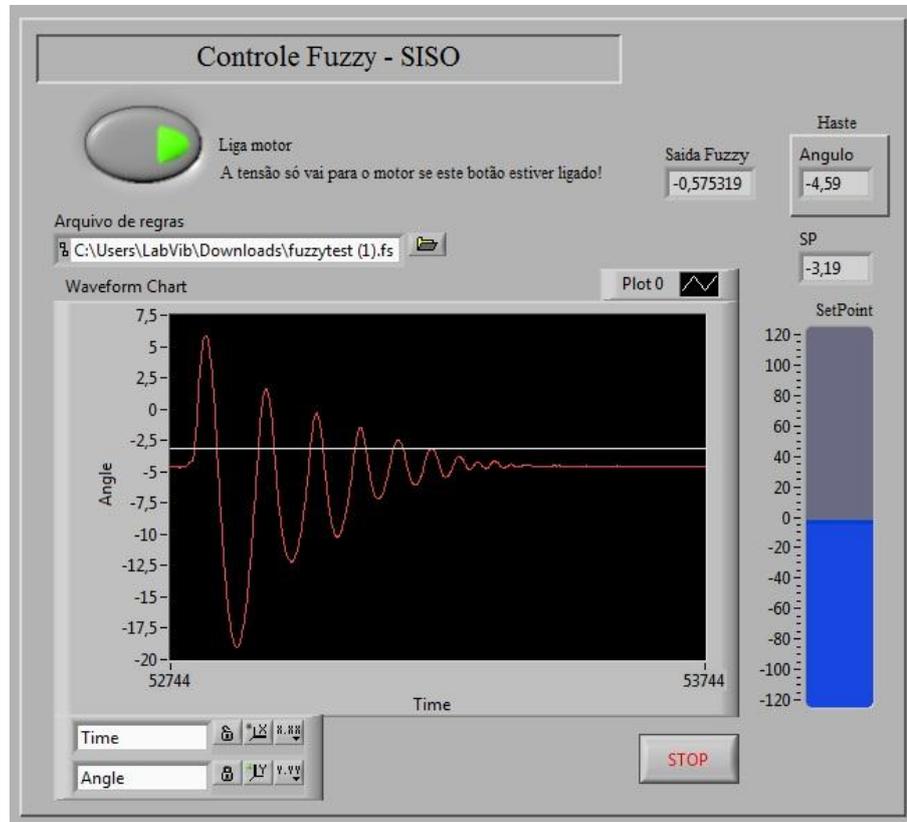


Figura 5.30 – Teste 2 do controle Fuzzy SISO

5.3.5. Teste com controle Fuzzy (DISO)

O último teste teve o diferencial de ter o controle de velocidade e de ângulo (Figuras 5.31 e 5.32), diferentemente dos anteriores que possuíam só o controle a partir do ângulo. Com isso, houve uma precisão maior na estabilização e menor oscilação.

Pode-se dizer que de todos os testes este foi o que obteve uma melhor resposta, tanto com menos oscilação como em comportamento do sistema, pois o rotor parou logo depois de chegar ao ponto desejado (Figuras 5.33 e 5.34).

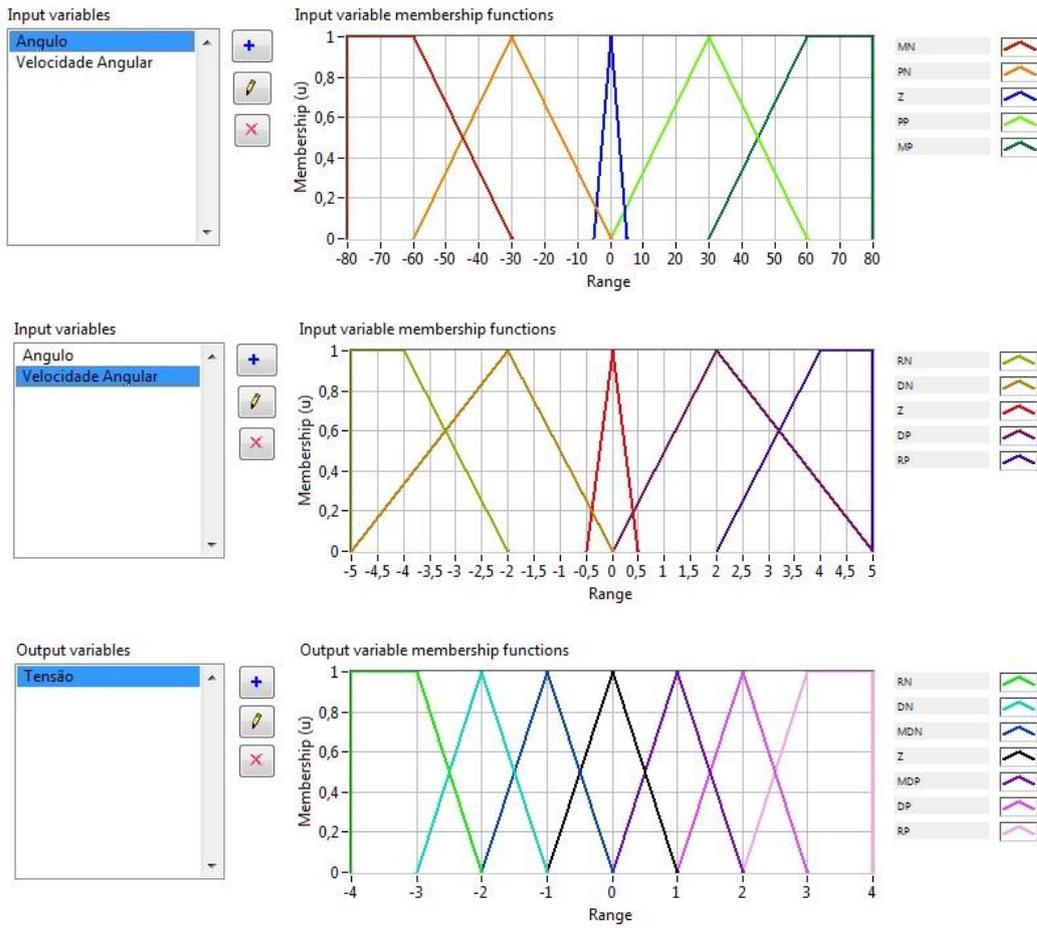


Figura 5.31 - Funções de pertinência das entradas e da saída

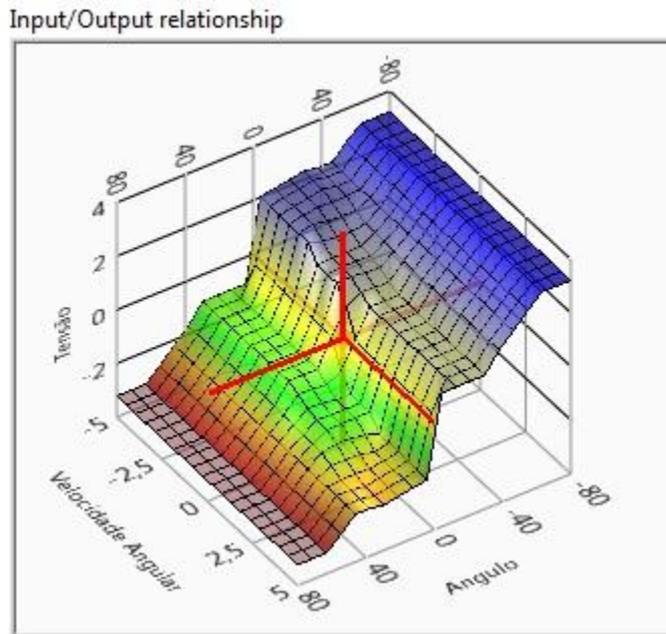


Figura 5.32 – Relação da entrada com a saída e seus valores

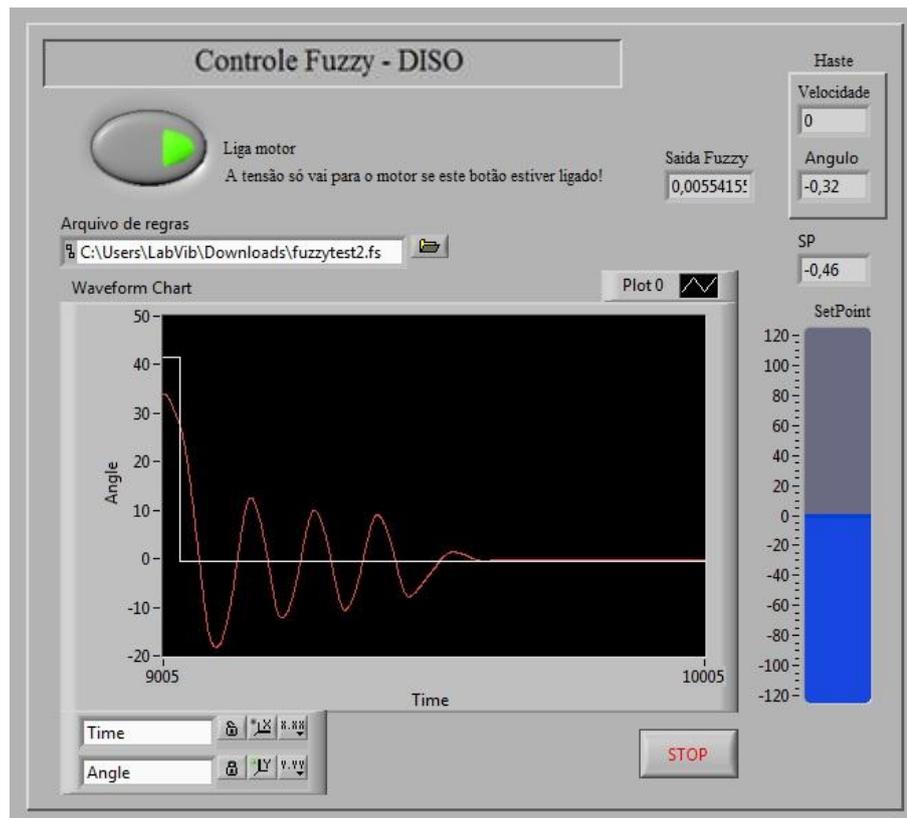


Figura 5.33 – Teste 1 do controle Fuzzy DISO

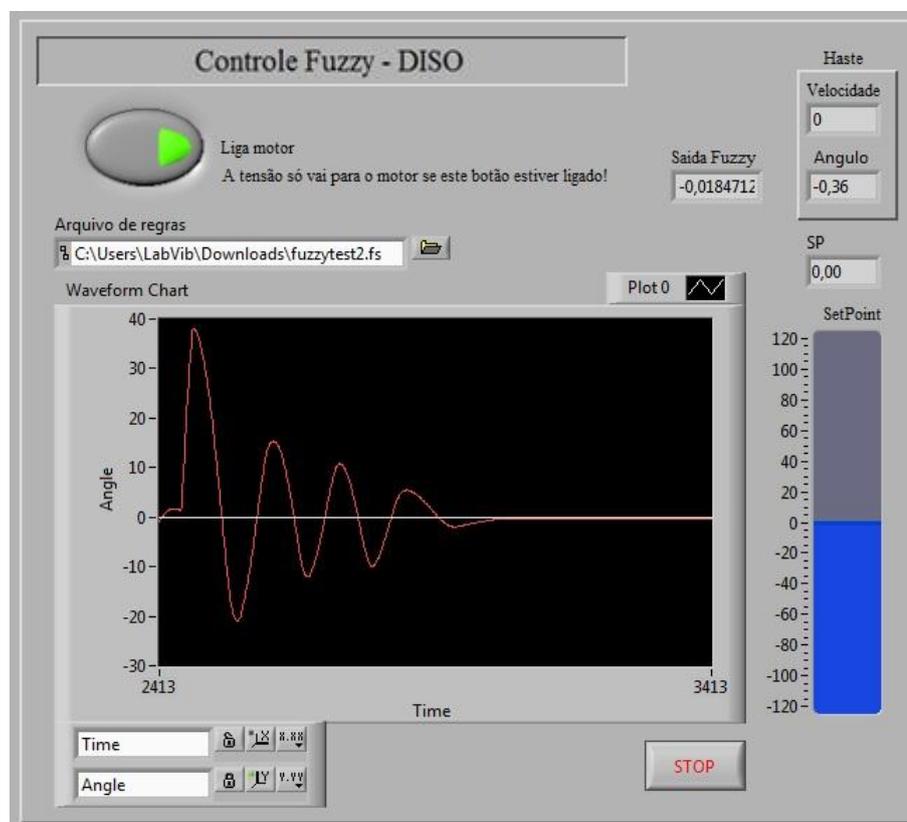


Figura 5.34 – Teste 2 do controle Fuzzy DISO

Finalmente no último teste desse controle, notou-se uma melhor resposta ao impacto na haste, com o menor número de oscilações até atingir o ponto desejado.

Em ambos os testes do controle com duas entradas o erro se manteve em média de 0,2 graus, o que é perto da resolução do encoder que é 0,18 graus.

5.4. Análise dos resultados

Dos 10 testes realizados, os 6 primeiros tiveram por finalidade avaliar o controle PID e suas derivações P e PI, e os 4 últimos testar o controle Fuzzy construído segundo regras atribuídas pelo usuário.

Antes de analisar os números é necessário mostrar que algumas fenômenos que não foram considerados no modelo matemático tiveram influência nos testes, como a vibração que o motor causou na haste, e conseqüentemente pequenos erros no encoder. Por mais que se tentou eliminar essas vibrações, com grandes pesos em torno da base (Figura 5.36), essa vibração continuou e pode ser vista na Figura 5.35.

Além disso, pequenas movimentações de torção foram observadas em certas velocidades do rotor, podendo causar alguns erros de medida, entretanto, na velocidade máxima do motor, que foi quando se obteve a melhor resposta para os controle PID e derivados, não houve muitas vibrações visíveis na haste.

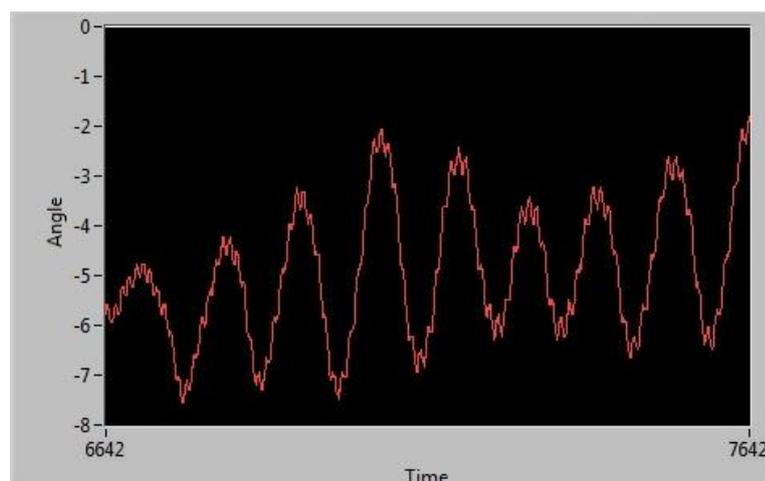


Figura 5.35 – Pequenos erros causados pela vibração



Figura 5.36 – Detalhe de pesos na base para evitar vibração

A falta da utilização de sensores de vibração como acelerômetros não permitiu que esses valores fossem medidos, mas é algo que deve ser considerado para um projeto futuro. Esse trabalho só visou o comportamento dos controles.

Pelos resultados obtidos, pode-se constatar como coerentes com o modelo matemático, o que mostra a semelhança dos testes.

Analisando os testes como um todo, aquele que se mostrou melhor foi o controle Fuzzy DISO, pela sua arquitetura heurística e adaptativa que independe do modelo matemático. Em segundo lugar o controle PID clássico se mostrou muito bom, mas com uma velocidade de estabilização maior que o controle Fuzzy com 2 entradas.