

## CONSTRUÇÃO E PROJETO DE CONTROLE DE UM AERO PÊNDULO, UMA ABORDAGEM BASEADA EM MODELOS

**Alexsandro F. de Barros Jr.** – alexsandro.junior@ee.ufcg.edu.br

Universidade Federal de Campina Grande – UFCG, Laboratório de Instrumentação Eletrônica e Controle - LIEC

R. Aprígio Veloso, 882 – Universitário

58429-900 – Campina Grande – Paraíba

**Rafael B. C. Lima** – rafael.lima@dee.ufcg.edu.br

**Raphael S. Galdino** – raphael.galdino@ee.ufcg.edu.br

**Victor R. B. Maciel** – victor.maciel@ee.ufcg.edu.br

**Christian C. Dias** – christian.dias@ee.ufcg.edu.br

**Resumo:** Neste artigo é descrita a aplicação do *Model-Based Design* na modelagem e controle de um Aero Pêndulo. O MATLAB/Simulink foi escolhido como software de geração de código automático para a implementação da leitura de sensores, acionamento dos atuadores e também dos controladores para o sistema em malha fechada. Os controladores utilizados, P, PD e PID, foram sintonizados utilizando uma ferramenta presente no MATLAB que permite o ajuste da dinâmica de resposta da planta controlada. O objetivo dos métodos utilizados foi otimizar o tempo de desenvolvimento do produto final e facilitar sua aplicação em ambientes educacionais.

**Palavras-chave:** MATLAB, Simulink, Model-Based Design, Aero Pêndulo.

### 1. INTRODUÇÃO

A experiência em laboratórios é uma parte fundamental no ensino de engenharia. Com o avanço tecnológico presenciado nas últimas duas décadas se torna necessário o desenvolvimento de novas plataformas de testes que abordem os assuntos pertinentes a um curso de engenharia (ENIKOV, 2012). Plataformas reais e simulações apresentam pontos positivos e negativos, dito isso, ao juntar os dois métodos consegue-se obter um resultado ainda melhor durante a realização de experimentos na área de Engenharia.

O aero pêndulo foi desenvolvido para que permita uma aproximação do aluno com a planta estudada, de forma a facilitar o entendimento da implementação de um controlador, utilizando da matemática envolvida, assim como visualizando o funcionamento do sistema controlado. Com equipamentos de fácil aquisição, como o Arduino, potenciômetros e motores, foi possível produzir uma plataforma educacional que, com o acoplamento do hardware com o software MATLAB/Simulink permite uma implementação ágil do sistema de controle.

Uma das características fundamentais desse trabalho é a utilização do *Model-Based Design*, que consiste na aplicação de modelos para as simulações facilitando a etapa de prototipagem física (AARENSTRUP, 2015), para isso utilizou-se o Simulink como uma ferramenta de simulação. O Simulink também foi utilizado para diminuir o tempo de codificação, pois nele é possível gerar automaticamente o código para o Arduino (MATHWORKS, 2018).

Os conteúdos são divididos da seguinte forma. A seção 2 descreve a plataforma do aero pêndulo e os componentes utilizados. Seção 3 introduz a teoria de *Model-Based Design*, assim

como apresenta o método de desenvolvimento com geração de código automático. Nas seções 4 e 5 os modelos matemáticos para a planta e para o controlador são definidos, respectivamente. A seção 6 apresenta os resultados obtidos experimentalmente com a plataforma real. E concluindo o artigo a seção 7.

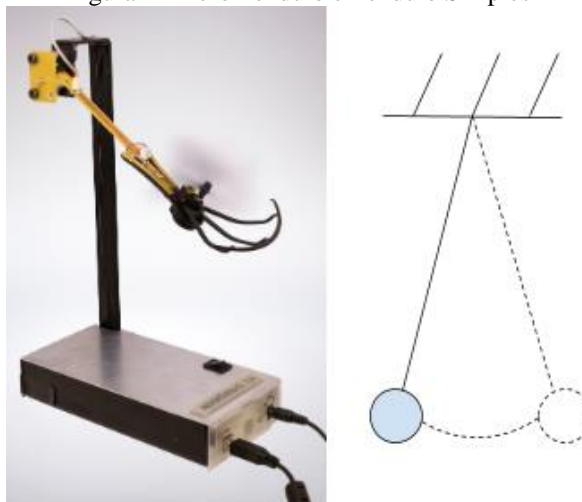
## 2. DESCRIÇÃO DA PLANTA

### 2.1. Princípios de funcionamento

Para explicar o funcionamento do aero pêndulo pode-se utilizar como analogia um pêndulo simples, uma massa acoplada a um fio inextensível e fixo em uma extremidade a uma superfície (JOB, 2015). No sistema desenvolvido o atuador, formado por um conjunto motor CC e hélice, faz com que o eixo se movimente, produzindo uma força que atua de forma contrária à gravidade, dependendo da posição do motor. Na Figura 1 é mostrado o aero pêndulo em funcionamento ao lado de um pêndulo simples para ilustrar a semelhança.

Normalmente os objetivos dos sistemas de controle se resumem a rastreamento de referência ou rejeição de perturbações. Para atingir tais objetivos, torna-se necessária a presença de um componente controlável, no caso do aero pêndulo o atuador é o conjunto de motor e hélice, alimentado por um sinal PWM. É possível controlar a velocidade do motor variando a largura do pulso.

Figura 1 - Aero Pêndulo e Pêndulo Simples



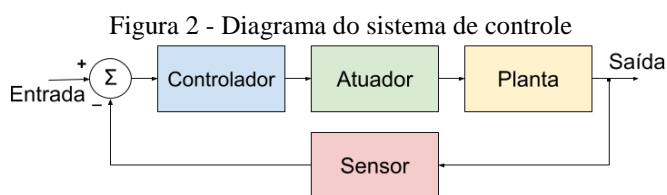
Fonte: Elaboração Própria

### 2.2. Características físicas

O aero pêndulo desenvolvido é composto por um conjunto motor hélice acoplado a um potenciômetro através de uma haste de 18cm, constituída de uma placa de fibra de vidro, onde as trilhas presentes na mesma servem para alimentação do motor. O potenciômetro é fixado num braço de chapa de aço. Esse material também é usado na caixa presente na base da plataforma onde o Arduino Uno e o circuito de alimentação do motor estão incluídos. Também foram adicionados limitadores de movimento angular na haste para que eventuais distúrbios não danifiquem a plataforma.

### 2.3. Instrumentação

O aero pêndulo desenvolvido neste trabalho apresenta componentes que são fundamentais para o funcionamento da maior parte dos sistemas de controle. Na Figura 2 é mostrado o diagrama de blocos do sistema de controle utilizado para o aero pêndulo.



Fonte: Elaboração própria

O potenciômetro B10K foi utilizado como sensor angular. De acordo com o seu *datasheet* a variação de resistência é linear em relação ao ângulo de giro do eixo, que possui uma limitação de 270 graus, com isso, conseguimos utilizá-lo para realizar a medição do ângulo na saída.

O motor caracteriza-se por possuir dimensão de 7 mm de diâmetro por 20 mm de comprimento, engrenagem de 9 dentes, fonte de alimentação de 5V CC e corrente de 2A. Com aproximadamente 5250 RPM na hélice, fazendo então a relação de engrenagens, onde a engrenagem da hélice possui 55 dentes, obtemos a rotação de aproximadamente 32000 RPM para o motor.

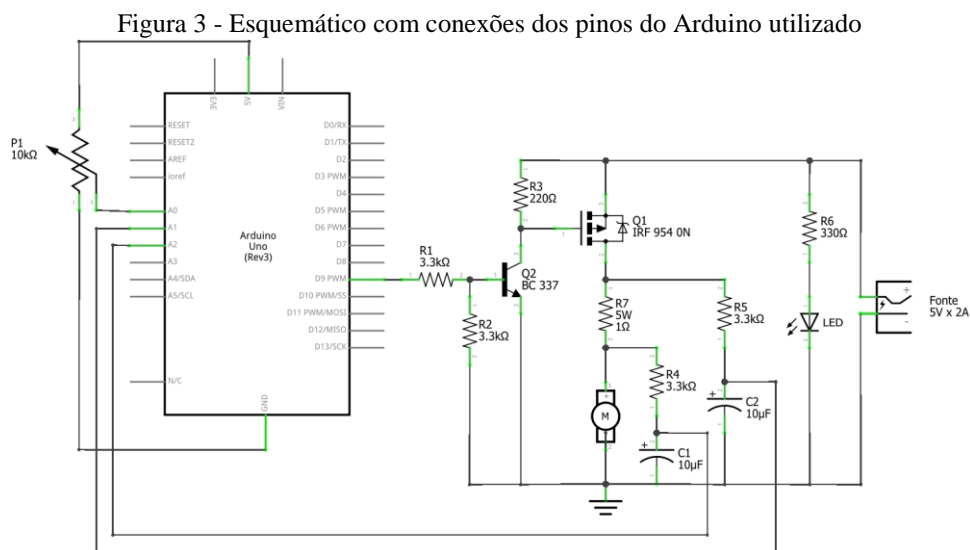
Para embarcar a lógica de controle utiliza-se um Arduino Uno, um micro controlador com entradas e saídas digitais e analógicas. Com programação baseada na linguagem C, possui uma interface simples, mas de grande confiabilidade (ARDUINO, 2018).

A Figura 3 mostra o circuito eletrônico do sistema, que consiste num driver PWM para o motor. Os componentes e os pinos utilizados no Arduino estão descritos no quadro 1.

Quadro 1 – Componentes do circuito eletrônico

Resistores		Transistores	
R1=R2=R4=R5	3,3kΩ	Q1	BC 337
R3	220Ω	Q2	IRF 954 0N
R6	330Ω	Arduino	
R7	1Ωx5W	Pino	Dado Enviado/Recebido
P1	10kΩ	A0	Potenciômetro
Capacitores		A1	Tensão do sinal PWM
C1	10μF	A2	Tensão no motor CC
C2	10μF	Digital ~9	Sinal PWM gerado

Fonte: Elaboração Própria



Fonte: Elaboração Própria



### 3. PROJETO BASEADO EM MODELO (*MODEL-BASED DESIGN*)

#### 3.1. Introdução teórica

No desenvolvimento de novas tecnologias é comum o uso de protótipos físicos, que são utilizados para realização de testes e validações, e documentos que descrevem o funcionamento do sistema desenvolvido.

O *Model-Based Design* (MBD) apresenta uma abordagem centrada no modelo, ou seja, em vez de depender de protótipos físicos e documentos o MBD usa modelos matemáticos, simulações, processamento de sinais, entre outros, ao longo do desenvolvimento (AARENSTRUP, 2015). Em sistemas embarcados e algoritmos os modelos sempre foram uma parte essencial no processo de design do projeto, logo, quando usado efetivamente, o MBD proporciona que o modelo criado possa ser melhorado e otimizado ao longo do tempo. Com isso, torna-se possível a realização de simulações de algoritmos ou da conexão entre componentes elétricos, mecânicos e embarcados, antes mesmo do código embarcado ser escrito ou que o hardware seja fabricado e esteja disponível para testes (EINFOCHIPS, 2018).

#### 3.2. Geração automática de código

Um dos elementos que caracteriza o uso do *Model-Based Design* (MBD) é a substituição dos processos manuais por processos automatizados (AARENSTRUP, 2015). Nesse artigo será utilizado o ambiente do MATLAB/Simulink para realizar as simulações dos modelos matemáticos.

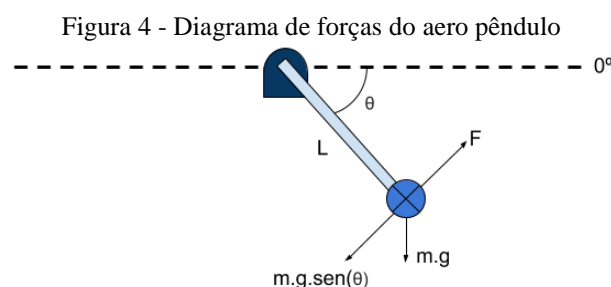
No Simulink são encontradas as mesmas funções existentes na IDE padrão do Arduino, porém dispostas em formato de blocos para conexão com qualquer elemento presente no Simulink, tornando a programação para o Arduino bem intuitiva (MATHWORKS, 2018).

Com isso, podemos utilizar o modelo desenvolvido em MBD para o sistema e acoplar os elementos do Arduino diretamente. O Simulink cria uma conexão de comunicação entre o Hardware e o Computador, permitindo que o código gerado seja transmitido para o Arduino e que os dados possam fluir entre os dois, assim podemos ter o monitoramento constante do que está acontecendo com o modelo criado ao ser aplicado no sistema embarcado.

### 4. MODELAGEM DINÂMICA

O diagrama de forças do aero pêndulo é mostrado na Figura 4. Após a aplicação de uma tensão no motor, a força gerada impulsiona o eixo para cima. Ao controlar a tensão enviada para o motor é possível ajustar a posição angular do eixo, logo, a função de transferência que relaciona posição angular com tensão é de suma importância.

No diagrama temos um ponto fixo localizado na parte superior e a marcação para o horizonte adotado. Foi considerado que a massa está localizada na extremidade do braço, marcado pelo círculo azul, onde também está localizada a hélice e motor que geram a força  $F$ . A haste de comprimento  $L$  foi considerada com massa desprezível (MIT, 2018).



Fonte: Elaboração Própria

#### 4.1. Sistemas de equações

De acordo com as Leis de Newton e Momento Angular, a equação do movimento para o aero pêndulo é a seguinte (JOB, 2015):

$$J \cdot \frac{d^2\theta}{dt^2} + c \cdot \frac{d\theta}{dt} + m \cdot g \cdot L \cdot \sin(\theta) = F \quad (1)$$

Onde,

$m$  = Massa do aero pêndulo (kg)

$L$  = Comprimento do eixo (m)

$\theta$  = Posição angular (rad)

$c$  = Coeficiente de amortecimento viscoso (Nms/rad)

$J$  = Momento de Inércia (kgm<sup>2</sup>)

$g$  = Aceleração da gravidade (m/s<sup>2</sup>)

$F$  = Força gerada pelo conjunto motor e hélice (N)

Para esse tipo de sistema deve-se considerar que existe uma força contrária ao movimento do aero pêndulo que é gerada pelo ar, ou seja, essa força é proporcional à velocidade do corpo. Na Equação (1) o Coeficiente de Amortecimento Viscoso ( $c$ ) representa o fator de proporcionalidade dessa força de atrito do ar (MOHAMMADBAGHERI, 2011).

O Momento de Inércia ( $J$ ) mede a forma em que a massa do sistema está distribuída em torno de um eixo, ou seja, ao realizar um movimento angular o corpo tem a tendência de permanecer no estado atual devido a sua massa (VON ZUBEN, 2010). Para realizar a medição dessa variável usamos a Equação (2),

$$J = m \cdot L^2 \quad (2)$$

Linearizando o sistema do aero pêndulo em torno do ponto de operação que é o horizonte (0°), a aproximação para pequenos ângulos pode ser aplicada na Equação (1),

$$J \cdot \frac{d^2\theta}{dt^2} + c \cdot \frac{d\theta}{dt} + m \cdot g \cdot L \cdot \theta = F \quad (3)$$

Assim, a função de transferência que relaciona a posição angular e a força produzida pelo aero pêndulo é dada pela Equação (4),

$$\frac{\theta(s)}{F(s)} = \frac{1/J}{s^2 + (c/J) \cdot s + (m \cdot g \cdot L)/J} \quad (4)$$

Nesse contexto, medir a força gerada pelo Aero Pêndulo se torna inviável, pois seria necessária a instalação de um dinamômetro, além disso, a variável controlada no sistema é a tensão enviada para o motor em questão. Logo, torna-se necessário realizar a caracterização do motor, medindo a força gerada ( $F$ ) pela tensão aplicada ( $V$ ) e obtendo um fator de proporcionalidade denominado  $\gamma$  (gamma) (JOB, 2015).

$$F(s) = \gamma \cdot V(s) \quad (5)$$

Substituindo a Equação (2) e o fator de proporcionalidade da Equação (5) na função de transferência do sistema, Equação (4), obtida anteriormente, tem-se uma nova função de transferência que relaciona tensão e ângulo.

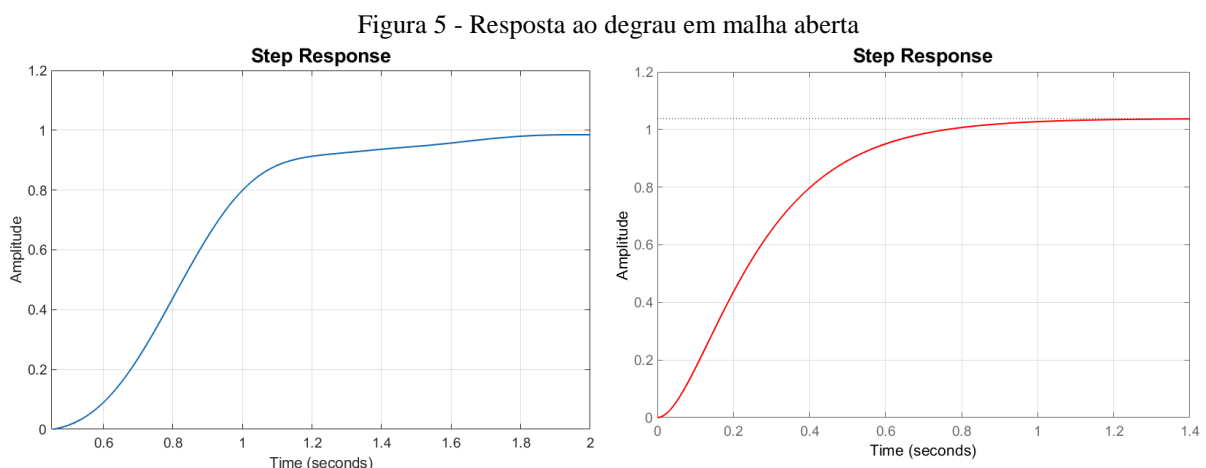
$$\frac{\theta(s)}{V(s)} = \gamma \cdot \frac{1/(m \cdot L^2)}{s^2 + (c/(m \cdot L^2)) \cdot s + g/L} \quad (6)$$

Os parâmetros do modelo foram medidos com auxílio de trena para medir o comprimento do braço e uma balança para determinar o peso da planta. O valor para o Coeficiente de Amortecimento Viscoso ( $c$ ) está intimamente relacionado com a viscosidade do fluido, nesse caso o ar, logo, possui um valor definido.

Para determinar o  $\gamma$  (gamma) foi realizado um experimento com balança de precisão e uma massa conhecida, onde ao medir o quanto de massa foi levantada pelo Aero Pêndulo e com a tensão necessária para isso, consegue-se obter um valor aproximado.

$$L = 0,18 \text{ m}, m = 0,015 \text{ kg}, g = 9,81 \text{ m/s}^2, c = 0,0076 \text{ Nms/rad}, \gamma = 0,0275.$$

Na Figura 5 estão representadas as respostas ao degrau em malha aberta para a função de transferência presente na Equação (6) e para o modelo real em malha aberta, aplicando um degrau de tensão na entrada. Em vermelho temos o gráfico referente a Equação (6) e em azul para a planta real.



Fonte: MATLAB R2018a

## 5. CONTROLE

Em sistemas de controle deseja-se que a planta definida siga um comportamento padrão, normalmente chamado de referência, e para que isso ocorra um controlador é adicionado ao sistema (BATISTA, 2014).

Na plataforma desenvolvida utiliza-se um dos modelos mais didáticos e que já está implementado em um bloco do Simulink, o controlador PID. No MATLAB existe uma ferramenta de sintonia para controladores P, PI, PD e PID que será abordada nas próximas seções.

### 5.1. Formulação

Os Controladores P, PD e PID determinam o sinal que deve ser enviado para a planta de acordo com o erro entre a saída do sistema e a referência escolhida. No PID o erro passa por três etapas, um ganho proporcional, um integrador e um derivador, que são responsáveis por ajustar os parâmetros de desempenho citados anteriormente. Ao ajustar os ganhos  $K_p$ ,  $K_d$  e  $K_i$ , podemos definir a dinâmica da saída ao tentar acompanhar a referência.



## 5.2. Sintonia

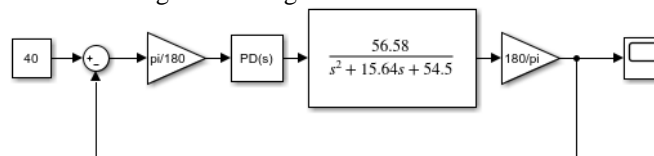
Quando controladores são utilizados deve-se procurar por valores de ganhos que satisfaça a especificação de desempenho desejada em malha fechada, esse processo é chamado de sintonia do controlador. Tendo em mãos a função de transferência de malha fechada do sistema, pode-se então usar métodos matemáticos como Ziegler-Nichols, SIMC, entre outros, ou então recorrer à softwares que calculam esses valores por meio de simulações (BATISTA, 2014).

A *toolbox* chamada 'PID Tuner' permite calcular, a partir da função de transferência de malha aberta, os valores dos ganhos para um controlador. Ajustando o tempo de resposta e a robustez do sinal de saída, pode-se obter diferentes valores para  $K_p$ ,  $K_i$  e  $K_d$ . Além disso, são gerados gráficos que mostram a resposta da função de transferência em malha fechada e também são fornecidos os valores dos parâmetros de desempenho para o controlador obtido.

## 5.3. Equações de malha fechada

De posse das equações de malha aberta e das equações do controlador, pode-se então obter o modelo de equações de malha fechada para o sistema do Aero Pêndulo. Na Figura 6 temos o diagrama de malha fechada do sistema usado no Simulink e na tabela 1 temos os ganhos sintonizados pelo PID Tuner Toolbox para os controladores. A Figura 7 mostra os gráficos de resposta com os controladores P, PD e PID sintonizados, para uma referência de 40°.

Figura 6 - Diagrama de malha fechada



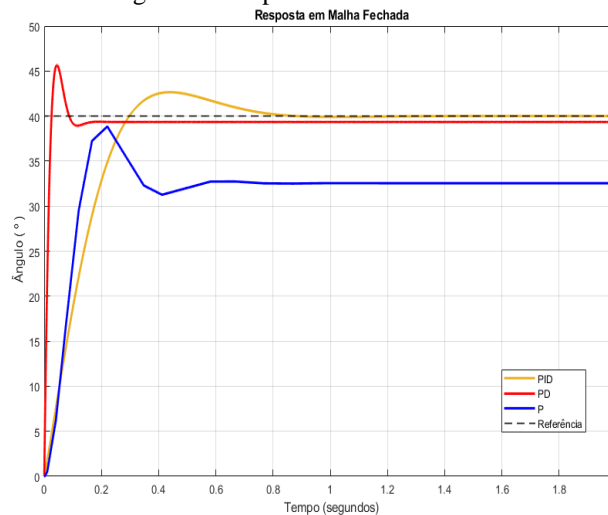
Fonte: MATLAB R2018a

Tabela 1 – Ganhos sintonizados do controlador

Controlador	P	PD	PID
$K_p$	4,205	57,931	1,9
$K_d$	-	1,0558	0,0788
$K_i$	-	-	9,694
N	-	8436,8264	843,7

Fonte: Elaboração Própria

Figura 7 - Respostas de malha fechada

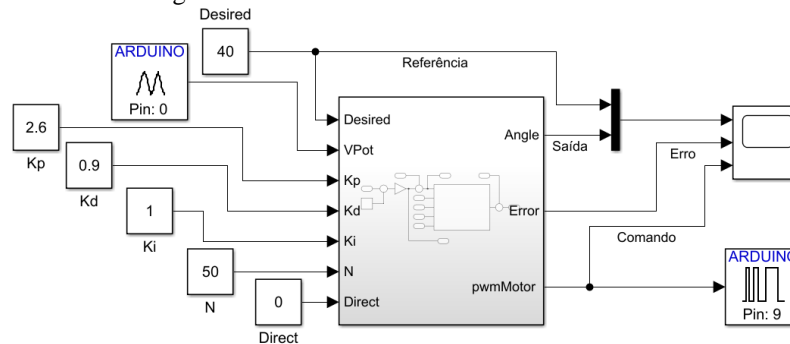


Fonte: MATLAB R2018a

## 6. RESULTADOS OBTIDOS

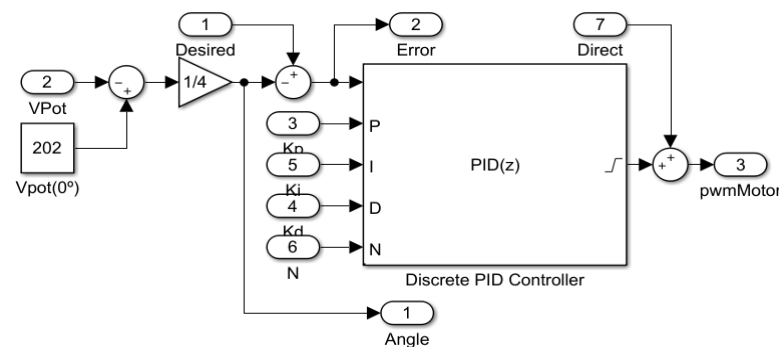
O modelo desenvolvido no Simulink está configurado para executar em modo 'External' permitindo a comunicação em tempo real com o Arduino Uno, tornando possível então a plotagem dos gráficos com dados de erro, ângulo medido e do sinal enviado para o atuador. As Figuras 8 e 9 mostram o modelo implementado.

Figura 8 - Modelo no Simulink do Aero Pêndulo



Fonte: MATLAB R2018a

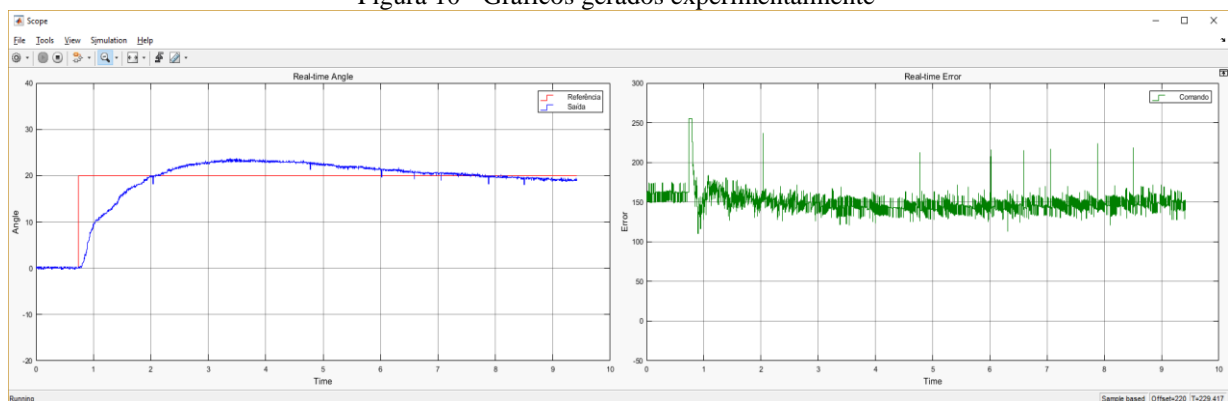
Figura 9 - Bloco central do modelo



Fonte: MATLAB R2018a

Os valores de ganhos para o PID aplicado na planta real são,  $K_p = 2,6$ ,  $K_d = 0,9$ ,  $K_i = 1$  e  $N = 50$ . Na Figura 10 são mostrados os gráficos gerados, respectivamente, ângulo e referência, erro e comando para o motor. Para esse teste foi considerada como posição angular inicial  $0^\circ$  e a referência desejada foi  $20^\circ$ .

Figura 10 - Gráficos gerados experimentalmente



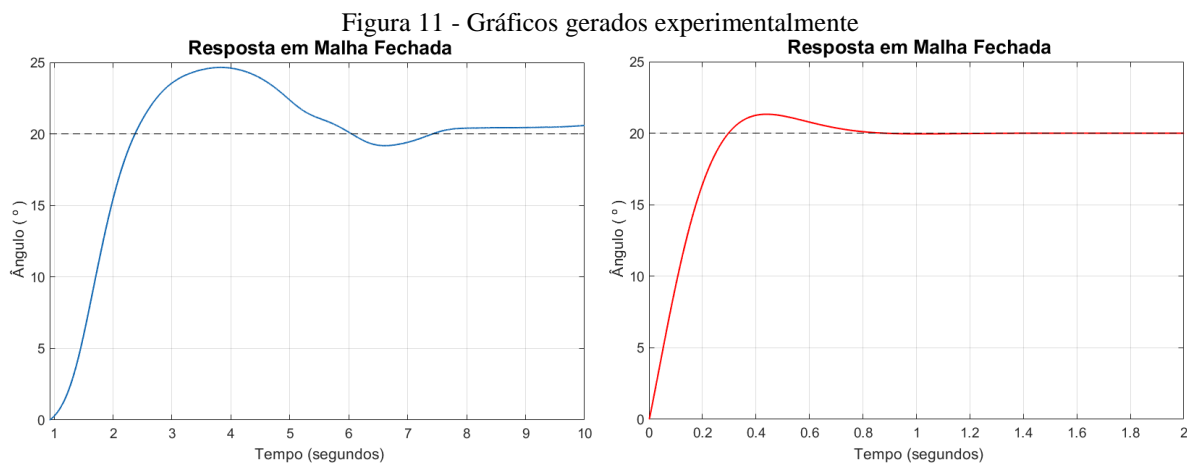
Fonte: MATLAB R2018a



Para realizar a medição do ângulo, usando a tensão medida no Potenciômetro pelo Arduino, usamos a Equação (10), onde, 202 equivale ao valor medido quando o braço está na posição 0°. Em seguida é calculado o erro entre a referência escolhida (*Desired*) e o ângulo medido (*Angle*), que então é enviado para o controlador. O valor de saída do controlador é saturado, entre 0 e 255, pois são os valores de *duty cycle* aceitáveis para o sinal PWM.

$$Angle = (202 - V_{pot})/4 \quad (10)$$

Na Figura 11 podemos analisar os gráficos obtidos na planta real, em azul, e na simulação, em vermelho, para a resposta em malha fechada. Para esse teste foi considerada a referência saindo de 0° até 20°. Percebe-se que as dinâmicas de resposta são praticamente iguais, diferenciando pelo tempo de subida e acomodação, que por questões de atrito mecânico entre as peças, atrito com relação ao ar e atrasos de comunicação entre Arduino, Simulink e Motor na planta real geram diferenças com relação ao modelo matematicamente simulado.



## 7. CONCLUSÃO

Conclui-se então que esse estudo possibilitou o desenvolvimento de uma plataforma didática para ensino de modelagem e sintonia de sistemas de controle, utilizando o software MATLAB/Simulink como ferramenta de codificação e simulação e o aero pêndulo como planta a ser controlada. Devido ao baixo custo dos componentes utilizados torna-se acessível à diversas aplicações de ensino.

Os resultados obtidos mostram que os controladores P, PD e PID apresentam resultados satisfatórios pois tornaram possível o controle da planta. Além disso, o processo de sintonia foi facilitado pelo 'PID Tuner Toolbox' e a geração de código automático permite a implementação do controlador sem a necessidade de conhecer a linguagem nativa de programação do micro controlador, diminuindo o tempo de implementação.

Podem ser aplicados ao aero pêndulo outras modelagens e outros métodos de sintonia, obtendo assim resultados diferentes, porém, podem ser utilizadas as mesmas ferramentas e a mesma plataforma deste trabalho para implementação do controlador, justificando assim o caráter didático da plataforma.

O Aero Pêndulo construído será aplicado em laboratórios da área de controle onde os alunos irão realizar testes de malha aberta, modelar o sistema matematicamente, sintonizar controladores e por fim testar o sistema em malha fechada, fixando assim o conhecimento adquirido em sala de aula durante as aulas teóricas com uma plataforma didática de baixo custo.

## REFERÊNCIAS

AARENSTRUP, Roger. **Managing Model-Based Design**. MathWorks, Inc. 2015.

ARDUINO, **Arduino Uno Rev3**. Disponível em: <https://store.arduino.cc/usa/arduino-uno-rev3>. Acesso em: 8 set. 2018.

BATISTA, L.; FERNANDES, G.; MAIA, A. **Estudo Comparativo de Técnicas de Sintonia de Controladores PID para Sistemas de Primeira Ordem com Atraso**. ABCM Symposium Series in Mechatronics – Vol. 6, 2014.

MIT. **Courses edX: Introduction to Control System Design – A First Look**. Disponível em: <https://edx.org/course/introduction-control-system-design-first-mitx-6-302-0x/>. Acesso em: 25 ago. 2018.

EINFOCHIPS. **Why is Model-Based Design important in Embedded Systems**. Disponível em: <https://www.einfochips.com/blog/why-is-model-based-design-important-in-embedded-systems/>. Acesso em: 10 set. 2018.

ENIKOV, E. T. CAMPA, G. **Mechatronic Aeropendulum: Demonstration of Linear and Nonlinear Feedback Control Principles With MATLAB/Simulink Real-Time Windows Target**. IEEE Transactions on Education. v. 55, n. 4, 2012.

JOB, M. M. JOSE, P. S. H. **Modeling and Control of Mechatronic Aeropendulum**. IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems. 2015.

MATHWORKS. **Simulink Support Package for Arduino Hardware**. Disponível em: <https://www.mathworks.com/help/supportpkg/arduino/>. Acesso em: 5 set. 2018.

MOHAMMADBAGHERI, A. YAGHOUBI, M. **A New Approach to Control a Driven Pendulum with PID Method**. UKSim 13th International Conference on Modelling and Simulation. 2011.

VON ZUBEN, F. J. **Modelagem de Sistemas Dinâmicos Contínuos no Tempo**. DCA/FEEC/Unicamp. p. 27 - 30, 2010.

## CONSTRUCTION AND CONTROL DESIGN OF AN AERO PENDULUM, A MODEL-BASED APPROACH

**Abstract:** *In this paper is described the usage of Model-Based-Design for modeling and control of a Aero Pendulum. MATLAB/Simulink was chosen as automatic code generation software to implement the measurement of sensors, actuators activation and also the controllers of the closed loop system. The P, PD and PID controllers used were tuned using a toolbox of MATLAB that allows you to adjust the controlled pant response dynamics. The used methods objectives were to optimize the final product time development and facilitate its application in educational enviroments.*

**Key-words:** MATLAB, Simulink, Model-Based Design, Aero Pendulum.