

## UMA PLANTA DE CONTROLE DIDÁTICA DE UM HELICÓPTERO COM 1 DOF

Artur Átila Moraes

Orientador: Prof. Luis H. C. Ferreira

Instituto de Engenharia de Sistemas e Tecnologia da Informação - IESTI

**Resumo** - Este artigo apresenta o desenvolvimento de uma planta de controle didática de um helicóptero com 1 grau de liberdade – *Degrees of Freedom (DOF)*. Na realização do projeto, foi utilizada a plataforma *open source* Arduino para fazer o controle PID de um motor *brushless*, o sensor MPU6050 para a realimentação da malha e o driver DRV8833 para substituição do ESC, assim como uma impressora 3D para prototipagem das peças, objetivando o baixo custo. Ao final, será mostrado como a planta, que possui modelo linear de segunda ordem, é estável com um *overshoot* de 16% e tempo de acomodação de 13s.

**Palavras-Chave:** Planta Didática, 1 DOF, Arduino, Controle PID, Motor *Brushless*, MPU6050, DRV8833, *Open Source*, Baixo Custo.

### I. INTRODUÇÃO

Uma das formas de se estudar a teoria de controle na prática é através do uso de equipamentos laboratoriais, que permitam enxergar um controlador atuando em um problema físico concreto. A isso, é dado o nome de plantas de controle didáticas, que geralmente possuem custo elevado, restringindo o acesso livre aos alunos.

Nesse contexto, foi construída uma planta de controle didática de um helicóptero com 1 grau de liberdade, com baixo custo e material *open source*, a fim de disseminar conhecimento facilitando o acesso.

Esse projeto é uma continuação da iniciação científica intitulada “Uma Controladora de Motores *Brushless* Utilizando Eletrônica Embarcada de Baixo Custo para Navegação de Multicópteros”, desenvolvida pelo mesmo aluno e orientador deste trabalho.

Assim, esse artigo irá apresentar o desenvolvimento dessa planta, compartilhada por meio do link <https://github.com/arturmora/Didactic-Control-Plant.git>.

Para melhor entendimento do projeto, este será dividido aqui em: modelagem, mecânica, eletrônica, controlador e programação. Ao fim, serão apresentados os resultados e a conclusão deste trabalho.

### II. MODELAGEM DO PROBLEMA

O modelo físico da planta consiste em uma haste horizontal de comprimento  $l$  que varia seu ângulo  $\theta$  em torno de seu centro geométrico, ligado a um eixo com atrito dinâmico  $B$ . Acoplado à ponta da haste, existe um motor com hélice capaz de gerar um empuxo  $E$ . O empuxo gerado pelo motor é o produto de uma constante  $K$  pela velocidade do rotor  $w$  elevado ao quadrado e  $I$  o momento de inércia. A Figura 1 exemplifica o modelo.

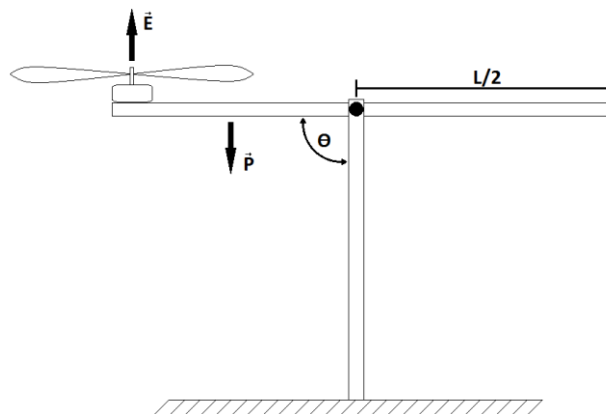


Fig. 1 - Modelo da Planta de um Helicóptero com 1 DOF

Esse sistema tende a ser instável já que o centro de massa está deslocado para o lado do motor. Assim, o problema de controle está em manter a haste na posição horizontal. Através de aproximações, o sistema é considerado linear e de segunda ordem [1], representado pela equação (1).

$$(I + m \cdot l^2) \cdot \theta'' + B \cdot \theta' + m \cdot l \cdot g \cdot \sin(\theta) = K \cdot w^2 \quad (1)$$

### III. MECÂNICA

A estrutura mecânica foi projetada levando-se em conta três requisitos. Primeiro, o projeto da planta deveria ser *open source* e de fácil construção. Assim, as peças foram

projetadas para serem manufaturadas em impressoras 3D. Segundo, o custo deveria ser baixo, para isso a estrutura foi projetada com o mínimo possível de peças. Terceiro, para a aproximação da modelagem ser possível, o atrito entre o braço móvel e a base deveria ser baixo, portanto, foi utilizado um rolamento acoplado ao centro da haste horizontal, como pode ser visto em detalhe na Figura 2.



Fig. 2 – Haste Horizontal com Rolamento no Eixo

As peças, que foram desenhadas no *software* Google Sketchup, foram impressas utilizando filamento ABS, com preenchimento de 15% do tipo *concentric*, utilizando uma impressora 3D modelo Prusa Mendel I2, construída pelo próprio aluno. Assim, a estrutura foi montada com 12 peças impressas e com as peças descritas na Tabela 1.

Tabela 1 - Tipo e Quantidade de Peças

Peça	Quantidade
Parafuso philips M2 8mm	2
Parafuso allen M3 10mm	8
Parafuso allen M3 15mm	4
Parafuso allen M3 25mm	4
Parafuso allen M4 35mm	1
Parafuso allen M8 35mm	1
Porca M3	36
Porca M4	2
Porca M8	1
Arruela M3	64
Arruela M4	3

O braço pode variar o ângulo em 90°, sendo o meio da variação a posição horizontal. O comprimento da haste horizontal possui 34 cm e a vertical 18 cm.

A Figura 3 mostra a planta construída em funcionamento, com o sensor inercial acoplado na ponta direita e o motor na ponta esquerda.

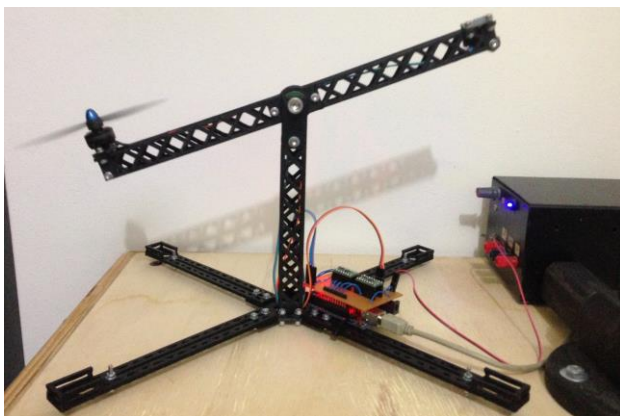


Fig. 3 – Planta em Funcionamento

## IV. ELETRÔNICA

Alguns conhecimentos prévios são necessários para entender como a eletrônica nesse projeto funciona. Assim, algumas informações básicas sobre o motor, o sensor e o microcontrolador utilizados serão explicadas a seguir.

### IV.2 Motor Brushless

O funcionamento de um motor *brushless* (BLDC), que na tradução livre significa “sem bucha”, é similar ao de um motor de passo. Neste, cada par de enrolamento é excitado numa dada ordem fazendo o motor girar.

O motor BLDC possui 3 fios que são ligados às bobinas. Elas são interligadas em configuração delta ou estrela.

O rotor é formado por ímãs permanentes que são atraídos pelo campo magnético criado pelas bobinas do estator, assim ele é considerado de tipo síncrono.

Além disso, ele pode ser de tipo *inrunner*, rotor interno ao estator, ou *outrunner*, rotor externo ao estator.

As vantagens do motor BLDC são sua alta eficiência, alta velocidade, longevidade, baixa manutenção, melhores características de velocidade por torque, elevada densidade de potência e baixo nível de ruídos [2].

A desvantagem está na complexidade de controle do motor, que para isso utiliza um sistema com alto custo, chamado ESC (*Electronic Speed Control*).

#### IV.2.1 Motor Brushless Skyartec 1806

Um ponto crucial do projeto foi a escolha do motor. Este deveria ter baixo custo, baixo consumo de corrente e ser capaz de gerar empuxo suficiente para movimentar a haste. O motor escolhido foi do fabricante Skyartec, modelo 1806, que possui uma constante  $K_v = 2200$ , com consumo máximo de 8 A e uma eficiência máxima de 90% [3].

A constante  $K_v$  está relacionada à velocidade máxima de rotação de acordo com a tensão nominal. A equação 2 mostra essa relação, sendo a tensão nominal desse motor de 3 a 12 V.

$$RPM_{m\acute{a}x} = K_v \cdot Tens\tilde{o}e_{m\acute{a}x} \quad (2)$$

A hélice recomendada para este motor possui tamanho 5x3, construída em fibra de carbono. Essa medida significa que ela tem 5 polegadas de comprimento e possui 3 polegadas de passo, que é o deslocamento a cada volta sob condições normais.

#### IV.2.1 *Electronic Speed Control*

O ESC é responsável por comutar de forma correta as fases do motor *brushless*. Esse circuito contém basicamente um microcontrolador, um circuito de *feedback* e um circuito de potência.

O circuito de potência possui 3 meias pontes H, como mostrado na Figura 4, que podem assumir 3 estados: nível alto, nível baixo e flutuante [4]. Cada meia ponte H é conectada a uma fase do motor.

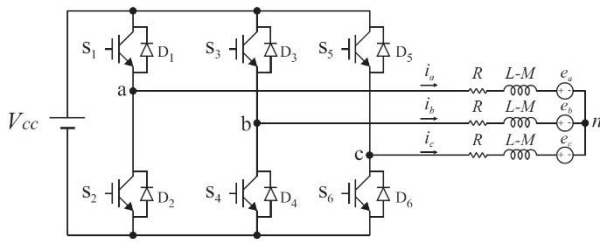


Fig. 4 - Circuito de Potência Simplificado

O circuito de *feedback* detecta a posição do rotor, através de duas técnicas. A primeira técnica é utilizada para motores com sensor, óptico ou de campo magnético (sensor Hall). Esse sensor é ativado quando um ímã passa por ele, indicando a posição do rotor. A segunda técnica é utilizada para motores tipo *sensorless*, na tradução livre “sem sensor”, para isso o circuito detecta a força contra-eletromotriz gerada pelo ímã permanente ao passar pela meia ponte H em estado flutuante [5].

Conhecida a posição do rotor, o microcontrolador decide o momento exato de comutar as fases, seguindo uma ordem estabelecida, mostrada na Tabela 2, que é repetida infinitamente fazendo o motor girar.

Tabela 2 – Sequência de Comutação das Fases

Fase A	Fase B	Fase C	Sequência
Alto	Flutuante	Baixo	1
Alto	Baixo	Flutuante	2
Flutuante	Baixo	Alto	3
Baixo	Flutuante	Alto	4
Baixo	Alto	Flutuante	5
Flutuante	Alto	Baixo	6

A velocidade do motor é determinada pelo tempo entre a troca de estados, sendo que quanto menor o tempo, mais rápido o motor.

#### IV.3 MPU6050

O MPU6050 é um *shield* composto por dois sensores inerciais, o acelerômetro e o giroscópio [6], que juntos são responsáveis pela realimentação da malha de controle do sistema, informando ao microcontrolador qual a orientação angular do sensor. Neste caso, a orientação angular da haste.

O giroscópio mede a velocidade angular de um objeto. Assim, quando o sensor está parado, ele fornecerá valores nulos. Quando em movimento, fornecerá valores positivos ou negativos, dependendo do sentido de rotação do giroscópio.

O acelerômetro mede a força de aceleração aplicada ao sensor, sendo essas forças compostas por forças estáticas, a própria aceleração da gravidade, e forças dinâmicas, quando o sensor sofre alguma força de outra natureza.

O *shield* também possui um processador de movimento embarcado (DMP) capaz de fazer a fusão sensorial e filtrar os dados.

#### IV.4 DRV8833

O *shield* DRV8833, da marca Pololu, é basicamente um *driver* com duas pontes H capaz de fornecer até 1,2 A por canal, trabalhando numa faixa de 2,7 V a 10,8 V [7].

Neste projeto, apenas um lado da ponte é acionado pelo Arduino, logo, ao deixar o outro lado desconectado, a ponte H se comporta como meia ponte. Por serem necessárias 3 meias pontes para acionar o motor, foram utilizados dois *shields*.

Apesar da corrente máxima suportada pelo driver ser inferior ao consumo máximo do motor *brushless*, eles foram utilizados no lugar do ESC por possuírem menor custo, já que nessa condição o empuxo gerado é suficiente para movimentar a haste da estrutura.

#### IV.5 Arduino

Nesse projeto, existe a necessidade de interpretar os dados gerados pelo sensor, verificar a velocidade atual do motor e, com base num controlador PID, decidir se a velocidade deve aumentar ou diminuir no intuito de controlar a planta. Essas funções são realizadas pelo Arduino, um microcontrolador altamente difundido no mundo acadêmico, devido ao seu baixo custo e por ser uma plataforma de *software* e *hardware open source*.

Portanto, foi utilizada a placa Arduino UNO que opera numa frequência de 16 MHz, possui 14 portas digitais e 6 portas analógicas. Além disso, o *shield* MPU6050 e o *shield* DRV8833 são compatíveis com o microcontrolador.

#### IV.5 Esquema de Ligação

Após a escolha de todos os componentes, foi criado um esquema de ligação, representado pela Figura 5, que consiste em conectar os circuitos de acordo com as especificações do fabricante encontradas em seus respectivos *datasheets*.

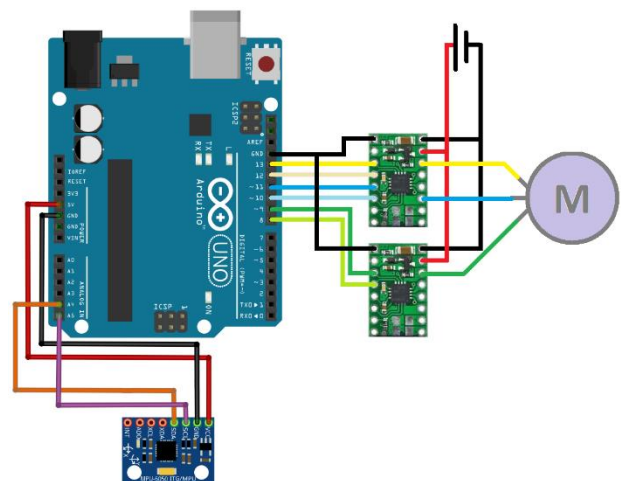


Fig. 5 – Esquemático de Ligação

Os componentes foram conectados e testados utilizando um *protoboard*. Em sequência, os componentes foram soldados sobre uma placa de cobre perfurada, comum em

projetos de prototipagem rápida, utilizando *jumper*s feitos de fio, no lugar das trihas. Para minimizar peças soltas, a placa foi dimensionada para encaixar sobre o Arduino Uno, como pode ser visto na Figura 6.

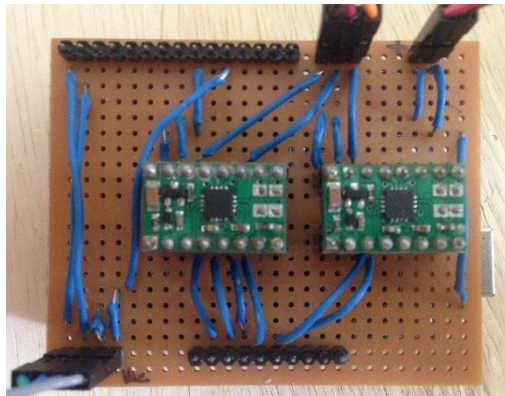


Fig. 6 – Circuito Encaixado no Arduino Uno

## V. CONTROLADOR

O modelo do sistema consiste em uma equação de segunda ordem que pode ser linearizada, como mostrado anteriormente. Desse modo, através da biblioteca PID do Arduino, foi implementado um controlador proporcional-integrativo-derivativo, descrito pela equação (3), capaz de levar o sistema a estabilidade [7].

$$C(s) = K_p + \frac{K_i}{s} + K_d \cdot s \quad (3)$$

## VI. PROGRAMAÇÃO

Considerando que a planta se encontra inicialmente em repouso e com o motor desligado, a haste estará em seu ponto mais baixo, numa inclinação de  $-45^\circ$  em relação ao plano horizontal. Assim, existem duas situações diferentes a serem controladas.

### VI.1 Inicialização

O acionamento correto do motor depende de parâmetros que são definidos no cabeçalho do programa:

- Velocidade mínima e máxima de rotação;
- Velocidade de *start* do motor;
- Portas de saída para acionamento do *driver* DRV8833;
- Ângulo mínimo e máximo da haste;
- *Offset* do *shield* MPU6050;
- *Set point* do controlador;
- Ganhos  $K_p$ ,  $K_i$  e  $K_d$ ;
- Tempo de amostragem.

Além desses parâmetros, o programa necessita das seguintes bibliotecas: controlador (PID\_v1.h), interrupção (TimerOne.h), comunicação I2C (I2Cdev.h) e sensor MPU6050 (MPU6050\_6Axis\_MotionApps20.h).

Após o cabeçalho, o programa entra numa rotina de partida do motor até atingir uma rotação mínima. Em seguida, o sensor é ativado e o controlador é colocado em modo automático, para se fechar a malha do sistema.

### VI.1 Controle

A rotina de controle repete infinitamente as funções:

- Ler novo valor do sensor;
- Calcular velocidade do motor;
- Atualizar velocidade do motor.

O cálculo da velocidade leva em consideração a diferença entre o *set point* e o valor lido pelo sensor. Caso esteja abaixo ou acima do *set point*, a velocidade aumenta ou diminui, respectivamente, de acordo com o valor calculado pela função do controlador PID.

Em paralelo, uma função que troca o estado do circuito de potência para girar o motor, de acordo com a Tabela 2, é chamada através de uma interrupção temporal.

O tempo que determina a ocorrência da interrupção é determinado por uma função de mapeamento, que converte a saída do PID, que varia de 0 a 255, em tempo entre as comutações, que varia de 350  $\mu$ s e 500  $\mu$ s.

Como o microcontrolador não conhece a posição do rotor, uma função que compara o tempo de comutação anterior com o novo limita a aceleração, para não ocorrer a perda de sincronia entre as comutações e a posição real do motor. Portanto, a velocidade obedece uma função rampa, que altera em 1  $\mu$ s o tempo de comutação a cada rotina completa.

## VII. RESULTADOS E ANÁLISES

O controle em malha fechada do sistema inicia após o motor atingir uma velocidade de 3333 RPM. Assim, a velocidade do motor em RPM, que é a entrada do sistema, é manipulada por um controlador PID com ganhos  $K_p = 1$ ,  $K_i = 1$  e  $K_d = 4$ , utilizando um tempo de amostragem de 30 ms. A Figura 7 apresenta esse controle, sendo o instante zero o momento de fechamento da malha.

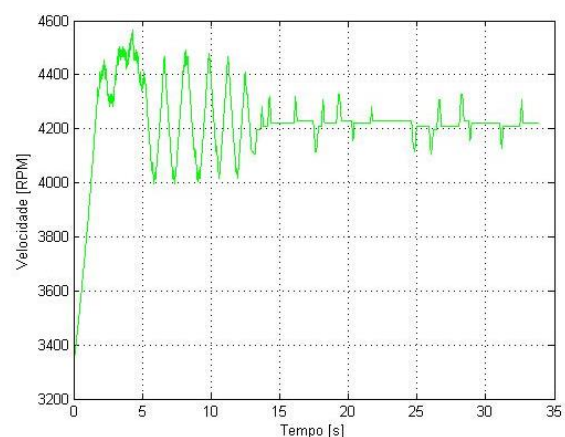


Fig. 7 – Gráfico da Entrada

Os ganhos foram sintonizados de forma empírica, escolhendo um valor de  $K_p$  para qual o sistema tivesse um *overshoot* de até 50%, em seguida foi adicionada a parte derivativa aumentando-a até o *overshoot* ficar abaixo de 20% e por fim foi adicionado a parte integrativa para zerar o erro em regime permanente. A Figura 8 mostra a



resposta do sistema, representada em graus, que representa a posição da haste horizontal em relação a vertical. O *set point* de 0° indica a haste posicionada horizontalmente.

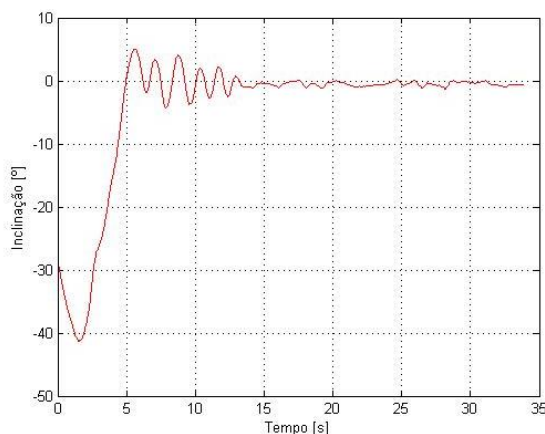


Fig. 8 – Gráfico da Saída

Foi obtido um overshoot de 16% e tempo de acomodação de 13s. A variação negativa no início do gráfico é devido a um erro de acomodação na inicialização do sensor. O sistema é estável na presença de distúrbios, garantindo um controle robusto. Isso foi verificado criando um distúrbio no lado direito da haste, batendo nela com a mão para cima e para baixo, como mostra a Figura 9.

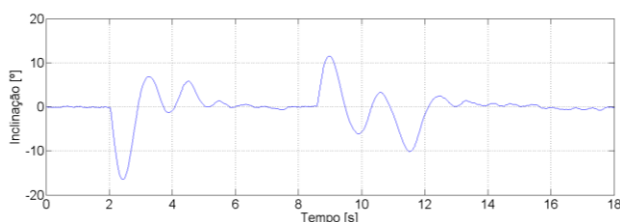


Fig. 9 – Gráfico da Saída com Distúrbio

Algumas vezes, pelo fato da técnica utilizada para rotacionar o motor não utilizar a localização da posição do rotor, o campo girante perde a sincronia com o rotor, parando o funcionamento do sistema.

O custo estimado do projeto foi de R\$147,00. Sendo R\$15,00 em parafusos; R\$15,00 do sensor MPU6050, R\$40,00 do motor, R\$12,00 da hélice, R\$15,00 do Arduino, R\$40,00 do *driver* e R\$10,00 em fios e filamento.

## VIII. CONCLUSÃO

Este trabalho final de graduação tinha como objetivo principal o desenvolvimento de uma planta na qual pudesse ser utilizada uma técnica de controle. Como foi possível utilizar o controlador PID, o objetivo foi alcançado. Outro objetivo era desenvolver um projeto *open source*, de baixo custo e fácil construção, trazendo a possibilidade de que outros alunos construam a planta didática a fim de estudar o controlador PID ou implementar outras técnicas de controle.

O aluno pôde se aprofundar em diferentes áreas da engenharia, como a mecânica, a eletrônica e a programação, mas também teve a oportunidade de utilizar os conceitos aprendidos durante a graduação. Além de desenvolver habilidades voltadas para o empreendedorismo, como planejamento de um projeto. Em trabalhos futuros poderá ser feita a identificação do sistema através de ensaios em malha aberta, para a modelagem do sistema. Além disso, é possível fazer a adição de mais um grau de liberdade, adicionar a técnica de localização do rotor e desenvolver uma interface gráfica, assim, o aluno poderia alterar os valores das constantes do controlador PID e gerar gráficos de entrada e saída do sistema em tempo real.

## IX. AGRADECIMENTOS

Primeiramente agradeço a minha família, a minha namorada e aos meus amigos, que me deram apoio para continuar acreditando e carinho para sentir que vale a pena. Agradeço também aos meus professores e toda a comunidade *open source* que dividem o conhecimento, sabendo que só assim teremos um futuro melhor. Finalmente agradeço ao meu orientador e ao grupo de microeletrônica, que me acolheram desde o início da faculdade, sempre me orientando, tornando esse TFG e outras conquistas possíveis.

## X. REFERÊNCIAS

- [1] K. Ogata e Y. Yang – “Modern control engineering”, Prentice-Hall Englewood Cliffs, NJ, 1970.
- [2] P. Yedamale - “Brushless DC (BLDC) Motor Fundamentals”, *Microchip*, 2003.
- [3] Skyartec – “1806 Brushless Motor (2200Kv CW)”.
- [4] D. Arrigo - “L6234 Three Phase Motor Driver”, STMicroelectronics, Abril, 2001.
- [5] W. Brown - “Brushless DC Motor Control Made Easy”, Microchip, 2002.
- [6] Invensense – “Mpu-6000 and mpu-6050 register map and descriptions”, 2012.
- [7] Texas Instruments – “Dual H-Bridge Motor Driver”, 2013.

## BIOGRAFIA:



### Artur Átila Moraes

Nasceu em São Paulo (SP), em 1992. Estudou em São Caetano do sul, tendo recebido o título de Técnico em Eletrônica na Escola Técnica Estadual Jorge Street. Ingressou na UNIFEI em 2011, no curso de Engenharia de Controle e Automação, onde realizou duas iniciações científicas como bolsista do CNPq. Estagiou na RZX Tecnologia, por 6 meses, desenvolvendo um sensor utilizado na área de óleo e gás. Em 2015, ganhou o concurso “Internet of Cows”, da BovControl, para desenvolver um sensor *wearable* com futuro registro de patente.